Andrzej Indrzejczak

Trends in Logic 30

Natural Deduction, Hybrid Systems and Modal Logics

Springer

Natural Deduction, Hybrid Systems and Modal Logics

TRENDS IN LOGIC

Studia Logica Library

VOLUME 30

SCOPE OF THE SERIES

Trends in Logic is a bookseries covering essentially the same area as the journal *Studia Logica* – that is, contemporary formal logic and its applications and relations to other disciplines. These include artificial intelligence, informatics, cognitive science, philosophy of science, and the philosophy of language. However, this list is not exhaustive, moreover, the range of applications, comparisons and sources of inspiration is open and evolves over time.

Volume Editor
Heinrich Wansing

For further volumes:
http://www.springer.com/series/6645

Andrzej Indrzejczak

Natural Deduction, Hybrid Systems and Modal Logics

Springer

Dr. Andrzej Indrzejczak
University of Lódz
Dept. Logic
Kopcinskiego 16/18
90-232 Lódz
Poland
indrzej@filozof.uni.lodz.pl

ISBN 978-90-481-8784-3 e-ISBN 978-90-481-8785-0
DOI 10.1007/978-90-481-8785-0
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2010922895

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

Introduction

A good title should be informative enough to illuminate a potential reader on the content of a book. We hope that the present title gives at least some hints of what this book is about. The notion of natural deduction or modal logic are rather well known, but the notion of "hybrid system" certainly needs some explanation.

In short, this study may be seen as a kind of search for good deductive systems. We think of systems good in practice which may be applied with ease not only by well trained logicians but also, for example, by philosophers who need handy deductive tools accompanying their analyses. In particular, we are interested in providing systems that may be widely applied in teaching logic. Nowadays one may observe that several courses in "critical thinking" tend to eliminate courses in practical logic. On the other hand, logic is often taught as a strictly mathematical discipline in very demanding courses. It is important to fill the gap between these extrema, and the crucial ingredient of any course which is supposed to teach how to use logic, is certainly a suitable deductive system.

Since we address this work to a wide audience interested in applications of logic, we were trying to make it self-contained and accessible to a reader with no hard training in logic. The assumed reader should have some background in logic (an elementary course covering classical propositional and first-order logic with basics of set theory is enough) but not necessarily in modal logic.

The search for a good deductive system is realized in stages. Standard natural deduction for classical and free logic is investigated in the first place as the proper candidate for this aim. Closer inspection shows that natural deduction in standard form has some limitations (which should) to be overcome. In search of better deductive tools we introduce some modified versions of natural deduction and the so called hybrid systems – combinations of natural deduction with other kinds of calculi. Next, applications of natural deduction in standard and extended (hybrid) form to several modal logics are analyzed. Finally, a labelled approach is examined: first, in external form and then, in the strong internalised form, commonly called hybrid logics.

If the aim of the study is to find a good deductive system, then we should ask what does "good" mean with respect to a deductive system? This is a very general question, concerning a very vague notion that may cover many different things. We must underline again that we think of practically usable systems; theoretical considerations often lead to quite different desiderata. Let us try to make some preliminary list of particularly important properties.[1] A good practical deductive system should be:

- universal
- general
- extensive
- natural
- simple
- efficient

These terms are not technical (except the last perhaps) but informal and ambiguous. In the following explanations we stipulate meanings which we want to attribute to them, at least in this book.

We say that a proof system has *universal application* (or simply that it is universal) if it may be used to perform different deductive tasks. For example, a universal system allows not only constructing proofs but also showing that a formula is invalid by extracting a falsifying model. It makes possible to define proof search procedures, and even if the formalized logic is not decidable, it gives some ground for application in automated theorem proving. Typical tableau and resolution calculi, and to some extent, sequent calculi, satisfy this property, whereas axiomatic systems and natural deduction systems in their standard form, are not universal.

By *generality* of a system we mean the ability to apply different proof search strategies and to simulate in a direct fashion other kinds of systems.

[1] One may find other interesting lists of valuable properties e.g. in Avron [16], Wansing [280], Indrzejczak [143] or Poggiolesi [214]. But those lists are often more theoretically oriented and formulated mainly for several forms of sequent calculi, whereas the present one has a very general character and is more related to practice.

Several technical notions of simulation will be defined in Chapter 1. Informally, we mean that it is possible to apply deductive techniques from several sources in general systems. In consequence they may be used as a handy tool for a comparison of different proof-search strategies and their efficiency.

Extensiveness of a system is connected with the scope of its applicability. It means that the system provides a uniform deductive framework for the formalization of several nonclassical logics. Extensive systems yield handy tools for the investigation of different logics in a uniform fashion. So far, axiomatic systems are unquestionable winners in this category. But recent developments of sequent calculi, especially of nonstandard character (like display calculi or hypersequent calculi), or tableau calculi offer some hope in this respect.

A proof system is *natural* if its rules are modeled after traditional methods of inference, known from antiquity and used by humans in their common thinking, as well as in informal mathematical proofs. Natural deduction systems seem to satisfy this requirement better than other systems, because the latter are often limited to the use of special types of rules only, regulated rather by theoretical than practical needs. It is not surprising; Jaśkowski and Gentzen had just this goal in mind when they have constructed the first systems of this sort. Most variants and modifications introduced later were also generally connected with this idea.

Naturalness seems to be in close connection with *simplicity* of the system but this property is an example of particularly vague notion. Moreover, several possible senses are hardly subject to any objective criteria. Anyway, it is worth exploring. In the case of proof systems simplicity means, among other things:

1. simplicity of inference rules;
2. simplicity of the construction, and the limited number of elements of the whole system (easy to describe, to implement);
3. easy to follow proofs, readable for humans;
4. ability to construct short and direct proofs;
5. applicability of simple proof search strategies.

It is easy to observe that these features are rather independent and, moreover, sometimes they even tend to be in conflict. For instance, the possibility of building short and direct proofs is usually the result of the rich structure of the system. On the other hand, systems simple in the sense 1 or 2 are often unable to produce short and easy-to-follow (and to find) proofs. For example, axiom systems are certainly simple in the 1st and 2nd sense, which is the source of their success in metalogic. Axiomatic proofs also have, in a sense, a very simple structure, but it does not mean that they are readable or short, or easy to find! Natural deduction systems are usually simple in the sense 1, 3, and 5, but the price for that is a complex structure of the calculus. Similar remarks may be applied to other types of proof systems which will be discussed below.

The notion of *efficiency* is usually applied on the field of automated theorem proving and measured in terms of speed of running program or memory required for computation. But proofs generated by efficient programs may be quite long and complicated, while it is possible to find short and direct proofs with the help of some ingenuity. On the other hand, algorithms devised for finding short and readable proofs tend to be rather more complicated. It means, in terms of implementation, that suitable programs may require much more time and memory. We are concerned in this study with systems of practical utility designed for humans not for machines, so efficiency is a good thing, but not at the expense of other features like naturalness or simplicity. Such an approach does not exclude automatization but introduces considerable complications because a proof generated by a program should be readable for men.

Natural Deduction

In general it is not reasonable to claim that some type of a system is better than other ones. The best we can do is to evaluate systems as better than others with respect to some of the properties. For example, if we compare resolution and axiom systems, then certainly, the former is more efficient but the latter is more extensive. In fact, not all discussed properties may be used as serious criteria of evaluation. For example, when we compare different systems with respect to naturalness and simplicity (as explained above) such an evaluation must be subjective. But still it is reasonable to search for systems that may be assessed as having sufficiently high rating in all categories.

In our opinion, natural deduction systems (shortly called ND systems) seem to be the most promising but their abilities were not fully recognized so far. In this book we will try to justify our belief and to show that ND may be extended and generalized in many ways.

What supports our conviction is, in the first place, the richness of deductive apparatus of ND. It makes even standard ND quite general type of a system but, as we will show, simple modifications may considerably increase their generality. We have already mentioned in what sense ND systems may be called natural and simple. In fact, many existing ND systems may provoke an opinion that they are natural only by definition. We will focus on the systems which, in our opinion, are the simplest, the most natural, and moreover, they may be modified in several ways.

In common opinion ND systems are not very useful as a tool for proof search or for automation. It is a consequence of unnecessary limitation of standard ND which are defined as systems devised for proof construction only. In this form they are not universal since realization of other deductive tasks, like falsification of a formula, is not possible. If it is an essential property of ND, it would be more proper to call them "natural *proof* systems" instead of "natural *deduction*".[2] Fortunately, it is not difficult to make ND systems more universal.

The possibility of using ND systems as, e.g. working decision methods, opens the gate for the question of efficiency of ND and prospects for automation. In fact, ND systems are rather not considered as good candidates for that purpose. Rich deductive toolkit mentioned above may be rather troublesome for implementation. So it is not surprising that there are not so many provers based on ND (cf. Chapter 4). The unquestioned leader in the field is the family of resolution calculi, but programs based on resolution usually do not satisfy the requirement of readability of the output. Clearly, if the result (not the way leading to it) was the only important factor, it is inessential. But we stated above that for us rather the way than the result is more important, and it is also somewhat connected with automated deduction. Since 80s, more attention is paid to the construction of several forms of interactive programs for teaching logic, with some support built in. One may note that many programs of this sort, like MacLogic, Heterogenous Logic or Mizar, are based on ND systems.

These brief remarks on some desiderata concerning good proof systems are intended as an explanation of the leading role of ND systems in this book. Details that substantiate our claims will be found in the text. We did not touch so far the question of extensiveness of ND.

[2]By the way, it is more proper to use a term "automated deduction" instead of often applied "automated theorem proving" since modern programs are not bound to proving but realize a variety of deductive tasks, e.g. model checking.

Modal Logic

In this book we focus only on one class of nonclassical logic, however, very important one. The choice of *modal logics* as the field of application of investigated deductive techniques follows from the personal conviction that it is one of the most natural and useful class of logics. This view corresponds well with our program of searching for the most natural, practical and simple system of deduction. Let us notice however, that although we do not extend our results to other nonclassical logics, we take the term "modal logics" in a wider sense than it is usually applied. In particular, we consider:

1. Not only monomodal logics but also multimodal ones, in particular, bimodal temporal logics.

2. Apart from normal modal logics, also weaker classes of regular, monotonic and congruent logics.

3. Several versions of first-order nonmodal and modal logics.

4. Not only modal logics formulated in standard languages but hybrid modal logics in extended languages.

We believe that systems providing uniform characterization of this vast and diversified class of logics are extensive enough. Moreover, many results may be easily adapted to other nonclassical logics. In case of intuitionistic or some superintuitionistic logics it is straightforward, e.g. systems for modal logics of linear frames may be easily redefined to obtain a formalization of Dummett logic, in other cases it may need some work. But existing ND systems for many nonclassical logics not discussed in this book, e.g. for relevant logics in [5], may be seen as an additional evidence for our claim.

Hybrid Systems

Finally, we should explain what is meant by a *hybrid system*. Readers acquainted with modern modal logic may suspect that we mean deductive systems for *hybrid logics*. In fact, hybrid logics and deductive systems for them (including hybrid systems as well) will be also dealt with in this book. But the term hybrid system is by no means reserved for hybrid logics. Basically, in this book, this qualification is applied to several kinds of deductive systems developed by combination of elements taken from different sources. The construction of such systems is undertaken for optimalization of deduction.

The need for hybrid systems is closely connected with the evolution of logic. Development of computer sciences, investigations on artificial intelligence, problems with knowledge representation and management, and many other related factors, resulted in substantial changes in logic. Problems traditionally seen as technical and disregarded by logic community, currently provide the main areas of research. These changes, making logic less theoretical and more practically oriented, have strong impact on the methodology of deduction as well.

One of the cornerstones of modern formal logic is the distinction between syntactical and semantical investigations. What is defined in terms of the shape of expressions belongs to syntactical studies, what is defined in terms of interpretation belongs to semantics. A privileged position of *completeness* and *soundness* proofs in many studies is a very good witness of the importance of this distinction. In traditional reflection on methods of logic both approaches are treated as complementary. Syntactical methods are seen as tools for proving, whereas semantical methods are seen as tools of falsification. The introduction of systems of universal character, like tableaux, did not change this popular view. Even today in many modern textbooks one may find this tendency still alive.

Likewise, in traditional metalogic also *decidability* was taken seriously but practical aspects were rather ignored. It was enough to show that there is an algorithm which in finite time may always find an answer. Development of computer industry and growing interests in fast-running programs have changed the perspective. Even in case of undecidable logics one may construct reasonably efficient programs, thus undecidability does not exclude automation. On the other hand, even decidable theories may be practically nontractable if they require too much time or memory. A dynamic development of *complexity theory* is one of the signs of this trend. Searching for efficient methods seems to be much more important nowadays than proving completeness or decidability. From the standpoint of practical applications of logic (like, e.g. in *expert systems* or *computer-aided decision making*) one is rarely interested in complete systems but always in fast ones. The great success of *Horn clauses* may serve as an example of wide applicability of such "partial" logics.

Hence, in search of optimal logical tools, problems of theoretical purity are not so important. Good results are often obtained by free combination of tools taken from several sources. Deductive systems resulting from such operations are called here hybrid systems. One may distinguish at least two types of hybridization:

1. An introduction of syntactically encoded elements of semantics into the realm of deductive (syntactic) system.

2. A combination of different types of deductive systems which were originally devised for the realization of different deductive tasks.

Both types of hybridization are of unequal status. Using elements of semantics in deductive systems has rather long tradition. In fact, the application of semantical information was always a natural way of supporting deduction. For example, it often appeared in several forms of diagrams representing (partial) interpretation.[3] Development of mathematical logic, and, in particular, the popularity of Hilbert program in its first phase, led to absolutization of the distinction between syntax and semantics, as we have already remarked. But this is not only artificial; it may be harmful, both from the standpoint of optimal tools of deduction, and of the practice of teaching logic. As for the latter case one may note a great popularity of such programs for teaching logic like *Tarski's World* or *Heterogenous Logic* (both due to Barwise and Etchemendy). Generally, it seems that several attempts at erasing the border between semantics and syntax appeared already in the first half of XX century. One may compare in this respect the approaches of Beth or Quine in their books on logical methods [27, 226].

Hybridization of this type is realized in different ways in modern logic. Some of the techniques applied in systems for modal logics, like introduction of diagrams, rectangles, brackets into tableaux, or generalizations of sequents (hypersequents, multisequents, e.t.c.) will be shortly described in Chapters 7 and 8. But the main technique for syntactical encoding of semantics considered in the book is labelling. It is superior to other techniques in two respects. First, labels may take different forms dependant on the kind of (elements of) semantics we want to use and on the grade of semantical involvement. Second, labels may be applied in combination with any kind of deductive system. In particular, deductive systems for hybrid logics may be also seen as an important example of labelled systems of a very special sort.

The second type of hybridization was not widely applied so far but it may lead to the great improvement of traditionally recognized deductive

[3] One may find a survey of such techniques in, e.g. Bocheński [43], Kneale [163], Marciszewski and Murawski [181].

systems. We have already remarked that no single type of a system may be viewed as better than others in all respects. But a reasonable combination of rules, techniques, strategies taken from different sources, may result in a system which inherits good properties of all parents. Such systems may realize different deductive tasks in satisfying way without the necessity of changing the basic system. The attempts of this kind are undertaken for the creation of integrated environments for working with different logics or theories. We mean here a variety of programs qualified as a *logical frame* or *generic prover*, like Isabelle, Automath, Otter or Mizar.[4] In our opinion, ND systems, due to their rich assortment of deductive tools, are best prepared to work as an uniform basis for the integration of other types of rules and techniques. In what follows we will provide some examples of such ND-based hybrid systems.

Overview

The book may be divided into three parts. The first part comprises 4 chapters and lays down the foundational issues concerning ND systems in standard and extended form. Chapter 1 introduces classical and free logics and several technical notions concerning deductive systems, rules e.t.c. The next Chapter presents a short history and a systematization of several forms of standard ND systems. The distinction between different formats of ND is essential since, as we shall see, not all of them may be used as a suitable basis for extension to modal or other nonclassical logics. In this book we focus on ND in Jaśkowski's format, in the version called KM due to Kalish and Montague. Several variants and generalizations of KM are provided in due course, but it is usually pointed out whether these modifications may be adjusted also to other versions of ND. Quite independently of the chosen ND format a lot of different formalizations of first-order classical and free logic were proposed. This is in contrast to rather uniform treatment provided for sequent calculi or tableau systems. We characterize the most important of these several solutions either. The next step is to overcome limitations of standard ND. For that reason in Chapter 3 we survey different types of deductive systems that may provide inspiration for generalizations of ND. It is a subject of Chapter 4, introducing analytic and universal versions of standard ND able to simulate tableau systems and KE system of D'Agostino and Mondadori. Finally, we introduce RND (resolution based ND) oper-

[4]Cf. Basin [23] for a good introduction into this theme.

ating on clauses, a powerful generalization of ND directly simulating many clausal systems like resolution calculi or Davis/Putnam method.

The second part (Chapters 5, 6, and 7) is concerned with modal logics and their formalization in standard ND systems. Chapter 5 provides a short introduction to the families of modal logics which are dealt with in this book. In Chapter 6 we present standard formalization of modal logics in sequent and tableau calculi and survey different approaches to construction of ND for modal logics. One of them, due to Fitch, is then exploited because it is the most extensive. We provide a few versions of KM system based on Fitch's approach. Chapter 7 shows some other possible extensions based on standard approach; in particular, two variants of RND for modal logics are introduced. Some theoretical problems, as well as limitations of standard method, are also discussed. We end up with the introduction of some nonstandard approaches to formalization of modal logics.

The third part is devoted to detailed exploration of one of the nonstandard approaches to formalization of modal logics, based on the application of labels. The term labelling is treated here in the very wide sense, comprising also hybrid logics, seen as a form of internal labelling. In Chapter 8 we start with general remarks on several forms of labelled systems and focus on the popular solution due to Fitting. In particular, we demonstrate that it is possible to apply this technique also to weak modal logics and combine labels not only with ND but also with RND. Chapter 9 shows that Fitting's labels may be also used to formalization of logics characterized by linear frames. This group is treated separately not only because of its importance (for instance in formalization of linear time) but because of specific technical problems we encounter. Moreover, the solution we propose may be also applied to other important logics determined by frames described by the so called universal implications. Chapter 10 has more technical character. We present a series of constructive completeness proofs for many analytic labelled ND systems, based on suitable proof search procedures. Some questions of efficiency and optimization are also briefly discussed.

Although systems based on the application of Fitting's labels are more extensive than standard ND systems, they still suffer from some limitations. The use of stronger forms of labelling leads to more extensive solutions. Particularly interesting form is provided by the use of hybrid logics. Chapter 11 is a short introduction to varieties of languages covered by this term. In Chapter 12 we survey deductive systems provided for hybrid logics and provide ND and RND systems of a very extensive character. In fact, these two chapters may be treated as a separate part, where the most extensive results are eventually offered.

Thus one may note that we try to extend the possibilities of ND systems gradually. Standard ND is first modified to obtain universal and analytic form for classical logic. Next we examine different ways of extending standard ND to modal logics. Then we add external labels to improve extensivity and to obtain an analytic ND for modal logics. At the end we introduce stronger (internalised) form of labelling which is the most extensive solution.

Finally, we should say a few words on what this book is not about.

Although there are parts of it where some algorithms are stated it is not a book on automated natural deduction. ND is treated here as a practical deductive tool of a great pedagogical value, useful rather for pen and paper implementation. Decision procedures for some systems are defined for the need of constructive completeness proofs rather than for real application. We are interested in proof search but performed by means of natural and simple rules. Hence although we show that resolution may be simulated in ND, no discussion of unification, and skolemization is provided.[5] In our opinion these techniques, despite their efficiency, are not natural; their application may speed up a proof, but not make it more readable. We also do not describe resolution proof search strategies since they are presented in many places; the interested reader is encouraged to test for himself how they work in the setting of RND.

Again, for the same reasons we do not take up theoretical questions connected with ND. In particular, problems of normalization of proofs are sometimes signalled but no systematic treatment is provided. In our opinion these results are very important but have rather theoretical character, whereas this study is concerned with these aspects of ND which may simplify doing proofs by hand. Also, no discussion of matters concerning encoding ND in lambda calculus through Curry-Howard isomorphism, e.t.c. is provided. These are certainly very important aspects of investigation on ND but again of theoretical character and very technical in nature. Inclusion of such considerations would result in another book, certainly harder to write. Instead, the question of analyticity is investigated as having a serious impact on the practice of proof search.

In the search of universal, general and extensive variants of ND we make free use of other types of deductive systems. But this is not a book on other

[5] One may find a presentation of respective forms of these techniques suitable for ND in Pollock [217].

proof systems so many of them are not even mentioned. In particular, our presentation of deductive systems for modal logics is far from being complete. Only these systems are introduced that satisfy at least one of the following criteria:

- They may be combined in some way with ND systems.
- They are extensive (at least potentially they have wide scope of application).
- They offer a possibility of formalization of bimodal temporal logics.
- They provide rules for logics of linear models.

The first criterion is obvious since we have chosen ND systems as a basis for obtaining uniform, general and universal hybrid systems. Remaining ones are connected with the application to modal logics.

Since the book is intended to a wide audience interested in practical application of logical tools, a detailed statement of technicalities is often avoided. In particular, if some definitions or proofs go along similar lines as those previously stated, the details are usually omitted and we rather encourage the reader to do it as an exercise. Certainly, the readers interested only in the application of systems, not in proving their properties, may skip respective parts of the text. On the other hand, there are some harder and more demanding parts which may be of interest for more technically oriented reader. They are generally connected with establishing adequacy of described ND systems. Since we are concerned with practical application of ND we were trying to reduce to minimum such metalogical proofs. Completeness results in most cases are obtained by simulation of other complete deductive systems, stated in quite an informal way and rather easy to follow. But in some cases there is no possibility to simulate something better known and full completeness proof must be stated instead (e.g. for labelled ND for linear logics in Chapter 9). Similarly, proofs of decidability based on proof search procedures defined for analytic version of ND in Chapters 4 and 10 are stated in detail, at least in these parts where specific features of ND play the crucial role. We have paid an attention also to soundness proofs for ND systems. Obviously, the reader interested only in finding working ND systems may skip these parts of the text without a loss.

This book is based on the author's habilitation published in Polish in 2006, and defended succesfully in 2007. Almost half of the present text is a (rather free and revised) translation of much of it. But this work contains substantial enlargement of that book. In particular, we have added here a treatment of first-order (classical, free, modal) logics, formalizations of weak (congruent, monotonic, regular) modal logics, and extended strongly the exposition of standard ND for modal logics and a treatment of hybrid logics. As a result, the book contains some parts based not on the habilitation but rather on some of my other papers devoted to natural deduction and modal logics. All dependencies on my other papers are credited in the text; in particular, Chapters 11 and 12 are heavily based on [155].

Acknowledgments

The list of people to whom I am greatly indebted is long. I would like to express my gratitude to my colleagues from the Department of Logic of Łódź University for stimulating environment and support, and to all friends who have read parts (or even all) of my work. I have learnt a lot from their valuable comments. In particular, I wish to thank: Szymon Frankowski, Janusz Kaczmarek, Grzegorz Malinowski, Marek Nasieniewski, Ewa and Adam Niewiadomski, Marek Nowak, Jerzy Perzanowski, Andrzej Pietruszczak, Marek Rosiak, Dorota Rybarkiewicz, Renate Schmidt, Andrzej Wiśniewski, Ryszard Wójcicki. Needless to say that I take a full responsibility for all the remaining defects.

I also owe a big debt of gratitude to Ryszard Wójcicki who was the source of kind encouragement in writing the book. I would also like to thank to Heinrich Wansing for his helpful support, and to Ingrid van Laarhoven and Willemijn Arts from Springer for their professional support.

The book is dedicated to the memory of Professor Jerzy Perzanowski who died in this year after long illness. His interests in my work, demonstrated in reviews of both my PhD thesis and habilitation, and in numerous comments I have obtained, were always very important for me.

Finally, I wish to thank my family for love and patience.

Łódź 2009 Andrzej Indrzejczak

Chapter 1

Preliminaries

This Chapter has an introductory character. The main objective of Section 1.1. has been to recall the basic information on the language of *classical propositional logic* – **CPL** and on the *quantificational logic* in classical (**CQL**) and free version (**FQL**). The approach chosen here is rather informal. In case of **CPL** we introduce only the language and syntactical conventions applied throughout, while in case of **QL**, a brief outline of classical and free logic is additionaly highlighted by some comments concerning philosophical motivations. The section contains also some technical information, e.g. on relations and trees, essential in the foregoing. It should be emphasized that this section is just to establish notation and to keep the text self-contained, so much of it may be skipped in the first reading and consulted when necessary for understanding later chapters.

Section 1.2 introduces some background information on formalization of logic in general, and the taxonomy of popular types of deductive systems. It is of a slightly different character than the previous one since it amplifies some theoretical apparatus concerning *deductive systems* extensively used in the sequel (although not in common use). Therefore, more careful reading is recommended here before going further.

1.1 Classical and Free Logic

1.1.1 Basic Propositional Language

Most of the material of this book is devoted to propositional logics, so it is reasonable to treat this part of the formal machinery separately. It is not necessary to present a classical propositional logic (**CPL** in short) since a

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*,
Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_1,

reader may find excellent presentations of it elsewhere. In this subsection we confine ourselves only to the introduction of the language, basic notation and conventions which will be used extensively in the text.

In what follows we will use rather standard form of propositional language for classical logic. Let $\mathbf{L_{CPL}}$ denote an abstract algebra of formulae:

$$\langle FOR, \neg, \wedge, \vee, \rightarrow \rangle \tag{1.1}$$

with denumerable set of propositional symbols.

$$PROP = \{\ p, q, r, \ldots, p_1, q_1, \ldots\} \subseteq FOR \tag{1.2}$$

Operations of this algebra correspond to the well known functors (or connectives) of *negation, conjunction, disjunction* and *implication.* φ, ψ, χ will be used to denote any formulae in $\mathbf{L_{CPL}}$ and in each language considered further. For unary functors we will apply prefix notation, and for binary functors – infix notation; in particular the rules for obtaining the set FOR of all formulae of **CPL** look as follows:

- if $\varphi \in FOR$, then $\neg\varphi \in FOR$;
- if $\varphi \in FOR$ and $\psi \in FOR$, then $(\varphi \odot \psi) \in FOR$, where $\odot \in \{\wedge, \vee, \rightarrow\}$

Elements of $PROP$ and their negations are called *literals*; positive and negative, respectively.

To limit the number of necessary parentheses, we admit a convention governing the strength of argument's binding. For binary functors we assume that $\wedge$ binds tighter than $\vee$, and $\vee$ binds tighter than $\rightarrow$.[1] Negation, and all unary functors in general, are assumed to bind their arguments tighter than binary functors. Additionally we omit outer parentheses for any formula and inner parentheses in case of many occurrences of associative operations like conjunction or disjunction. Thus:

$$p \vee \neg q \wedge r \rightarrow p \wedge \neg s \vee q \wedge s$$

is meant as a shortcut for:

$$((p \vee (\neg q \wedge r)) \rightarrow ((p \wedge \neg s) \vee (q \wedge s)))$$

[1] In case we use additional functor $\leftrightarrow$ we assume that $\rightarrow$ binds tighter.

After Smullyan [261] we will use symbols α, β for denoting the following types of formulae:

α	α_1	α_2	β	β_1	β_2
$\varphi \wedge \psi$	φ	ψ	$\neg(\varphi \wedge \psi)$	$\neg\varphi$	$\neg\psi$
$\neg(\varphi \vee \psi)$	$\neg\varphi$	$\neg\psi$	$\varphi \vee \psi$	φ	ψ
$\neg(\varphi \rightarrow \psi)$	φ	$\neg\psi$	$\varphi \rightarrow \psi$	$\neg\varphi$	ψ

Following the above convention we will often divide compound formulae on α- and β-formulae. Γ, Δ, Σ denote any (usually finite) sets of formulae.

Definition 1.1 (Complements, subformulae)

- $-\varphi$ denotes the *complement* of φ or complementary formula of φ (sometimes also called a conjugate); it denotes the negation of φ, if it is unnegated formula, otherwise it refers to the formula where negation is deleted (it is ψ, if $\varphi = \neg\psi$, or $\neg\varphi$ otherwise). $-\Gamma$ is the result of turning all elements of Γ into their complements.

- Subfor(φ) (Subfor(Γ)) is the set of all subformulae of formula φ (the set of formulae Γ) defined as usual. The union of all subformulae of the set Γ and their complements will be denoted as $\overline{\Gamma}$, in particular, for single formula φ such a set will be denoted as $\overline{\{\varphi\}}$

Let us note that the complement of any α-formula is always β-formula and conversely: the complement of any β-formula is α-formula. Also the complement of every literal is always a literal.

We will often use also propositional constants: $\bot, \top$ and binary connective of equivalency $\leftrightarrow$, defined in a standard way:

Definition 1.2 **($\bot, \top, \leftrightarrow$)**

$$\bot := p \wedge \neg p \ ; \quad \top := \neg\bot$$

$$\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$$

Note that if $\bot$ and $\top$ are in use they are also treated as literals.

Every finite set of formulae may be interpreted either disjunctively – as a disjunction of its elements (D-set), or conjunctively (C-set). If we need to show explicitly the intended interpretation of Γ as D-set (or C-set), then we use symbols $\vee\Gamma$ ($\wedge\Gamma$). Empty D-set is understood as $\bot$; empty C-set is understood as $\top$.

Sets containing a pair φ, $-\varphi$ are of special importance and will be called *complementary*. Complementary D-set is obviously interpreted as $\top$ whereas complementary C-set is interpreted as $\bot$. Following Fitting's custom we will often use the name "clause" instead of D-set; it is a natural generalization of the standard use of this notion. Ordinary clauses containing only literals will be called *atomic clauses*. In particular, *Horn clauses* are atomic clauses with at most one positive literal. Let's note that a formula in normal conjunctive-disjunctive form (CNF) is a C-set containing only atomic clauses (or elementary disjunctions); dual disjunctive-conjunctive form (DNF) is a D-set containing only C-sets of literals, called elementary conjunctions (i.e. made of literals only).

Ordered pairs consisting of finite (possibly empty) C-set followed by D-set will be called *sequents*, and denoted as $\Gamma \Rightarrow \Delta$, where C-set Γ is the antecedent and D-set Δ is the succedent of a sequent. In view of the above claims concerning C- and D-sets, $\Gamma \Rightarrow$ means that $\Gamma \Rightarrow \bot$, whereas $\Rightarrow \Delta$ means that $\top \Rightarrow \Delta$; $\Rightarrow$ is just another notational convention for $\bot$. Note that Horn clauses are often presented as sequents with an atomic clause consisting of only positive literals as the antecedent and at most one positive literal in the succedent. So instead of $\neg p \vee \neg q \vee \neg r \vee s$ we may write $p, q, r \Rightarrow s$ to the same effect. It is clear that every sequent may be expressed as a clause and vice versa and, in particular, every atomic clause can be expressed equivalently as atomic sequent.

1.1.2 The Language of First-Order Logic

In what follows we will consider only first-order logic, with quantifiers binding only individual variables. We do not treat terms in their full variety with functions and definite descriptions. Thus the basic quantificational language with identity ($\mathbf{L_{QLI}}$[2]) is built as follows:

1. Vocabulary of $\mathbf{L_{QLI}}$ consists of the set of connectives and:
    - denumerable set of individual variables $VAR = \{x, y, z, ...\}$
    - denumerable set of individual constants $CON = \{a, b, c, ...\}$
    - den. set of predicate symbols of n-arity $PRED = \{A, B, C, ...\}$

[2] Sometimes the reduct of this language without $=$ will be considered called $\mathbf{L_{QL}}$.

- first-order quantifiers and identity predicate: $\forall, \exists, =$

2. The set of terms $TERM$ is the union of VAR and CON. Sometimes we will restrict this set to variables only; such a version of quantificational language will be called pure.

3. Simple formulae (atoms) of $\mathbf{L_{QLI}}$ are defined by the clause:

   - if P is n-ary predicate, then $P\tau_1 \ldots \tau_n \in FOR$, where the list $\tau_1 \ldots \tau_n$ consists of n (not necessarily different) terms.

4. Compound formulae are built as in the propositional case (including bracketing conventions) with addition of the following clause:

   - if $\varphi \in FOR$, then for any variable x, $\forall x\varphi, \exists x\varphi \in FOR$; φ is the scope of a quantifier.

Note that propositional symbols are treated as predicate symbols of arity 0. For binary constant predicate $=$ we apply usual convention writing $\tau_1 = \tau_2$ instead of $= \tau_1\tau_2$. We assume that quantifiers bind their arguments with the same strength as negation.

An occurrence of a variable x in the scope of $\forall x$ or $\exists x$ is *bound*, otherwise it is *free*. A variable is bound (free) in a formula if it has at least one bound (free) occurrence in that formula. Note that a variable may be both free and bound in the same formula. $\varphi(x)$ denotes a formula with free variable x (containing at least one free occurrence of x), $VF(\varphi)$ ($VF(\Gamma)$) denotes the set of all free variables of formula φ (the set Γ). If $VF(\varphi) = \varnothing$, then φ is called a sentence (or closed formula), otherwise it is an open formula.

$\varphi[x/\tau]$ denotes the result of *proper substitution* of the term τ for x. It means that all free occurrences of x are replaced by τ and, in case $\tau \in VAR$, that all new occurrences of τ are also free in φ.

$\varphi[\tau_1//\tau_2]$ denotes a result of *replacement* of τ_1 by τ_2. Replacement is not bound to variables but may be performed on any terms, and is not bound to all but may be performed on some chosen (free, if $\tau_1 \in VAR$) occurrences.

Remark 1.1 We could avoid some complications connected with the definition of proper substitution if we follow Gentzen and introduce additional category of terms, called *parameters* (or quasi-names, or arbitrary names). In such versions of first-order languages, there is no open formulae in the sense given above. Semantical status of parameters is ambiguous; some authors treat them as another sort of variables with only free occurrence, and some use just individual constants as parameters (cf. Garson [105]). We will follow the latter solution when presenting deductive systems with rules operating on parameters. ♣[3]

Once again, after Smullyan [261] we may extend uniform notation for denoting the following types of formulae:

γ	δ	$\gamma(\tau)$ and $\delta(\tau)$
$\forall x\varphi$	$\exists x\varphi$	$\varphi[x/\tau]$
$\neg\exists x\varphi$	$\neg\forall x\varphi$	$\neg\varphi[x/\tau]$

Following the above convention we will often divide compound formulae on γ- (or universal) and δ- (or existential) formulae.

All the notions from the preceding subsection concerning formulae and their sets in propositional languages are easily redefined for first-order language, in particular atomic formulae and their negations are counted as literals.

A classical version of first-order logic will be called briefly **CQL** (without identity) or **CQLI**. But perhaps it is not the best first-order logic.

1.1.3 Some Reasons for Introducing FQL

Considerations on first-order modal logics have shown that there are several problems with satisfactory combination of quantifiers and modality. As we will see, these problems have lead either to changes in quantifier behavior or to weakening of modal apparatus. But even without troubles raised by modal constants we have numerous problems with **CQL** and **CQLI** applied to formalization of natural language that lead to introduction of some nonclassical forms. Let us look briefly at some important questions; for more detailed discussion one should consult e.g. Garson [105] or Fitting [96].

First of all: a classical **QL** forces very narrow understanding of the category of names. Nondenoting terms are not considered as possible values of individual constants.

Second, quantifiers have existential import which leads to many uncomfortable consequences. Below we list some examples:

- Sentences like "Some things do not exist.", "Something exists." cannot be formalized at all.

[3] We will use ♣ to signal the end of the remark.

- Some theses of **CQL** like $Aa \to \exists xAx$, $\forall xAx \to Aa$ also force names to yield existential import, and their consequence $\forall xAx \to \exists xAx$ excludes empty domains in models.
- In **CQLI** $\exists x(x = a)$ forces existence of any objects (e.g. God), on the other hand, a formalization of any sentence about nonexistence of some fictional object leads to contradiction.

There are several strategies of dealing with these problems. Following Garson [105] we recall the most important:

1. One may treat names that have uncertain status with respect to denotation as predicates. Sentences claiming either existence or nonexistence of some object, e.g. God, may be formulated ($\exists xGx, \neg\exists xGx$) and they are neither theses nor contradictories.
2. Russell's theory of definite descriptions may be used to eliminate such troublesome names.
3. We may change a traditional interpretation of quantifiers thus abandoning existential import.
4. Finally, we may leave **CQL** and introduce free logic **FQL**

In our opinion the last choice is the best because strategies 1–3 suffer from serious drawbacks. One of them is practical. In case of strategy 1 or 2 we obtain for many sentences rather complicated formalizations. Moreover, sentences like "Some things do not exist.", "Something exists." still cannot be formalized.

There is also some theoretical problem. There are no criteria for deciding which names should be treated as terms and which as predicates or eliminable descriptions. Consequently, it may lead to total elimination of names.

The last problem has a logical character; in case of strategies 1 and 2, either leads to some unnatural inferences. For example from the sentence "Pegasus does not exist." we may draw the following logical consequences:

- In strategy 1, it follows that "Pegasus is winged." but also "Pegasus is a lizard.", since $\neg\exists xPx \models \forall x(Px \to Ax)$ for any A.
- In strategy 2, it follows that "Pegasus is not a lizard." but also that "Pegasus is not winged.", since $\neg\exists xPx \models \neg\exists x(Px \wedge \forall y(Py \to x = y) \wedge Ax)$

It seems that solution 3 works better. Assume that we read $\exists x$ as "for some possible x", $\forall x$ – "for any possible x". Many problems disappear; e.g. we may use some empty names since $\forall x Ax \rightarrow Aa$ does not imply an existence of the designate of a, thesis $\exists x(x = a)$ states only the possibility of existence of this designate. Moreover, if we add existence predicate E, then sentences like "Some things do not exist.", "Something exists." may be formalized as: $\exists x \neg Ex$, $\exists x Ex$

But some doubts may be still raised against this solution. First of all, such a reading is not compatible with Quinean tradition, in particular with his famous criterion of existence. Also addition of existence predicate is against strong philosophical tradition (e.g. Thomist criticism of ontological proof). Also, many philosophers or logicians (like Quine) have objections to the acceptability of possibilia, and even those who do accept them, may still rise some objections. Note, that in this approach we do not claim that there exist designates of any name, but we still obtain that they (e.g. God) possibly exist. Some philosophers maintain that it should be rather proved, not assumed. On the other hand, philosophers believing in rich universe (like Meinong) might be dissatisfied because names of contradictory objects still cannot be treated as terms.

The last solution is in a sense the most radical since we abandon some very characteristic theses of **CQL**. The rules of alternative logic are more complicated, but in a sense more natural, particularly in modal setting. The most important feature of **FQL** is that existential import is saved for quantifiers but not for names. As a result $\forall x Ax \rightarrow Aa$, $\exists x(x = a)$ are not theses. It allows more natural formalization of natural language sentences – we may treat every name as a term and read quantifiers in a traditional way. Also in free logic with identity (**FQLI**) existence predicate may be defined:

$$E\tau := \exists x(x = \tau) \tag{1.3}$$

In a version without identity we will use it as an additional primitive constant.

1.1.4 Formalization of CQLI and FQLI

In this place we recall the basic information concerning semantics of classical and free logic and their axiomatic formalization. In case of free logic we restrict the presentation to *positive free logic* and the kind of semantics which is minimal modification of classical semantics and closely related to the solution applied in modal **QL**.[4]

Models

First-order models are structures of the form $\mathfrak{M} = \langle D, V\rangle$, where D is a nonempty domain and V is an interpretation of nonlogical constants defined as follows:

- $V(c) \in D$, for any individual constant c
- $V(P^n) \subseteq D^n$, for any $n-$ary predicate P^n

An assignment of variables a is a function $a : VAR \longrightarrow D$. By a^x_o we will denote an $x-$variant of a, i.e. an assignment which is like a but with object o from D specified as a value of x.

An interpretation I of the term τ in a model and under an assignment is defined as follows:

$$I(\tau) := \begin{cases} a(\tau) & \text{if } \tau \in VAR \\ V(\tau) & \text{if } \tau \in CON \end{cases}$$

The key notion of satisfaction of a formula in a model under an assignment is defined as follows:

$$\begin{array}{lll}
\mathfrak{M}, a \vDash P^n(\tau_1 ... \tau_n) & \text{iff} & \langle I(\tau_1), ..., I(\tau_n)\rangle \in V(P^n) \\
\mathfrak{M}, a \vDash \neg\varphi & \text{iff} & \mathfrak{M}, a \nvDash \varphi \\
\mathfrak{M}, a \vDash \varphi \wedge \psi & \text{iff} & \mathfrak{M}, a \vDash \varphi \text{ and } \mathfrak{M}, a \vDash \psi \\
\mathfrak{M}, a \vDash \varphi \vee \psi & \text{iff} & \mathfrak{M}, a \vDash \varphi \text{ or } \mathfrak{M}, a \vDash \psi \\
\mathfrak{M}, a \vDash \varphi \rightarrow \psi & \text{iff} & \mathfrak{M}, a \nvDash \varphi \text{ or } \mathfrak{M}, a \vDash \psi \\
\mathfrak{M}, a \vDash \tau_1 = \tau_2 & \text{iff} & I(\tau_1) = I(\tau_2) \\
\mathfrak{M}, a \vDash \forall x\varphi & \text{iff} & \mathfrak{M}, a^x_o \vDash \varphi \text{ for all } o \in D \\
\mathfrak{M}, a \vDash \exists x\varphi & \text{iff} & \mathfrak{M}, a^x_o \vDash \varphi \text{ for some } o \in D
\end{array}$$

φ is satisfiable (generally **L**-satisfiable, for any considered logic **L**) if there is a model and an assignment that satisfies φ, otherwise it is unsatisfiable. $\models \varphi$ means that φ is satisfied by all models and assignments, hence it is a classical tautology (or classically valid formula). Note that if x is not free in φ, then $\mathfrak{M}, a \vDash \varphi$ iff $\mathfrak{M}, a^x_o \vDash \varphi$, for any x-wariant of a.

[4]For more information on different versions of free logic and their semantics consult e.g. Bencivenga [25].

In order to obtain a semantics for **FQL** we introduce the so called *inner domain – outer domain* models. In this case a domain of a model is the union of two disjoint and possibly empty sets: D_o – outer domain, and D_i – inner domain. The latter set is interpreted as the set of all existing objects, whereas the former is the set of all nonexistent (possible) objects. All other details of characterization put above for classical case are left intact except two clauses for satisfaction of quantified formulae. In these models they are defined not for the whole domain but for D_i, i.e.

$$\mathfrak{M}, a \vDash \forall x\varphi \quad \text{iff} \quad \mathfrak{M}, a^x_o \vDash \varphi \text{ for all } o \in D_i$$
$$\mathfrak{M}, a \vDash \exists x\varphi \quad \text{iff} \quad \mathfrak{M}, a^x_o \vDash \varphi \text{ for some } o \in D_i$$

In case we do not have identity but there is E as a primitive predicate constant we add a clause:

$$\mathfrak{M}, a \vDash E\tau \quad \text{iff} \quad I(\tau) \in D_i$$

The definition of satisfiability and validity in free logic is the same as in classical case. But still we have two possible variants. Semantics with models admitting empty D_i is adequate for *universally free logic*,[5] where e.g. $\forall x\varphi \rightarrow \exists x\varphi$ is not tautological. If we restrict the class of models to those with nonempty D_i we obtain a stronger variant of free logic, where the above formula is a tautology.

The notion of entailment (or consequence relation) for all the logics described above may be reduced to the notion of validity as follows:

Definition 1.3 (Entailment)

$\Gamma \models_L \varphi$ iff $\models_L \psi_1 \wedge .... \wedge \psi_n \rightarrow \varphi$, where $\{\psi_1, ..., \psi_n\} \subseteq \Gamma$ and **L** is classical or free logic.

Axiomatizations

The earliest, and still the most popular, syntactic style of defining logics was axiomatic, especially in the form provided by Hilbert. In what follows such systems are called H-systems. Although we do not aim to focus on axiomatic formalizations and their properties, it is handy to recall some systems in order to show clearly the differences between classical and free

[5] Sometimes logics of this sort are called *inclusive*, particularly in contexts where the problem of empty domains is considered separately from the problem of non-denoting terms – there are logics which are inclusive but not free.

version of **QL**. Any axiomatic (or Hilbert) formalization of the logic **L** will be denoted as H-**L**. In particular H-**CQLI** consists of:

1. Axioms of **CPL**.

Any complete set is suitable but for the sake of concreteness and for future reference we display the following:

1	$\varphi \to (\psi \to \varphi)$
2	$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$
3	$\varphi \wedge \psi \to \varphi$; and $\varphi \wedge \psi \to \psi$
4	$\varphi \to (\psi \to \varphi \wedge \psi)$
5	$\varphi \to \varphi \vee \psi$; and $\psi \to \varphi \vee \psi$
6	$(\varphi \to \chi) \to ((\psi \to \chi) \to (\varphi \vee \psi \to \chi))$
7	$(\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$

2. Axioms characterizing quantifiers and identity:

$\forall E$	$\forall x\varphi \to \varphi[x/\tau]$
$\exists I$	$\varphi[x/\tau] \to \exists x\varphi$
ID	$\tau = \tau$
LL	$\tau_1 = \tau_2 \to (\varphi \to \varphi[\tau_1//\tau_2])$, where φ is atomic

3. Rules:

(MP)	$\vdash \varphi \to \psi, \vdash \varphi \ / \ \vdash \psi$
$(\forall)$	$\vdash \varphi \to \psi(x) \ / \ \vdash \varphi \to \forall x\psi$, where $x \notin VF(\varphi)$
$(\exists)$	$\vdash \varphi(x) \to \psi \ / \ \vdash \exists x\varphi \to \psi$, where $x \notin VF(\psi)$

Note that since we use axiom schemata instead of specific formulae, we do not need to add substitution rule.

Some explanation is needed concerning the usage of $\vdash$. In general, $\vdash \varphi$ applied in schemata of rules means that φ is a thesis of respective system i.e. φ has a proof being a sequence of formulae deduced from axioms with the help of the rules. $\vdash \Gamma$ means that all formulae in Γ are theses of the system. Usually the usage of $\vdash$ with no decorations is convenient and sufficient if the system and the logic under consideration is known from the context or if it is irrelevant. If such an information is essential we will be using $\vdash_L$ (or even $\vdash_{\text{DS-L}}$) to denote that it is a thesis of logic **L** (or a thesis of **L** in deductive system DS). Thus in case of the above rules we should formally use $\vdash_{\text{H-CQLI}}$ to supply the exact information.

In what follows the set of all theses of any logic **L** will be denoted by Th(**L**). $\nvdash_L \varphi$ means that φ is not a thesis of **L**. The relation of deducibility (provability) and inconsistency/consistency may be defined as follows:

Definition 1.4 (Deducibility, consistency)

- $\Gamma \vdash_L \varphi$ iff $\vdash_L \psi_1 \wedge .... \wedge \psi_n \to \varphi$, where $\{\psi_1, ..., \psi_n\} \subseteq \Gamma$
- Γ is **L**-inconsistent iff $\Gamma \vdash_L \bot$, otherwise it is **L**-consistent.

In order to get an axiomatization of **CQL** we should get rid of two axioms for =. On the other hand, for an axiomatization of universally free logic we must replace axioms and rules for quantifiers in H-**CQL** (or H-**CQLI**) with the following weaker versions[6]:

$F\forall E \quad \forall x\varphi \to (E\tau \to \varphi[x/\tau])$
$F\exists I \quad \varphi[x/\tau] \to (E\tau \to \exists x\varphi)$
$(F\forall) \quad \vdash \varphi \to (Ex \to \psi(x)) \ / \ \vdash \varphi \to \forall x\psi$, where $x \notin VF(\varphi)$
$(F\exists) \quad \vdash \varphi(x) \to (Ex \to \psi) \ / \ \vdash \exists x\varphi \to \psi$, where $x \notin VF(\psi)$

With a view of strengthening the system so that the logic of nonempty (inner) domain we must add one more axiom:

$$\exists x Ex \tag{1.4}$$

Adequacy and Decidability

The systems discussed above provide adequate formalizations of respective logics. We recall that the link between H-systems (and syntactic formalizations in general) and suitable classes of models is obtained via soundness and completeness theorems of the form:

- (Soundness) if $\Gamma \vdash_L \varphi$, then $\Gamma \models_L \varphi$
- (Completeness) if $\Gamma \models_L \varphi$, then $\Gamma \vdash_L \varphi$

The last one is often formulated equivalently:

- if Γ is **L**-consistent, then Γ is **L**-satisfiable

[6]We assume that E is present in a language as primitive or definable by =, otherwise the axiomatization is slightly more complicated.

If the first theorem holds, then the system is sound with respect to the suitable class of models; if the second holds, then it is (strongly) complete. It is adequate if it is both sound and complete. Note that if Γ is empty (in the first formulation) or finite, we have weak completeness, otherwise we have a strong form (i.e. admitting infinite Γ).

Standard proofs of completeness apply the well known construction of a model due to Henkin (based on the earlier Lindenbaum result). Essentially it consists in showing that there is a unique infinite model that falsifies every formula unprovable in **L**. This method of proof is therefore not constructive. For questions of decidability and automated theorem proving it is more important that for many logics we will consider there are constructive methods of proving completeness. They show how to find for any unprovable formula some finite falsifying model. We shall be concerned with them in Chapters 4 and 10.

CPL is decidable logic, in the sense that there is an effective procedure which, for every $\varphi \in \mathbf{L_{CPL}}$, resolves whether it is a tautology of this logic or not. This is is a special form od decision problem generally called *validity problem* for a logic **L**. Very often decidability of **L** is posed as the so called *satisfiability problem* (or shortly, sat-problem) for **L**: given $\varphi \in \mathbf{L}$, decide whether φ is **L**-satisfiable. Clearly, these instances of decision problem are complementary since, for every logic we will be dealing with, it holds that $\models \varphi$ iff $\neg\varphi$ is not **L**-satisfiable. Hence, for any logic, validity problem is decidable iff sat-problem is decidable.

The notion of effective procedure, or an algorithm, used in the characterisation of decidability needs some explanation. There are plenty of formal explications of it, in terms of Turing machines, Markov algorithms, recursive functions, e.t.c, to mention the oldest and the most popular. Advanced investigations on these mathematical models of effectiveness form the core of computability theory. We do not need in this book any formal treatment of these matters however. Informally, it must be a method which is mechanical (it works without any need for ingenuity), fair (it does what it is assumed to do for every input) and terminating (it works in finite time). On the other hand, in all mathematical models of effective procedure no real bounds are put on the time of performance or amount of memory needed to store the data. Hence effective is not the same as efficient. Investigations on practical (time and space) requirements of algorithms, and classification of decidable problems belong to complexity theory; we will say a bit more about it in Chapter 5.

First-order logic is not decidable but it is semi-decidable.[7] It means that there are procedures which applied to a thesis give you an answer "yes" in finite time. But if tested formula is not a thesis we have no guarantee that a procedure stops with negative answer. It is important that in practise it does not preclude the automatization of a proof search. Moreover, there are numerous fragments of first-order logic which are not only decidable but even very efficient, in the sense of having very fast decision procedures (e.g. the logic of Horn clauses).

1.1.5 Important Derived Notions

First-order language allows us to introduce precisely some notions concerning relations and special first-order theories needed in the sequel.

Definition 1.5 (Closures of relations) Let $\mathcal{R}$ be any binary relation on a nonempty set A, then:

- $\mathcal{R}^+$ is a *transitive closure of* $\mathcal{R}$ iff, $\mathcal{R}^+$ is the smallest transitive relation on A such that $\mathcal{R} \subseteq \mathcal{R}^+$;
- $\mathcal{R}^*$ is a *reflexive transitive closure of* $\mathcal{R}$ iff, $\mathcal{R}^*$ is the smallest reflexive and transitive relation on A such that $\mathcal{R} \subseteq \mathcal{R}^*$

Recall that a binary relation $\mathcal{R}$ on a nonempty set A is a set of ordered pairs from A^2. In particular: $\mathcal{R}$ is transitive on A iff for every x, y, z from A, $\mathcal{R}xy$ and $\mathcal{R}yz$ implies $\mathcal{R}xz$, and reflexive iff $\mathcal{R}xx$ for every $x \in A$. A relation which is both reflexive and transitive is a relation of *quasi order* (on some set A). It is an *equivalence* relation if additionaly it is symmetric i.e. if $\mathcal{R}xy$ implies $\mathcal{R}yx$ for all x, y, z from A. In case of equivalence relation $\mathcal{R}$ we may obtain a division of A on nonempty and mutually exclusive subsets (classes of abstraction from $\mathcal{R}$) containing exactly these elements of A which are related to each other. The hierarchy of ordering relations is also built up from quasi orders but by the addition of other properties. $\mathcal{R}$ is a *partial order* on A iff it is antisymmetric quasi order, i.e. if for any different x, y, $\mathcal{R}xy$ implies $\neg\mathcal{R}yx$. We obtain a *linear order* if $\mathcal{R}$ is additionaly dichotomic,

[7]More formally, the set of tautologies is not recursive but still it is recursively enumerable. In fact, undecidability of **QL** follows from the so called Church-Turing thesis which is not provable but commonly believed to be true. It is a claim that the set of computable functions coincides with the set of functions computable on any of the proposed mathematical models of computation, like Turing machines.

i.e. either $\mathcal{R}xy$ or $\mathcal{R}yx$ for any x, y in A. There is a parallel hierarchy of *strict orders*, where suitable relation is transitive but irreflexive, i.e. $\mathcal{R}xx$ holds for no $x \in A$. These two properties imply that a relation is also asymmetric in the sense that $\mathcal{R}xy$ always implies $\neg\mathcal{R}yx$. These properties of a relation define the basic kind of *strict partial order*; we obtain a *strict linear order* if it holds also trichotomy, i.e. either $\mathcal{R}xy$ or $\mathcal{R}yx$ or $x = y$ for any x, y in A.

We will need also some binary operations on relations; ordinary set theoretical union $\cup$ and specifically relational composition $\circ$ defined as follows:

Definition 1.6 (Operations on relations)

- $\mathcal{R}_1 \cup \mathcal{R}_2 xy := \mathcal{R}_1 xy \vee \mathcal{R}_2 xy$
- $\mathcal{R}_1 \circ \mathcal{R}_2 xy := \exists z(\mathcal{R}_1 xz \wedge \mathcal{R}_2 zy)$

In particular, $\mathcal{R}^m xy$ denotes $m-1$-ary application of $\circ$ to $\mathcal{R}$, or the distance between x and y, thus $\mathcal{R}^2 xy$ is $\mathcal{R} \circ \mathcal{R}xy$, whereas $m = 1$ is just $\mathcal{R}xy$ and $m = 0$ means that $x = y$.

For further considerations the notion of a tree is of a basic importance. It is characterized in several ways, not always equivalent, in mathematics, computer science, logic. For our purposes we follow the definition from [35].

Definition 1.7 (Tree) $\mathfrak{T}$ is called a *tree* if it is a relational structure $\langle \mathcal{T}, \mathcal{R} \rangle$, such that:

- there is a unique element $r \in \mathcal{T}$, called *root* such that $\forall t \in \mathcal{T}, \mathcal{R}^* rt$;
- every element $t \in \mathcal{T}$ distinct from r has a unique predecessor i.e. there is only one $t' \in \mathcal{T}$, such that $\mathcal{R}t't$;
- $\mathcal{R}$ is acyclic, i.e. $\forall t \in \mathcal{T}$ it is not true that $\mathcal{R}^+ tt$.

All elements of $\mathcal{T}$ are called *nodes*, for every pair t, t' such that $\mathcal{R}tt'$, t is called *the parent* and t' – *a child*; if $\mathcal{R}^+ tt'$, then t is *an ancestor* and t' is *a successor*. Every node t with no children is called *a leaf*.

N-ary sequence $\langle t_1, \ ..., \ t_n \rangle$, where for each $i < n$ we have $\mathcal{R}t_i t_{i+1}$ is called *a path*; every maximal path is *a branch* (in finite case it is a sequence from the root to a leaf). A path, where every node (except possibly the last) has only one child is called *a segment.*

The number of children of a node is *the branching factor* of this node; the branching factor of a tree is the biggest branching factor of its nodes. If the branching factor of a tree is a natural number we have *finitely generated tree*; trees with nodes having at most two children are called *binary trees.*

In what follows we will meet trees in several places, e.g. proof-trees, trees of proof-search, tree-models. Very useful for our needs are finitely generated trees, binary trees in particular. Below we recall some important result concerning such trees – the König Lemma.

Lemma 1.1 (König) *Every finitely generated but infinite tree has at least one infinite branch.*

By the König Lemma, to show that a tree is finite it is sufficient to show that it is finitely generated and that every branch is finite.

Finally, we define three types of first-order theories characterized by formulae of specific form. They will be of particular importance for relational semantics of wide classes of modal logics.

The notion of Horn clause naturally extends to first-order language. It is any sentence of the form:

$$(hc) \quad \forall x_1, ..., x_k(\varphi_1 \wedge ... \wedge \varphi_n \rightarrow \psi),$$

where $k \geq 1, n \geq 0$, each φ_i and ψ is an atom.

Horn theory is a finite set of Horn clauses. Clearly, the scope of application of Horn theories depends on what we count as atoms. Inclusion of $\perp$ and identities in the set of atoms yields a significant generalization. Horn clauses without such atoms (only predicate letters with terms) will be called strict.

A direct generalization of Horn clause is that of a universal implication of the form:

$$(ui) \quad \forall x_1, ..., x_k(\varphi_1 \wedge ... \wedge \varphi_n \rightarrow \psi_1 \vee ... \vee \psi_m),$$

where $k \geq 1, n, m \geq 0$, each φ_i and each ψ_i is an atom ($m = 0$ is interpreted as $\perp$).

A *universal theory* consists of a finite set of such formulae. In fact, every universal theory may be reduced to $\forall x_1, ..., x_k \varphi$, where φ is in CNF, since every implication of atoms is equivalent to ordinary clause, finite set of clauses is equivalent to their conjunction and $\forall$ distribute over $\wedge$. It is obvious that every Horn clause is universal implication, provided we keep the same specification of atoms.

The most general class of first-order theories we encounter for future use is the class of the so called *geometric theories* (see [286]). A first-order formula is geometric if it is built up from atoms of the form Rxy or $x = y$ with the help of $\bot, \wedge, \vee$ and $\exists$, only. A geometric theory is a finite set of first-order sentences of the form:

$\forall x_1, ..., x_k(\varphi \rightarrow \psi)$, where φ and ψ are geometric formulae.

For better understanding in what way the notion of geometric formula generalizes that of universal implication it is convenient to introduce the concept of basic geometric formula. Simpson [257] has proved that each geometric theory is equivalent to *basic geometric theory*, where each formula has the form:

$$(bgf) \quad \forall x_1, ..., x_k(\varphi_1 \wedge ... \wedge \varphi_n \rightarrow \exists y_1, ..., y_l(\psi_1 \vee ... \vee \psi_m)),$$

where $k \geq 1, l, n, m \geq 0$, each φ_i is an atom and each ψ_i is an atom or a finite conjunction of atoms.

To appreciate the generality of these notions let us note that such important mathematical theories like theory (classical and constructive) of projective geometry, ordered fields or Robinson's arithmetic may be axiomatized as geometric theories.

1.2 Deductive Systems, Rules, Proofs

1.2.1 Deductive Systems

In subsequent chapters we will discuss not only ND systems but many other types of formalizations of logics. Since we are interested in their comparison, in the ways of combining them, and the transfer of results, it is crucial to establish some form of unified description. It is not an easy task; existing systems differ in many ways and sometimes even essentially the same (in some respect) systems are presented by several authors by means of very different tools. The invention of logicians in creating new forms of presentation is amazing; in this respect they are quite similar to artists always ready to break the existing boundaries. On the other hand, systematic reflection on the general schema of formalization is rather poor and often not general enough to cover all the existing systems.

In this book we use the term *deductive system* (DS in short) not in the sense of Tarski (i.e. as a set of formulae closed under some operation of consequence), but as a name for several formalizations of logics. Every DS may be characterized on two elementary levels of description:

- *the calculus*, being a theoretical description of the set of primitive rules;
- *the realization*, where we describe a practical form of application of a calculus.

Remark 1.2 The distinction between the calculus and its realization is essential because these two levels are independent. For example, two different axiomatizations of the same logic differ with respect to calculus (sets of axioms and rules) but usually not with respect to realization, since the form of defining and setting down a proof in Hilbertian proof theory is rather standard. On the other hand, logic textbooks present many different realizations of essentially the same (with respect to calculus) ND system.

Usually these two levels are not distinguished but for our future considerations this distinction is important. In fact, the level of realization is rather neglected by logicians; they simply use some form of realization but they do not take this fact under consideration. The rare exception to this custom may be found in Appendix 2 of Prawitz [220], where some remarks on several forms of realization of ND systems are spelled out. ♣

1.2.2 Calculus

A calculus is a nonempty and finite set of schemata of rules of the shape:

$$X_1,\ ...,\ X_k\ /\ Y_1,\ ...,\ Y_n \qquad k \geq 0, n \geq 1 \tag{1.5}$$

with possible list of side conditions. Symbols X_i denote some *data structures*, which are transformed (by the rule) into data structures Y_j. Which type of data structures occurs in rules depends on the type of a system; there may be single formulae, sets of formulae, sequents, labelled formulae e.t.c. The notion of a rule we admit is very general but we assume that every rule is decidable, in the sense that for any concrete sequence of data structures it may be established in finite time whether it is an instance of a rule.

Usually we will abandon the set-theoretical notation in the description of rules, e.g. the scheme of a rule modus ponens (MP) will be written as $\varphi, \varphi \rightarrow \psi / \psi$, not as $\{\varphi, \varphi \rightarrow \psi\}/\psi$. Sometimes it may cause some misunderstandings, particularly in rules where the sets of formulae are single data structures. In this case we will apply " , " to separate formulae in some set, and " ; " to separate sets, e.g.: $\varphi, \psi \ ; \ \chi$ will denote $\{\varphi, \psi\}, \{\chi\}$. To display *invertible rules* in some DS we will apply symbol //; e.g. Γ // Δ means that, in a system we have in fact two rules: Γ / Δ and Δ / Γ.

Remark 1.3 Our concept of a calculus is somewhat different from more traditional understanding like in Church [70] or Wójcicki [285], where this term corresponds to logics (syntactically understood) not to their formalizations. Calculus in the sense of Church is a quadruple containing a language, a set of formulae, a set of axioms, a set of rules; in the sense of Wójcicki it is a pair: a language and a consequence operation. Such concept of a calculus is more general than ours – since we do not put specification of a language in a calculus – but, in some other respect, too narrow for our purposes – since, in principle, it is suitable only for axiomatic systems.

In fact, our concept of a calculus is closer to what Wójcicki describes as a *deductive base* – a triple consisting of the set of formulae (axioms), sequents (H-rules from Hilbert) and sequent rules (G-rules from Gentzen). It is easy to note that all elements of any deductive base are representable as elements of a calculus in our sense: axioms are represented as rules of the form $\varnothing/\varphi$, H-rules as Γ / φ and G-rules as $\Gamma_1 \Rightarrow \varphi_1, ..., \Gamma_k \Rightarrow \varphi_k$ / $\Delta \Rightarrow \psi$. In fact, our concept of a calculus is still a generalization of the concept of a deductive base since we admit different forms of data structures as values of X_i, Y_j. ♣

1.2.3 Realization

In several DS's we deal with a variety of forms in which a calculus may be realized in deductive practice. Usually it is a set of instructions of how to build a proof/derivation and apply rules, sometimes with additional technical devices. In our usage a derivation is a more general term than a proof. There are systems admitting only proofs of theses or deduction of consequences from assumptions (e.g. axiomatic systems or standard ND); for them a definition of a proof is sufficient. But there are also systems allowing disproofs i.e. deductive falsification of non-theses (e.g. tableau systems); such systems require a definition of derivation. Sometimes refutational calculi are distinguished (e.g. tableaux, resolution), where instead of a proof of φ we attempt to refute $\neg\varphi$, i.e. to derive $\bot$ from it. Clearly a succesfull refutation is just a (indirect) proof of φ, whereas failed (but somewhat completed), refutation is a disproof of φ.

A proof (derivation) may have tree or linear structure. We focus on this subject later on (cf. Chapter 2) but notice that it is not only a difference of presentation. In linear proofs (L-proofs for short) we use data structures (formulae, sequents e.t.c.), whereas in tree-proofs (T-proofs) we manipulate on their concrete occurrences labelling particular nodes of the tree. It is of great importance, for example, in finding suitable realization for substructural logics, but even in case of classical logic it makes some difference – as we shall see in Section 2.4. – at least for ND systems. In what follows we will divide DS's on T- and L-systems according to the form of a proof/derivation. In case of systems applying T-proofs we will often use vertical notation already on the level of calculus in order to graphically highlight the form of realization. So in case of ordinary trees with root at the bottom instead of $X, Y \;/\; Z$ we will write:

$$\frac{X \quad Y}{Z}$$

In case of inverted trees (root at the top) instead of $X/Y, Z$ we will write:

$$\frac{X}{Y \mid Z} \qquad \text{or} \qquad X \;/\; Y \;\mid\; Z$$

This convention in many places will allow of shorter description of DS's by simply displaying the elements of a calculus without any detailed exposition of its realization. But in case of ND systems we must focus on forms of realization since they modify seriously the character of the system. For example, we will obtain (in Chapter 4) two different ND systems on the basis of the same calculus by putting suitable constraints on the application of rules in the definition of a derivation. So, any precise definition of a derivation added to a calculus and leading to full description of DS is, in a sense, a restriction of a calculus.

The distinction between T- and L-systems in case of ND leads to some additional complications as well. In some ND-systems that will be of special interest for us, L-proofs have more complex form. It is connected with the possibility of introduction of additional assumptions in the course of proof. These assumptions initiate subordinate proofs (subproofs) embedded in the main proof which must be closed at some stage. As a result, instead of a sequence of formulae in such L-proofs, we have a sequence of nested (sub)sequences of formulae. On the level of calculus we will describe the ways of introducing and closing subproofs together, as complex *rules of proof construction* of the shape:

$$If\ \ X_1 \vdash Y_1, ..., X_k \vdash Y_k,\ then\ \ Z \vdash W \tag{1.6}$$

So, theoretical characterization of such ND system (i.e. as a calculus) contains two nonempty sets of rules: the *rules of inference* of the shape X/Y and the rules of proof construction regulating the process of subproof generation.[8]

Calculi and their realizations do not provide an exhaustive way of analyzing formalizations of logics, especially if we are interested in their practical applications. The next possible step in constructing DS (the next level of description) is to devise some algorithms for proof search; decision procedures in particular. It is more strict characterization of the application of DS than description of realization, since in the definition of procedures we often put severe restrictions on the applicability of rules. This level of description makes possible an implementation of the system for the needs of automated theorem proving. Sometimes it is in fact a revision of underlying DS, where suitable constraints are formulated on the level of calculus or its realization. For example, in some types of DS's devised originally for automated theorem proving (cf. the presentation of Davis/Putnam procedure in Chapter 3) it is difficult to extract a pure calculus and its realization from proof search procedure. In this work we do not treat this aspect in a systematic way, but (e.g. in Chapters 4 and 10) we provide some procedures suitable for some types of ND systems. These are devised mainly for the need of constructive completeness proofs, and we do not aim to formulate definite claims on their efficiency. Further work in this direction would need an implementation of these solutions and empirical testing of the performance.

[8]It may seem at first sight that the separation of these two types of rules in some kind of ND systems is incoherent with our general characterization of a calculus. But it is apparent, because every proof construction rule of the form (1.6) may be also presented as an instance of a rule (1.5) with sequents $X_i \Rightarrow Y_i$ as premises and $Z \Rightarrow W$ as (the only) conclusion. It is just a notational convention for making clear the difference between such rules in ND L-systems using formulae and ND systems using sequents, where separation of proof construction rules does not make sense, cf. Section 2.3.

1.2.4 Extensions and Simulations

Usually to characterize a system we need a list of primitive rules. From the theoretical point of view it is a sufficient description of the calculus, but – at least in case of ND systems – we will consider also many secondary rules. In general, they are of two types:

Definition 1.8 (Secondary Rules) For any DS and logic **L** let $\vdash_{DS\text{-}L}$ mean some deducibility relation constituted by primitive rules of the system, then:

1. a rule of the schema $X_1, ..., X_k / Y_1, ..., Y_n$ is *DS*-**L**-*derivable* iff, $X_1, ..., X_k \vdash_{DS-L} Y_1, ..., Y_n$;
2. a rule (r) is *DS*-**L**-*admissible* iff, $[X_1, ..., X_k \vdash_{DS\text{-}L} Y_1, ..., Y_n$ iff, $X_1, ..., X_k \vdash_{DS'\text{-}L} Y_1, ..., Y_n]$, where DS' is DS with added rule (r).

The set of DS-**L**-derivable rules is denoted by *DER(DS*-**L**), and the set of DS-**L**-admissible rules by *ADM(DS*-**L***)*.

It is easy to show that for any logic **L**, $DER(DS\text{-}\mathbf{L}) \subseteq ADM(DS\text{-}\mathbf{L})$; the reverse inclusion holds only for logics having structurally complete formalizations (e.g. many axiomatic systems for **CPL**). To show that some rule is derivable in a system it is sufficient to build a scheme of its proof. In consequence, the set of derivable rules for a formalization of some logic is preserved for every extension of this logic in the system of the same sort, i.e. $DER(DS\text{-}\mathbf{L}) \subseteq DER(DS'\text{-}\mathbf{L}')$, where $\mathbf{L}'$ is any extension of **L** and DS' is an extension of DS.

A proof of admissibility of some rule for DS-**L** is usually more complicated since it is necessary to show that every proof using (r) may be transformed into a proof where (r) is dispensable.[9] Generally, in case of admissible rules, preservation for any extension of **L** is not saved, since the addition of some new primitive rules to the respective calculus may destroy the result. Famous illustration of this problem may be provided by several sequent calculi where cut elimination theorem holds for some logic but not for its extension. For us the question will be interesting with respect to ND systems, where we will often show admissibility of some proof construction rules in order to prove that ND system may simulate proofs and proof search strategies of other systems.

[9] That's why in literature it is often said that the rule is eliminable.

The concept of a simulation of one system by the other needs some explanation.

Definition 1.9 (Simulation)

- Deductive system SD1 may simulate a system SD2 iff, there is some computable function that allows every proof of SD2 to be reconstructed in SD1.
- Deductive system SD1 may p-simulate (polynomially simulate) a system SD2 iff, a result of a simulation is bounded by some polynomial function in the length of a proof in SD2.[10]

The concept of p-simulation is particularly useful since in this case we obtain such a form of mapping, from proofs in SD2 into proofs in SD1, which shows that SD1 is, from the computational point of view, not worse than SD2. Since a relation of p-simulation is a relation of quasi-order, and its symmetric closure gives a relation of equivalence, we may obtain a systematization (i.e. linear ordering of classes of abstraction from this equivalence relation) of deductive systems with respect to their relative complexity. Systems belonging to the same class of abstraction may be treated as representing the same level of complexity (up to a polynomial). Clearly, one should remember that the only measure of system's complexity taken here into account is the length of proofs, so it is not justified to say about two systems from different classes of abstraction that the one is more efficient than the other. For example, a system with essentially longer proofs may have essentially smaller space of proof-search than a system with short proofs but requiring much ingenuity for their construction.

In fact, in many cases we may provide even smarter forms of simulation. [196] generalizes the notion of p-simulation from proofs to derivations in general and introduces the notion of step-wise simulation. The last obtains if we may show that there is some n such that for every inference step in the simulated system there is at most n inferences performed in the simulating system. In many cases we will show how to make a step-wise simulation of some deductive system in the suitable version of ND system.

---

[10] We do not introduce formally the concepts of computable functions because the questions of computability and complexity theory are not the subject of this book but cf. some elementary remarks in Section 5.5. The above informal characterization of p-simulation will do for our interests; the reader may consult e.g. D'Agostino [2] or Schmidt [242] for more information on several forms of simulation.

1.2.5 Semantical Side

Our study is not proof-theoretical in the strict sense, we will be often dealing with semantics. When presenting a variety of ND systems we will always show their completeness, although sometimes in a sketchy way. In many cases it suffices to obtain the result indirectly, by simulation of other system which is known to be complete. For example, to simulate an axiomatic system in ND system it is enough to prove all axioms and show that primitive rules of the axiomatic system are primitive or secondary in ND. Simulation of other systems in ND often requires more complicated operations (cf. e.g. Chapter 4). In case of some logics (e.g. modal logics characterized by models with linear accessibility relation) we will provide direct proofs of completeness because the proposed analytic formalizations of these logics have no counterparts in other types of DS's (at least formalizations easily simulated in ND).

Proofs of soundness of ND systems would need some semantic qualification of rules. We restrict our considerations to rules of inference of the shape $\Gamma \;/\; \Delta$ and proof construction rules of the shape: if $\Gamma \vdash \Delta$, then $\Pi \vdash \Sigma$.

Definition 1.10 (Correctness of rules)

1. a rule of inference $\Gamma \;/\; \Delta$ is **L**-normal iff, in semantics for logic **L** it holds $\Gamma \models \varphi$ for every $\varphi \in \Delta$;

2. a proof construction rule "if $\Gamma \vdash \Delta$, then $\Pi \vdash \Sigma$" is **L**-normality preserving iff, if $\Gamma \;/\; \Delta$ is **L**-normal, then $\Pi \;/\; \Sigma$ is **L**-normal.[11]

The above concepts may be generalized for rules operating on other data structures and for other relations of consequence. We will do it successively in suitable places. Here we recall that for every relation of consequence $\models$ we assume that it satisfies some structural conditions, namely:

- (ID): $X_1, \ ..., \ X_k \models X_1, \ ..., \ X_k$

- (MON): If $X_1, \ ..., \ X_k \models Y_1, \ ..., \ Y_n$, then $X_1, \ ..., \ X_k, X_{k+1} \models Y_1, \ ..., \ Y_n$

- (TR): If $X_1, \ ..., \ X_k \models Z$ and $Z, Y_1, \ ..., \ Y_n \models Y_{n+1}$,
  then $X_1, \ ..., \ X_k, Y_1, \ ..., \ Y_n \models Y_{n+1}$.

[11] This property is a generalization of the concept of **L**-validity of rules introduced for rules of inference which preserve the set of **L**-tautologies; cf. e.g. Pogorzelski [215].

These conditions have more or less direct manifestation in several deductive systems. Particularly interesting formalization in this respect is Gentzen's sequent calculus (SC), where all these properties are directly expressed by rules: $(AX), (W)$ (weakening) and (Cut) (cf. Chapter 3). The last rule is of great importance, it is present in many sorts of systems under several names and shapes, often difficult to compare. For example: in resolution systems it is a basic rule (Res), in axiomatic systems it is encoded in (MP) and as a secondary rule (HS) (hypothetical syllogism), in system KE it is a branching rule (BP) (bivalence property), and in ND systems it is encoded directly not only by variants of (MP) but also by $[RED]$ (the rule of indirect proof or reductio ad absurdum). Moreover, (TR) is not only represented directly by several rules but also indirectly involved in the process of deductive inference, where we conclude that the output follows from starting premises although it is deduced by the chain of intermediate steps. In general, we will use the name *cut* always when it is not important which particular rule of a particular system is under consideration. But we will distinguish between the rules that express (TR) in a progressive way, i.e. starting from antecedent of (TR) and going to succedent, and the rules that express (TR) in a regressive way, from succedent to antecedent. For example, (MP) of axiomatic systems, and some of the rules considered in Chapter 3, like (Cut) in SC systems, or resolution, are the examples of progressive cut; on the other hand, (PB) in KE or tableau systems, $[RED]$ in ND are the examples of regressive cut.

1.2.6 Types of Deductive Systems

We close this Chapter with some systematization of existing systems which will be the point of reference in further considerations. The general concept of a deductive system needs some specification since there is a huge number of such systems of different character and applicability. Some of them are devised for very special purposes, whereas others have quite universal character; some of them are meant as formalization of just one method of deduction, whereas others offer more freedom in choosing ways of proof construction. Even in case of ND-systems a variety of forms of realization causes a lot of problems in their systematization as we will see in the next Chapter. It is not surprising that a good classification of DS's is not an easy task. Perhaps the best we can do is some reasonable typology. But for our needs it is not necessary. With no pretension to exploit the subject we may divide the basic types of DS's into:

- Axiom systems of the Hilbert type (H-systems)
- Natural deduction (ND)
- Resolution systems
- Sequent calculi (SC)
- Tableau systems (TS)
- KE system
- Connection systems
- Refutation systems
- Davis/Putnam system (DP)
- Goal oriented systems

Using different criteria we may divide all the systems into some more extensive classes collecting some types of DS's representing family resemblance. In particular, it is important for us to distinguish between systems that are based on the application of cut and those that profit from the possibility of elimination of explicit form of this rule.

We are not going to discuss all the systems from the list, since the book is devoted to ND systems for classical and modal logic. But some of them will be important as the source of inspiration either for obtaining more general forms of ND system or for providing suitable rules for modal logics. So after the presentation of the standard form of ND we recall the basic information on some other DS's in Chapter 3. Our further treatment of several systems is not uniform, since their applicability in modal logic, and relationship to ND systems is not equal, and these are our fundamental criteria of choice.

Axiom systems are still the most prevailing way of presentation of many logics. Hence we used them in presenting classical and free logic, and we will do it in case of modal logics, as well. But it is for reference only, as axiom systems are not the subject of our consideration.

As far as the application to nonclassical logics, especially modal, is concerned, sequent and tableau systems are the most important. Other types of systems were not extended to other logics (like DP) or only in a limited way (resolution, connection, refutation systems)[12] Obviously, the paucity

[12]In case of resolution calculi this claim may be disputable. But we mean here the so called direct resolution calculi for modal logics i.e. with no use of translation to e.g. first-order logic, cf. Section 3.2.

of application represent the actual situation only and by no means implies that these systems are not good for that. Nevertheless, the actual situation is an important factor and a main reason that we will treat sequent calculi and tableaux in more detail than other kinds of systems.

On the other hand, resolution systems and Davis/Putnam system will be also reviewed, as important for automated theorem proving. This field is not particularly important for us in this work, however, if we want to show universal and general character of ND, we cannot avoid completely its discussion. We would like to show that even for aims of automation, ND, in suitable version, may be as good as other popular systems. Moreover, the obtained proofs may be more readable for humans.

We do not consider connection calculi, refutation systems and goal oriented systems. Although some of them were used in the formalization of some modal logics[13] their construction cannot be easily compared with our paradigm of a deductive system. In fact, this remark applies only to some kind of refutation systems, based on the original idea of Łukasiewicz, which may be called refutation systems in the strict sense. Sometimes the name refutation (or rejection) system is used in a more general sense covering also some variants of tableau systems.

[13]One may mention here the works of Skura [258] and Goranko [113] concerning refutation systems for modal logics, and the work of Wallen [278], where connection method is applied.

Chapter 2

Standard Natural Deduction

This Chapter is devoted to the description of standard systems of natural deduction (ND). After historical introduction in Section 2.1 we present some preliminary criteria which should be satisfied by any system of natural deduction. Sections 2.3 and 2.4 develop a systematization of existing systems based on two features: the kind of data used by a system and the format of proof setting. As a result we divide traditional ND systems into F- and S-systems (rules defined on formulae or sequents), and on L- and T-systems (linear or tree proofs). The division is not exhaustive as there may be systems operating on other types of items (e.g. labelled formulae – cf. Chapter 8) The main conclusion of this part is that almost all of the known variants of ND may be traced back to the independent works of Jaśkowski and Gentzen who started the investigation on non-axiomatic deductive systems.

Section 2.5 introduces ND system for **CPL** which will be used as a basis for further modifications in the sequel. It is F- and L-system KM developed by Kalish and Montague on the basis of Jaśkowski original ND. We have chosen KM as the basic system because it is particularly useful for our considerations and in many respects seems to be superior to other known systems. Section 2.6 contains adequacy proof for KM; in particular it lays down the general schema for proving soundness of all extensions of KM in later chapters. Finally, we present an extension of KM to first-order classical and free logic (with and without identity) with a discussion of the most important alternatives. In particular, we introduce variants based on Jaśkowski style rules and on Gentzen style rules, both in two formulations using variables or parameters.

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*, Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_2,

2.1 Origins of ND

We would like to start with a brief history of natural deduction. In fact, the historical remarks will also accompany more systematic treatment of ND in further sections. It is not our intention to compete with the excellent and very detailed account of these matters by Pelletier [205]. So we rather do not repeat the information contained in his paper, except from the most basic. Instead we provide some additional information, particularly in Sections 2.3 and 2.4. The pages devoted to the system of Gentzen from [110] and some of his followers, as well as the remark on Suszko work [267], may help to extend and complete the picture of the discipline.

1934 is commonly accepted as the first year in the history of non-axiomatic deductive systems in general, and natural deduction systems in particular. Two papers: of Jaśkowski [157] and Gentzen [109], published in this year, are the fundamental publications on the subject. The name was introduced by Gentzen; Jaśkowski used the term "composite system" in contrast to Hilbert axiomatic "simple system". ND systems were constructed independently by these two logicians as formal realizations of traditional means of proving theorems in mathematics, science and ordinary discourse. Since then, several variants of ND were devised and presented in hundreds of logic textbooks, giving an evidence that ND systems are commonly accepted as the most efficient way of teaching logic. Still, simplifying a bit, but truly indeed, we may say that everything sofar constructed in the field of ND and the related systems is based, more or less directly, on the ideas developed by these two researchers.

According to many authors, the origins of ND, recognized as a manner of deduction based on arbitrary assumptions and application of simple, self-evident rules, should be traced back to Ancient Greece. One can mention e.g. the discussion between Corcoran and Łukasiewicz, concerning the interpretation of Aristotle's syllogistic. Corcoran [73] claimed that Łukasiewicz in [180] mistakenly interpreted syllogistic as an axiomatic system, and proposed an interpretation in terms of inference rules and proofs from assumptions. One can also look for the genesis of ND system in Stoic logic, where many researchers (like Mates [187]) identify a practical application of *Deduction Theorem.* But all these examples, even if we agree with the arguments of logic historians, are only examples of using some proof techniques. There are no traces of theoretical interest in their justification, so we should agree that Jaśkowski and Gentzen are the inventors of ND.

Jaśkowski was somewhat influenced by Łukasiewicz, who posed on his

Warsaw seminar in 1926 a problem: how to describe, in a formally proper way, proof methods applied in practice by mathematicians (cf. Woleński [283]). Hilbert's proof theory already offered high standards of precise formalization in terms of axiom systems. But the process of actual deduction in Hilbert calculi is usually complicated and needs a lot of invention, also, the ready proofs are lengthy, difficult to decipher, and far from informal arguments produced by means of commonly used techniques like *conditional proof, indirect proof, proof by cases.* In consequence, axiom systems, although theoretically satisfying, were considered by many researchers as practically inadequate and artificial. Hence, two goals were involved in Łukasiewicz's idea: first, theoretical justification of traditional proof methods on the ground of modern logic; second, formally correct and practically useful system of deduction.

In response to Łukasiewicz's problem, Jaśkowski presented his first results on ND in 1927, at the First Polish Mathematical Congress in Lvov, mentioned in [156]. Final solution was offered in [157] because Jaśkowski had a lengthy break in his research due to illness and family problems. Gentzen published the first part of his famous paper also in 1934, but the first results are present in [108]. This early paper, however, is concerned not with ND but with the first form of sequent calculus. Gentzen was influenced by Hertz [129], where a tree-format notation for proofs, as well as the notion of a sequent were introduced.

It should be of no surprise that the two logicians with no knowledge of each other's work, independently proposed quite different solutions to the same problem. The need for deduction systems of this sort was in the air. In fact, one can mention other efforts in this respect. It is worth saying that ND-like rules were practically applied in the twenties by many logicians from Lvov-Warsaw School like Leśniewski, Salamucha, Tarski, as is evident from their papers.

In particular, the introduction of deduction theorem into the realm of modern logic seems to be one of the most important steps in this direction. Although Herbrand has proved it formally for axiomatic systems in 1930 [127], it was stated by him already in [126]. At the same time Tarski included DT as one of the axioms of his *Consequence Theory* in [272]; in practice he has used it since 1921. All these results may be seen as the first bridge between the theory and practice, finally realized by Jaśkowski and Gentzen. But what kind of deductive systems were exactly proposed by these two logicians? They were not identical; in fact, each of them has offered essentially two different systems that gave an impact to creation of a great variety of systems nowadays called natural deduction. Before describing some types of ND, and in particular showing their origins in papers of Jaśkowski and Gentzen, we should try to delimit a scope of the family of ND-systems.

2.2 Preliminary Characterization

It seems that there is no precise definition of ND-systems that would be generally accepted. The term is often used in a very broad sense, so that it covers almost everything which is not an axiom system. Some authors use to say that Gentzen's *sequent calculus* is ND-system, or that several forms of tableaux are natural deduction. All these systems are actually in close relationship, but here this notion is taken in a narrow sense. There are at least three reasons to make such a choice:

- Historical. Original ideas of Gentzen, who in [109] introduces two systems: NK (Natürliche Kalkül) and LK (Logistiche Kalkül). The former is just ND system, whereas the latter, which is a sequent calculus, is meant as a technical tool to prove some metatheorems on NK, not as a kind of ND.

- Ethymological. ND is supposed to reconstruct, in a formally proper way, traditional ways of reasoning (cf. remarks on Łukasiewicz's seminar problem in the preceding section). It is disputable if existing ND systems realize this task in a satisfying way, but certainly systems like TS or SC are worse in this respect.

- Practical. Taking a term ND in a wide sense would be classifying operation of doubtful usefulness. From the point of view of our task it is more convenient to use more fine-grained concept.

But what do we mean by ND in a narrow sense? Pelletier [205] shows that some of the proposed definitions are too strict since they exclude some systems usually treated as ND. It is better not to be very demanding in the preliminary characterization. Informally, we treat as ND-system any deductive system satisfying at least 3 criteria:

- There are some means for entering assumptions into a proof and also for eliminating them. Usually it requires some bookkeeping devices for indicating the scope of an assumption, and showing that a part of a proof depending on eliminated assumption is discharged.

- There are no (or, at least, very limited set of) axioms, because their role is taken over by the set of primitive rules for introduction and elimination of logical constants which means that elementary inferences instead of formulae are taken as primitive.

- Genuine ND system admits a lot of freedom in proof construction and possibility of applying several proof search strategies, like conditional proof, proof by cases, proof by reductio ad absurdum e.t.c.

This is a very broad characteristics and, as we will see, it allows a lot of freedom in concrete realizations. Some authors (c.f. [20] or [205]) formulated additional conditions, but in our opinion these 3 are essential. The main point is that real ND-system should be open for different proof constructions. The user is free in constructing direct, indirect or conditional proofs. He may build more complex formulae or decompose them, as respective introduction/elimination rules allow. Instead of using axioms or already proved theses, he is rather encouraged to introduce assumptions and derive consequences from them (although the presence of axioms is permitted). This flexibility of proof construction in ND is in striking contrast to other types of deductive systems usually based on one form of proof.[1]

Many existing systems satisfy this loose characteristics but differ in many other respects – Pelletier describes nine such points of choice. Many of these differences are superficial, but not all, so it is reasonable to distinguish some types of ND, provisionally mentioned in Chapter 1. In our opinion the most important differences – at least on the propositional level – are two:

- The kind of basic items (data structures) on which inference rules are defined.

- The overall format of proof representation.

The first distinction occurs on the level of a calculus, the second, on the level of its realization. In short: proofs in ND-systems are settled down generally as trees (tree- or Gentzen-format or T-system, see [109]) or sequences (linear- or Jaśkowski format or L-system, see [157]), basic items of these proofs (nodes of proof-tree) may be formulae (F-systems), sequents (S-systems), sets of formulae (generalized clauses) or other structured data (e.g. formulae with labels). We take a historically oriented look at some of these distinction to show their advantages and disadvantages.

[1] This is why we do not treat sequent calculi and tableau systems as examples of natural deduction systems. In ordinary SC we have only cumulative proofs and introduction rules, in tableau systems only indirect proofs and elimination rules.

2.3 Data Structures

2.3.1 F-Systems

The first ND systems were F-systems, i.e. their rules were defined on formulae as basic items. In Chapter 1 we have mentioned that in ND-systems of this sort one has two types of rules: *rules of inference* and *proof construction rules*. The former have the form Γ/φ; we read them as follows: if we have all formulae from Γ in the derivation we can add φ to this derivation. By derivation we mean an attempted proof, i.e. unfinished tree or sequence.

In ND-system we also need some *proof construction rules* that allow us to build a proof, enter additional assumptions opening nested subderivations, and show under what conditions we may discharge these assumptions and close the respective subderivations. For systems considered they have a general form:

$$\text{if } \Gamma_1 \vdash \varphi_1, ..., \Gamma_n \vdash \varphi_n, \text{ then } \Delta \vdash \psi.$$

In this schema the antecedents refer to subderivations which, if completed (φ_i is inferred from $\Gamma_i, i \leq n$), give a justification for ψ on the basis of Δ. Typical proof construction rules are meant to formalize the old and well known proof techniques like conditional proof, indirect proof, proof by cases e.t.c.

This preliminary characterization applies equally well to Jaśkowski's and to Gentzen's system from [109]. On the level of calculus both systems are similar; in Jaśkowski system, formulated in implicational-negational language, we have the following propositional rules[2]:

$(\rightarrow E)$	$\varphi, \varphi \rightarrow \psi \;/\; \psi$
$[\rightarrow I]$	if $\Gamma, \varphi \vdash \psi$, then $\Gamma \vdash \varphi \rightarrow \psi$
$[\neg E]$	if $\Gamma, \neg\varphi \vdash \bot$, then $\Gamma \vdash \varphi$

Here $(\rightarrow E)$ is the only inference rule letting for deduction of new formulae from the old ones,[3] whereas $[\rightarrow I]$ and $[\neg E]$ are proof construction rules formalizing traditional forms of conditional proof and indirect proof. Jaśkowski

---

[2]Note that Jaśkowski did not use $\bot$, so it is present in the formulation of rules as a metalinguistic sign of inconsistency. Also the names of rules are not original; he just used on the level of realization names like: rule I, rule II, ... Note that in general we do not use the original names of rules from described systems since several authors either use different notation for the same things or the same names for different things.

[3]In rules of elimination like $(\rightarrow E)$ we will ocassionaly use traditional distinction between major premise which contains eliminated constant, and minor premise.

proposed also modifications of his calculus leading to weaker propositional logics, namely, he has observed that the last rule may be weakened:

$[\neg I]$ if $\Gamma, \varphi \vdash \bot$, then $\Gamma \vdash \neg\varphi$

This form yields ND formalization of Kolmogorov's version of intuitionistic logic, whereas deletion of any rule for $\neg$ captures positive logic.[4] By the end of his paper he also considered proper rules for conjunction and proposed the following:

$(\wedge I)$ $\varphi, \psi \ / \ \varphi \wedge \psi$
$(\wedge E)$ $\varphi \wedge \psi \ / \ \varphi$ and $\varphi \wedge \psi \ / \ \psi$

Gentzen's ND system NK (Natürliche Kalkül) from [109] has $[\neg I]$ and the same rules for $\rightarrow$ and $\wedge$. Additionally he considered further rules for $\neg$ and rules for $\vee$:

$(\neg E)$ $\varphi, \neg\varphi \ / \ \bot$ and $\bot \ / \ \varphi$
$(\vee I)$ $\varphi \ / \ \varphi \vee \psi$ and $\psi \ / \ \varphi \vee \psi$
$[\vee E]$ if $\Gamma, \varphi \vdash \chi$ and $\Delta, \psi \vdash \chi$, then $\Gamma, \Delta, \varphi \vee \psi \vdash \chi$

Although we have in fact three rules for $\neg$, the calculus is adequate for intuitionistic logic. Gentzen himself obtains formalization of **CPL** by addition of the law of excluded middle $\neg\varphi \vee \varphi$ as a sole axiom, although he noted that one may use for that aim a rule of inference:

$(\neg\neg E)$ $\neg\neg\varphi \ / \ \varphi$

But this rule was not in harmony with his basic requirement to have a pair of rules (introduction-elimination) for each constant.

One should note one more thing concerning generally the calculus of any F-system. In order to have a complete formalization of **CPL** or any other logic, we must assume that it contains two more rules:

(A) $\varphi \ / \ \varphi$
$[TR]$ if $\Gamma \vdash \varphi$ and $\Delta, \varphi \vdash \psi$, then $\Gamma, \Delta \vdash \psi$

These rules are implicit in any form of realization of F-system. (A) justifies introduction of assumptions to a proof, $[TR]$ is a form of cut justifying the very process of deduction of the last formula from assumptions.

[4] Jaśkowski formulated also ND system for propositional logic with quantifiers.

2.3.2 S-Systems

The differences in the set of rules proposed by Jaśkowski and Gentzen are not sufficient to maintain that they represent two different approaches to ND. The deep difference beetwen their systems lies on the level of realization and will be discussed in the next section. But one, more serious, difference appears also on the level of calculus if we take under consideration the next ND system of Gentzen [110].

Let us recall that rules of ND systems may be defined not only on formulae but also on other items, e.g. on sequents (S-systems). The first ND S-system should not be identified with sequent calculi of the sort described in the next Chapter. The latter (Gentzen's LK – Logistiche Kalkül – from [109]), uses only introduction rules for constants (but defined both for antecedent and succedent – cf. Chapter 3), whereas the former, also due to Gentzen, is essentially ND system since it has both introduction and elimination rules (but usually only in the succedent; antecedent simply displays active assumptions). So it is a kind of a compromise between his own NK and LK systems. Although the first such system was officially introduced by [110] in fact it was already present implicitly in the proof of adequacy of NK in [109]. Gentzen proved adequacy by showing equivalence of NK with axiom system by means of his sequent system LK. One part of the proof shows how to transform each NK proof into LK proof. In order to do that Gentzen rewrites each rule in such a way that it has an added record of all active assumptions. In [110] the system is defined in such a way just from the beginning. In its propositional part (the whole system has also rules for quantifiers and Peano arithmetic) it has one sequent (A_S) $\varphi \Rightarrow \varphi$ regulating introduction of assumptions, and the following rules for **CPL**[5]:

$$
\begin{array}{rl}
(W_S) & \Gamma \Rightarrow \varphi \;/\; \psi, \Gamma \Rightarrow \varphi \\
(\wedge I_S) & \Gamma \Rightarrow \varphi; \Delta \Rightarrow \psi \;/\; \Gamma, \Delta \Rightarrow \varphi \wedge \psi \\
(\wedge E_S) & \Gamma \Rightarrow \varphi \wedge \psi \;/\; \Gamma \Rightarrow \varphi \;\text{ and }\; \Gamma \Rightarrow \varphi \wedge \psi \;/\; \Gamma \Rightarrow \psi \\
(\vee I_S) & \Gamma \Rightarrow \varphi \;/\; \Gamma \Rightarrow \varphi \vee \psi \;\text{ and }\; \Gamma \Rightarrow \psi \;/\; \Gamma \Rightarrow \varphi \vee \psi \\
(\vee E_S) & \Gamma \Rightarrow \varphi \vee \psi;\; \varphi, \Delta \Rightarrow \chi;\; \psi, \Lambda \Rightarrow \chi \;/\; \Gamma, \Delta, \Lambda \Rightarrow \chi \\
(\rightarrow I_S) & \varphi, \Gamma \Rightarrow \psi \;/\; \Gamma \Rightarrow \varphi \rightarrow \psi \\
(\rightarrow E_S) & \Gamma \Rightarrow \varphi;\; \Delta \Rightarrow \varphi \rightarrow \psi \;/\; \Gamma, \Delta \Rightarrow \psi \\
(\neg I_S) & \varphi, \Gamma \Rightarrow \psi;\; \varphi, \Delta \Rightarrow \neg\psi \;/\; \Gamma, \Delta \Rightarrow \neg\varphi \\
(\neg E_S) & \Gamma \Rightarrow \neg\neg\varphi \;/\; \Gamma \Rightarrow \varphi
\end{array}
$$

[5] In fact, Gentzen defined antecedents of sequents rather as lists of formulae, so he needs also a rule of permutation and contraction for elements of antecedents.

There is a clear correspondence between inference rules of F-system and rules of S-system. Every rule of the form:

$(Rule)\ \varphi_1, ..., \varphi_k \ / \ \psi$

may be changed into:

$(Rule_S)\ \Gamma_1 \Rightarrow \varphi_1, ..., \Gamma_k \Rightarrow \varphi_k \ / \ \Gamma_1, ..., \Gamma_k \Rightarrow \psi$

The correctness of this operation is justified by k applications of cut on a sequent representing $(Rule)$ and sequents being premises of $(Rule_S)$. One may note that all rules of Gentzen's S-system may be derived from his F-system.

The relationship between proof construction rules and the respective rules of S-system is even more straightforward. One may observe that in S-system we do not need to distinguish a category of proof construction rules; they are just represented by rules operating on sequents, e.g. $(\rightarrow I_S)$ is a counterpart of $[\rightarrow I]$ e.t.c. Only $(\vee E_S)$ slightly departs from this scheme, probably because Gentzen wanted to avoid a rule with explicit introduction of a constant in the antecedent. In fact, there is some subtle difference between proof construction rules in F-systems and their counterparts in S-systems concerning interpretation of sets Γ, Δ. In S-system an antecedent of $\Gamma \Rightarrow \varphi$ is always a set of active assumptions of φ, whereas in F-system it is not necessarily so. Γ in $\Gamma \vdash \varphi$ is meant as a set of (not necessarily all) formulae from which φ is deduced; so they may be not only active assumptions of φ but also some formulae which were deduced from them. In case we have some formula displayed before $\vdash$ as an additional assumption of a subproof, Γ is just a set of all formulae which are accessible for further inferences at the moment when this assumption was introduced. This is important to note because it would be difficult to state suitable rules for modal logics (cf. Chapter 6) if we restrict Γ only to the record of assumptions. Clearly, one may introduce more accurate notation having separate symbols for assumptions and for other formulae[6] but for our purposes it is not necessary. Only in case of some rules for quantifiers the restriction to assumptions is important but it may be stated in side conditions.

Anyway, the difference in reading Γ-s in both systems is not an obstacle for seeing a deep similarity of the rules in both approaches. The relationship between these two types of ND shows that on the level of a calculus we may

[6]This is, for example, a solution of Dyckhoff in [81] with H denoting hypotheses (assumptions) and F denoting facts, i.e. assumptions and formulae inferred from them.

obtain a uniform characterization of both F- and S-systems, if we admit that every ND system consists of sequents and sequent rules.[7] In this perspective the difference is in the interpretation of constituents and in their proportion. In F-system sequents correspond to inferences and sequent rules to proof construction; in S-system sequent rules are just inferences.

Note that one may obtain S-system based on the calculus given for F-system in a slightly different way. We may directly use all inference rules of F-system as starting sequents (assumptions). If we take a full set of rules of Gentzen's F-system, the new S-system would consists of many starting sequents instead of one (A_S), and of only a few rules. The calculus of Jaśkowski's system, if treated as a kind of S-system, would contain two sequents: (A_S) and ($\rightarrow E_S$) $\varphi, \varphi \rightarrow \psi \Rightarrow \psi$, and three sequent rules:

$$
\begin{array}{ll}
(\rightarrow I_S) & \Gamma, \varphi \Rightarrow \psi \ / \ \Gamma \Rightarrow \varphi \rightarrow \psi \\
(\neg E'_S) & \Gamma, \neg\varphi \Rightarrow \bot \ / \ \Gamma \Rightarrow \varphi \\
(\mathit{Cut}\text{-}\mathit{ND}) & \Gamma \Rightarrow \varphi \ ; \ \varphi, \Delta \Rightarrow \psi \ / \ \Gamma, \Delta \Rightarrow \psi
\end{array}
$$

Note that except original rules of suitable F-system we must explicitly add (A_S) and a form of cut. The last rule is necessary in such S-system to cover transitivity of deductions which in F-system is covered by implicit rule [TR]. S-system of this sort (many sequents, few rules) was proposed in 1940s by Suszko [267], although with different set of sequents and with only structural rules of cut, weakening, contraction and permutation. Such S-systems are not very popular however; one may find rather several ND systems more similar to original Gentzen's system, with many rules. One of the first is implicitly present in influential Kleene's textbook [162], although he introduced a complete set of rules for classical logics not as an independent system but as an enrichment of an axiom system with derivable rules. Various systems of this sort may be found e.g. in [234] and [82]. Particularly interesting is a variant of S-system due to Hermes [128] because it contains rules for introduction and elimination of constants also in antecedents of sequents which makes it a kind of syntactic hybrid of sequent ND and ordinary sequent calculus. The richness of its deductive apparatus is clearly connected with practical orientation of this system. Original system of this sort representing purely theoretical interests is provided by von Plato and Negri [193], where only the rules of elimination from both antecedents and succedents are provided. This hybrid of SC and ND is obtained via the analysis of normalization proof.

[7] Similar way of description is in Wójcicki [285], where every deductive system consists of axioms, H-rules (i.e. sequents) and G-rules (i.e. sequent rules).

Sequential character of some ND systems is sometimes obscured; in the next section we will describe a system of Suppes which is usually treated as a kind of F-system.

2.4 Trees or Sequences?

2.4.1 Problems with Trees

The description of rules on the level of calculus is rather insensitive to the format of proof. F-systems of Jaśkowski and Gentzen are not much different in this respect, however, they differ in realization: Jaśkowski proposed systems with linear proofs (L-system), whereas Gentzen's NK uses tree proofs (T-system).

The distinction between tree- or linear-format ND-systems bears little theoretical import as binary tree can be redefined as a sequence.[8] Nonetheless in practice it does matter since in L-systems we deal with formulae, whereas in T-systems we deal with their concrete occurrences. As we may use the same formula many times in L-proof, we must have some devices for cancelling the part of a proof which lies in the scope of an assumption already discharged. Otherwise, we could "prove" everything – the well known phenomenon to all teachers of introductory courses in logic.

All this is not possible in tree-proofs. As we are operating not on formulae but on their single occurrences, every leaf of a proof tree is an assumption and the root is a formula to be proved. Transitions between nodes correspond to elementary inferences, and premises of any application of a rule must always be displayed directly above the conclusion. Consequently, we cannot use in a proof anything that depends on discharged assumptions, because the part of a proof responsible for deduction of a formula must be reproduced above. Hence Gentzen did not have to bother about technical devices to block nonvalid deductions. Tree format requires less complicated machinery and as such is very handy in representing ready proofs – the structure of inferential dependencies is readably represented. No wonder that in works concerned with theoretical investigations this format is very popular (good witness is Prawitz [220]).

As it is often the case, a feature attractive on the one hand, is the source of problems, on the other. Although trees clearly show the structure of completed proofs, they are hardly suitable for actual derivation. Mental process of proof construction has rather a linear structure; we start with

[8] Cf. e.g. an algorithm in Chapter 4.

assumptions and deduce conclusions until we get the desired goal. Gentzen himself was well aware of this fact when he wrote that "we are deviating somewhat from the analogy with actual reasoning. This is so, since in actual reasoning we necessarily have (1) a linear sequence of propositions due to the linear ordering of our utterances, and (2) we are accustomed to applying repeatedly a result once it has been obtained, whereas the tree form permits only of a single use of a derived formula." ([109] citation from [268, page 76])

According to Gentzen, however, this form of representation is simpler and resulting deviations are inessential. On the other hand, despite the above mentioned inconvenience, Jaśkowski decided to use linear format as much closer to actual reasoning, and much more useful for actual proof-search.

The choice of proof format has also some computational advantages; we can show that for each proof $\mathcal{D}$ in tree format we can provide linear proof $\mathcal{D}'$ such that the length of $\mathcal{D}'$ is the same or smaller than the length of $\mathcal{D}$. The converse does not hold because in tree proofs we work with occurrences of formulae when in linear proofs we work with formulae themselves. It forces us to repeat many times the same proof-trees if their starting assumptions are used several times. Consider, e.g. any thesis of the form $\varphi \to (\psi \to \chi_1 \wedge \ldots \wedge \chi_n)$, where $\varphi = (\varphi_1 \wedge (\psi \wedge \varphi_1 \to \chi_1)) \wedge \ldots \wedge (\varphi_n \wedge (\psi \wedge \varphi_n \to \chi_n))$.

General schema of a tree proof looks as follows:

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\mathcal{D}_1 & & \mathcal{D}_n \\ \chi_1 & \cdots\ \cdots & \chi_n\end{matrix}}{\chi_1 \wedge \cdots \wedge \chi_n}}{\psi \to \chi_1 \wedge \cdots \wedge \chi_n}}{\varphi \to (\psi \to \chi_1 \wedge \cdots \wedge \chi_n)}$$

For simplification we assumed the generalized form of $(\wedge I)$ by which we can deduce n-ary conjunction from n premises in one step. Every $\mathcal{D}_i (1 \leq i \leq n)$ represents a proof of χ_i and has the following form:

$$\dfrac{\dfrac{[\psi] \quad \dfrac{\dfrac{[\varphi]}{\varphi_i \wedge (\psi \wedge \varphi_i \to \chi_i)}}{\varphi_i}}{\psi \wedge \varphi_i} \qquad \dfrac{\dfrac{[\varphi]}{\varphi_i \wedge (\psi \wedge \varphi_i \to \chi_i)}}{\psi \wedge \varphi_i \to \chi_i}}{\chi_i}$$

$[\psi]$(and $[\varphi]$) means that an occurrence of an assumption ψ (φ respectively) is being discharged in effect of application $[\to I]$ in the last line of our proof.

It is clear that in the tree proof we have to assume ψ n-times and φ even $2n$-times which is necessary to derive each χ_i. On the other hand, in Jaśkowski format linear proof we are forced to assume both φ and ψ only once and use them repeatedly which simplifies greatly the representation of proof (see an example in Section 2.5.3).

To the best of my knowledge no results were obtained showing how much computationally worse are tree ND-proofs in comparison to linear ones. One can compare this with the situation in other types of deductive systems. For axiom systems, Krajicek [168] has shown that linear proofs may be p-simulated by tree proofs, but for resolution it is known (e.g. [26]) that linear representation is even exponentially more efficient than refutations in tree form. It is an open question whether ND is closer to axiom systems or to resolution in this respect, but it is obvious that the ability to reuse formulae in a proof leads to shorter proofs.

2.4.2 Problems with Linear Proofs

On the other hand, and exactly for the same reason, linear proofs have scoping difficulties and require some kind of bookkeeping devices for separating the parts of proof which are in the scope of discharged assumption (not available). This is the case of rules, as $[\rightarrow I]$ or $[\neg E]$, that are not inference rules but proof construction rules. As such, they show that if some proof is being constructed on the basis of some assumption, then, in the effect, we obtain another proof in which this assumption is not in force. Thus a linear proof admitting additional assumptions and means for closing dependant parts of a proof is, in fact, not a simple sequence of formulae but rather a richer structure containing nested subderivations (subordinate proofs). Precise definitions will be provided later in Section 2.5.3.

There are many techniques devised for several ND-variants from logic textbooks for dealing with this problem. Basically, all of them are some variations of two solutions introduced by Jaśkowski. The first original Jaśkowski solution of the problem consisted in making boxes for each assumption and the dependent part of a proof. Every introduction of an assumption was connected with starting a new box, and this assumption was always put as the first formula in it. An application of any proof construction rule was connected with closing a current box, and the inferred formula was immediately put as an element of outer derivation. He also used an additional rule of repetition to shift a formula from outer open box to the inner one; the transition in the other direction was of course forbidden. Schematically, the application of both proof construction rules in his system looks as follows:

$$
\begin{array}{l}
\Gamma \\
\quad \boxed{\begin{array}{l} \varphi \\ \Gamma' \\ \vdots \\ \psi \end{array}} \\
\varphi \rightarrow \psi
\end{array}
\qquad\qquad
\begin{array}{l}
\Gamma \\
\quad \boxed{\begin{array}{l} \neg\varphi \\ \Gamma' \\ \vdots \\ \bot \end{array}} \\
\varphi
\end{array}
$$

In the diagrams possibly empty $\Gamma' \subseteq \Gamma$ refers to formulae obtained by repetition.

In publication Jaśkowski applied (perhaps due to editorial problems) an apparatus of numeric prefixes, instead of boxes. These are finite sequences of natural numbers separated with dots and written before each formula in a proof, except a thesis. Each time we enter an assumption we extend prefix with additional number[9]; each application of $[\rightarrow I]$ or $[\neg E]$ is connected with subtraction of the last number in the prefix. Application of inference rules, like $(\rightarrow E)$ is admitted only if prefixes of both premises are initial parts or are identical to the prefix of the last formula in a proof (prefix of a conclusion must be identical to it). In this way Jaśkowski avoided the introduction of repetition as a rule. Jaśkowski thought of prefixes as indicators of domains in which formulae with this prefix are valid. Thus formulae with empty prefixes are ordinary theses, and those with nonempty prefixes are theses relative to some domain in which some suppositions are postulated as valid. Prefixes are then records of dependency of a formula on assumptions in the context of a proof.

Incidentally, the innovation introduced by Jaśkowski (i.e. prefixes) may be classified in different way, especially if we take into account his philosophical motivations (borrowed from Leśniewski) concerning the dynamic nature of a deductive system. In brief: a prefix is seen as a domain where assumptions connected with this prefix and with each of its subprefixes are believed to be valid; every prefixed formula is a thesis of this domain. In this perspective, we may treat his second version as the first example of ND defined not on formulae but on labelled formulae!

Below we illustrate both versions with an example of a proof.

[9] Additionally Jaśkowski has used a prefix S (from supposition) in front of any assumption.

1	p	$ass.$
2	$\neg\neg\neg p$	$ass.$
3	p	$(1, rep.)$
4	$\neg\neg p$	$ass.$
5	$\neg\neg\neg p$	$(2, rep.)$
6	$\neg p$	$[4, 5, \neg E]$
7	$\neg\neg p$	$[3, 6, \neg E]$
8	$p \rightarrow \neg\neg p$	$[\rightarrow I]$

1	$1.Sp$	$ass.$
2	$1.1.S\neg\neg\neg p$	$ass.$
3	$1.1.1.S\neg\neg p$	$ass.$
4	$1.1.\neg p$	$[2, 3, \neg E]$
5	$1.\neg\neg p$	$[1, 4, \neg E]$
6	$p \rightarrow \neg\neg p$	$[\rightarrow I]$

Despite the apparent differences, both Jaśkowski systems have one thing in common – the essential idea of dividing a proof into separated and partially ordered subproofs. It appears as the most popular solution in hundreds of textbooks where ND-techniques are applied. His first version, although abandoned by the author himself, is much more popular nowadays. It has many variants but there is always some graphic device added to linear sequence of formulae in a proof. Original format of boxes was used by Kalish and Montague [159]), but with some adjustments, introduced earlier in [158], which make their system one of the most flexible in practice. Simplified account, where each assumption is entered with the vertical line which continues until this subproof is in force, is due to Fitch [89], whereas popular system of Copi [72] applies bracketing to closed subproofs. The second solution of Jaśkowski is not so popular in ND setting. Borkowski and Słupecki [54] in their ND system followed this route but also with some simplifications; prefixes are mixed together with the numbers of lines of the proof.

Such ND systems are commonly called Fitch-style ND, however here we will reserve for them historically more appropriate name, i.e. ND in Jaśkowski's format.[10] Since in this approach parts of proof are separated, it proved especially useful with respect to many nonclassical logics formalized via ND systems. In what follows, we will present in detail some solutions for modal logics which are hardly realizable without assuming Jaśkowski format.

[10] It must be said that Jaśkowski was not as lucky as Gentzen, whose contribution into development of proof methods is widely known. There are a lot of books and papers using some variant of ND in Jaśkowski format but crediting it to Fitch or to Gentzen.

2.4.3 Suppes' Format

Finally we turn to the problem of realization of S-systems of ND. Gentzen himself in [110] adopted tree proofs, exactly as in NK. But S-system may be realized in the linear format, as well, as was noticed by some of his followers (e.g. Hermes [128]) Moreover, in the linear format it is even more advantageous than in Jaśkowski format requiring bookkeeping devices to indicate dependence of a formula on assumptions. It is so because we are operating on sequents where each formula (succedent) has a record of its assumptions (antecedent) and there is no risk of performing deductions on the basis of discharged assumptions. This system turned out to be very influential in the field of applied ND systems thanks to Suppes' modification [266]. In his ND systems we apparently operate on formulae not on sequents, because instead of antecedents we simply write down numbers of lines where suitable assumptions appeared for the first time. This device was originally introduced by Feys and Ladriere in their French translation [86] of Gentzen [109]. In Suppes' version, popularized by Lemmon [174], it appeared in many textbooks on logic. So to avoid confusion with Gentzen's original F-system but with tree proofs, we will call such an approach, ND in Suppes' format. One should add that there are also ND L-systems operating on explicit sequents, and even using rules defined on antecedents, e.g. Hermes [128].

Below we give some example of proof in Suppes' format:

$\{1\}$	1	$p \rightarrow q$	$ass.$
$\{2\}$	2	$r \rightarrow q$	$ass.$
$\{3\}$	3	$p \vee r$	$ass.$
$\{4\}$	4	p	$ass.$
$\{1,4\}$	5	q	$(1, 4, \rightarrow E_S)$
$\{6\}$	6	r	$ass.$
$\{2,6\}$	7	q	$(2, 6, \rightarrow E_S)$
$\{3,1,2\}$	8	q	$(3, 5, 7, \vee E_S)$
$\{1,2\}$	9	$p \vee r \rightarrow q$	$(8, \rightarrow I_S)$
$\{1\}$	10	$(r \rightarrow q) \rightarrow (p \vee r \rightarrow q)$	$(9, \rightarrow I_S)$
	11	$(p \rightarrow q) \rightarrow ((r \rightarrow q) \rightarrow (p \vee r \rightarrow q))$	$(10, \rightarrow I_S)$

Let us noticed by the way that Pelletier [205] in his historical and systematical account of ND has omitted the second system of Gentzen, consequently crediting such a kind of system to Suppes. He is certainly right that

ND in Suppes' format is really different from Jaśkowski's prefixes because of the reasons we discuss in the sequel. Certainly Suppes was also unaware of Gentzen's S-system when creating his own (e.g. his rules for quantifiers significantly differ from Gentzen's rules), but from the taxonomical point of view it is in fact the same approach!

Apparently, the solution of Suppes resembles more the second system of Jaśkowski, but there is a big difference between them. In Jaśkowski's approach any two formulae from neighbour-lines may have only the same prefix or one digit longer or smaller, because subproofs are not only separated but also ordered – that is the very idea of *subordinated proofs*. In Suppes' format in neighbour-lines we may have formulae depending on different sets of assumptions, because the order of assumptions plays no role in this approach. So, a proof is not divided into separated subproofs but is really a linear sequence of sequents. That is why in [144] ND in Jaśkowski's format was called an *ordered assumptions approach*, and linear S-systems were called a *recorded assumptions approach*.

This difference may have strong impact on the length of proofs. In Jaśkowski format all formulae deduced inside a current subderivation are treated as dependent on its assumption (and all assumptions of outer open derivations), even if this assumption was not actually used in their deduction. In consequence, after closing this subproof, all formulae in it are unavailable, even if they are not really dependent on discharged assumption. This may lead to repetition of deduction of formulae that were deduced before. In Suppes' format there is no danger of such repetitions since there is no isolation of subproofs, and for each formula we have a record of its real assumptions.

So not only bookkeeping devices can be dispensable with in the latter format, but also, in some sense, this approach allows for more flexibility in carrying actual derivations than Jaśkowski's approach, at least in case of classical logic. For modal logics the situation is different as the prevailing technique of formalization due to Fitch is essentially based on the explicit isolation and hierarchization of subproofs (cf. Chapter 6). One may also diminish the risk of repetition of the same inferences in Jaśkowski's format by introducing secondary inference rules. The redundancy of inference rules usually helps to decrease the number of necessary additional assumptions (and subderivations).

Following the tracks of both Jaśkowski and Gentzen on the development of proof methods, we cannot forget about the possibility of mixing different techniques. A good example is Anderson and Belnap's ND system for relevant logics (cf. [5]). It is basically a system in Jaśkowski's format (Fitch's lines for subproofs) but formulae are indexed in the way which make them essentially Suppes' quasi-sequents. Such a "hybrid" technique makes possible controlling relevancy conditions. Also prefixed analytic tableaux of Fitting [93] for modal logics may be treated as mixing Smullyan's tree system with Jaśkowski's device of prefixing each formula. We will say more about this in Chapter 8 but one remark is in place here, namely, these examples show that a distinction between a sequent and a labelled formula is not sharp. Formally, in Suppes' and in Anderson/Belnap's ND we have formulae decorated with the sets of natural numbers. For this reason Gabbay [99] treats such a solution as one more example of application of labels (cf. Chapter 8)

To summarize the above historical-taxonomical considerations let us conclude that Jaśkowski and Gentzen laid down the foundations for further investigations of ND but in a slightly different fashion. Jaśkowski seemed to be more concerned with the practical aspects of deduction and his general approach as well as his technical solutions are of common classroom and textbook use. On the other hand, Gentzen was more theoretically oriented; his investigations led him to profound results in general proof theory. Some special approach to applied ND is rather a by-product of his later paper [110].

2.5 System KM

2.5.1 Rules

In this section we describe our official ND system for **CPL** that will be the point of reference in the subsequent investigations. We prefer to use Jaśkowski format (F- and L-system) because it is much more convenient for modal and other nonclassical logics. Presenting a calculus we follow quite closely the mode of presentation from Fitting [95] with his uniform compact notation. Our standard ND calculus for **CPL** consists of:

1. Inference rules

   (αE) α / α_i, where $i \in \{1,2\}$
   (αI) α_1 , α_2 / α
   (βE) β , $-\beta_i$ / β_j , where $i \neq j \in \{1,2\}$
   (βI) β_i / β , where $i \in \{1,2\}$
   ($\bot I$) φ, $-\varphi$ / $\bot$

$(\bot E)$ $\quad \bot \;/\; \varphi$
$(\neg\neg)$ $\quad \neg\neg\varphi \;//\; \varphi$

2. Proof Construction rules

$[COND]$ If $\Gamma, -\beta_1 \vdash \beta_2$, then $\Gamma \vdash \beta$
$[RED]$ If $\Gamma,\; -\varphi \vdash \bot$, then $\Gamma \vdash \varphi$

Note that the set of rules is highly redundant. It is common practice in ND (due to Gentzen) that for theoretical purposes each constant is characterized in a unique way with a pair of rules to introduce and eliminate this constant in a proof. The general schemata displayed above share both primitive rules and many others which in standard systems are treated as derivable. For example, conjunction is almost always[11] characterized by the rules of type (αI) and (αE) (cf. Section 2.3.1) but not by the rules of type β, where conjunction is negated; such rules are in most cases introduced later as derivable rules. On the other hand, disjunction and implication are characterized rather by the rules of type β, but also in a different manner for each constant; (βI) for implication is rather not applied being performed via $[COND]$ — and quite the contrary for disjunction. In the latter case a suitable rule of elimination is very often not our (βE), which is a traditional Modus Tollendo Ponens, but rather a proof construction rule $[\vee E]$ due to Gentzen (cf. Section 2.3.1) which is a formalization of traditional "proof by cases". One may easily observe that $[COND]$ covers classical $[\rightarrow E]$ limited to implications, whereas $[RED]$ covers both $[\neg E]$ and $[\neg I]$.

In general, the proportion of inference rules to proof construction rules may be different from the above. Theoretical reasons may lead to the introduction of other proof construction rules – as in von Plato/Negri [193]. On the other hand, practical reasons may lead to their reduction, like in the analytic version of ND presented in Chapter 4, where $[RED]$ is the only proof construction rule. Also $[RED]$ may be eliminable in a system with inference rules corresponding to double negation law and Modus Tollendo Tollens, and these are already covered in our calculus by $(\neg\neg)$ and general form of (βE). In fact, Quine's ND system [226] dispenses with indirect proof using only $[\rightarrow I]$.

In this book we are more concerned with utility than with theoretical purity, so this redundancy is certainly not a drawback but an advantage.

[11] This remark applies to practically oriented ND; in systems constructed for the needs of theoretical investigation other solution may work better, e.g. von Plato/Negri idea of generalized elimination rules in [193] leading to smooth proof of normalization.

We have already noted, when comparing Jaśkowski's and Suppes' formats that a redundant set of rules may potentially reduce some complexities of proof (defined below as the depth of a proof[12]). But there are also some theoretical advantages. The redundant set of rules facilitates a comparison and simulation of several deductive systems on the basis of ND. It is also handy to have a rich basis and make modifications by elimination and/or restriction of some rules; the possibility extensively investigated in Chapter 4. At this point one can also compare our calculus with remarks in [2] concerning the treatment of negation in ND and in tableaux. Negation in ND is not treated in symmetric way, as classical semantic features of this constant seem to dictate. Gentzen's rules give rather intuitionistic characterization. As a result, proofs of some DeMorgan's laws are not really "natural" in ND. The introduction of rules operating on negated formulae eliminates this inconvenience. Notice that even more redundant set of rules may be obtained, if we admit more general form of $[COND]$ covering also conditional contrapositive proofs:

If $\Gamma, -\beta_i \vdash \beta_j$, then $\Gamma \vdash \beta$, where $i \neq j \in \{1, 2\}$

In what follows this rule will be sometimes referred to as $[COND]$, too.

2.5.2 Realization

As we already know, these rules may be realized in different ways, depending on the variant of ND-system. We prefer a variant of Jaśkowski's format based on boxes since prefixes will be applied later to realize different goals (cf. Chapter 8). In what follows we will apply, as our formal basis, a system due to Kalish and Montague [159] (hence the name of the system). In KM nested subproofs are also put into boxes for making clear which part of a proof is active and which is not, but in a slightly refined way. The basic innovation of KM consists in using two types of lines in a proof:

- Usable-lines (U-lines for short) being ordinary lines of a proof containing assumptions, premises, and conclusions of applied rules; formulae occuring in U-lines are called U-formulae.

- Show-lines (S-lines for short) displaying formulae that one attempts to prove; from now on called S-formulae.

[12] In Chapter 4 we will introduce some drastic modification of ND leading to "flat" (no additional subderivations) proofs.

Actually, the application of Show-lines is a distinctive feature of KM. Their role may be compared to the function of queries in programming languages like Prolog or SQL. Show-line displays a formula which is the current (sub) goal of our derivation (i.e. at a given stage). Every S-line introduces a new subderivation but it is not a part of it. Technically, Show-line is a formula preceded with the prefix SHOW and it is introduced into every derivation at least once. We put "SHOW:φ" always at the beginning of a derivation of φ as the global goal of this proof, and we enter "SHOW:ψ" also if we realize that ψ is what we need to complete the (current stage of the) derivation; they are local goals (or subgoals). It is some β in case we are trying to proceed via $[COND]$, or any formula for $[RED]$. Thus one may enter arbitrarily many show-lines, dividing the process of proving into the realization of many simpler subgoals.

The part of a derivation beneath some S-formula will be called a subproof if it is finished (i.e. boxed), or an open subderivation if yet unfinished. Thus every open subderivation may contain U-formulae of this subderivation, S-formulae (with their subderivations) and subproofs (boxes), whereas a subproof may contain only formulae that were U-formulae before closing in a box, and other subproofs (boxes). It is convenient to note the level of nesting of every subderivation in the structure of the whole construction. We will call this parameter a *degree*, formally:

Definition 2.1

- Part of a derivation starting after the first S-line is called a subderivation of the 1st degree.
- If S-formula is put immediately after U-formula belonging to subderivation of degree n, then a subderivation introduced by this S-formula is of $n + 1$ degree.

We will say also about subproofs of degree n in case they are closed subderivations of degree n. It is easy to observe that in one derivation there may be a lot of independent subderivations of degree $n > 1$, whereas there is only one subderivation of degree 1 and in case it is closed it is just a proof. One of the possible measures of complexity of proofs is to count the number of all its subproofs. Another interesting measure is to count how many subproofs are nested inside one another. We say that a proof is of *depth* k, if the maximal degree of its subproof(s) is k. A *current subderivation* is this part of a derivation where we actually add conclusions from the application of inference rules.

Assumptions are put down immediately after S-formulae as the next lines (the first U-line of a given subderivation). Their introduction is regulated by two rules (or rather instructions since the term "rule" is not used here in the technical sense):

(a) if the last line is S-formula β, then we may add $-\beta_1$ as a conditional assumption of current subderivation

(b) if the last line is S-formula φ, then we may add $-\varphi$ as an indirect assumption of current subderivation.

The interesting thing is the way we complete (or close) subderivations in KM. One must remember that S-formula is in a sense not a part of a derivation; it simply displays the immediate goal. We cannot use such a formula as a premise of any inference rule but it can be turned into ordinary formula if this goal is reached through some subsidiary derivation. There are two such rules of closing a subderivation[13]:

$[COND]$: If β is an S-formula opening a subderivation and β_2 has appeared as an U-formula of this subderivation, then we can close it.

$[RED]$: If $\perp$ has appeared as an U-formula in a subderivation, then we can close it (in this case the shape of an S-formula is of no importance).

After completion of a subderivation it is boxed and the prefix is cancelled (which looks like SHØW). In this way we make evident that the attempted goal has been realized. From now on formulae in a box are not available, on the other hand, a formula with cancelled prefix "Show:" is not more an S-formula. If it was S-line opening $k+1$ degree subderivation, then it would become the last U-formula of outer subderivation of k-degree. Shortly, – S-formulae and boxed formulae are inactive but formula with cancelled SHOW is active. Obviously, no subproof with noncancelled S-lines can be completed. It means that within an open subderivation we can start many new subderivations but all such nested subderivations must be completed first. In fact, the formulation of rules forbids a closure of a subderivation of k-degree if there are some S-lines in this subderivation because each of them enters a subderivation of a degree$> k$. Hence, if the formula required for closing a subderivation is in their scope it is not a U-formula

[13] For simplicity we keep the same names as for suitable rules in the calculus, but note that "rules" from the level of realization correspond only partially to them; cf. Section 2.5.4.

of this k-degree subderivation. In result, each initiated subgoal must be realized before we finish the whole proof.

One may observe that a notion of U-formula as we used it before is imprecise. Due to the dynamic character of proof construction in KM we should rather say what is *U-formula at some stage of a derivation.* It may be defined precisely as any formula which at this stage is neither S-formula nor in a box. The set of U-formulae (S-formulae) of the whole derivation (or some subderivaton) $\mathcal{D}$ at some stage will be designated by U($\mathcal{D}$) (S($\mathcal{D}$), respectively). Notice that the number of formulae of U($\mathcal{D}$) and S($\mathcal{D}$) changes as the derivation proceeds; every application of inference rule increases U($\mathcal{D}$) while every application of $[COND]$ or $[RED]$ usually decreases U($\mathcal{D}$). It is so because we are boxing the last subderivation and the only new element of U($\mathcal{D}$) is the the last S-formula. We said "usually decreases" since it may be that in the closed subproof there was only one U-formula; in such cases an application of a proof construction rule does change U($\mathcal{D}$) but not its cardinality. The use of both proof construction rules in KM may be rendered by the following diagrams:

$$\begin{array}{ll} & \Gamma \\ i & \text{SHØW: } \beta \\ i+1 & \qquad \boxed{\begin{array}{l} -\beta_i \\ \vdots \\ \beta_j \end{array}} \\ k & \end{array} \qquad\qquad \begin{array}{ll} & \Gamma \\ i & \text{SHØW: } \varphi \\ i+1 & \qquad \boxed{\begin{array}{l} -\varphi \\ \vdots \\ \bot \end{array}} \\ k & \end{array}$$

where there is no show-lines in a box.

2.5.3 Derivations

Although we have introduced KM in rather informal way, paying attention rather to readability, the concept of a derivation and a proof may be encountered with formal definition.

Definition 2.2 (Derivation of φ)

A. "SHOW:φ" is a derivation of φ

B. Let $\mathcal{D}$ be a derivation of φ, then:

1. "$\mathcal{D}\oplus$SHOW:ψ" is a derivation of φ
2. "$\mathcal{D} \oplus \bot$" is a derivation of φ, provided $\{\psi, \neg\psi\} \subseteq$U($\mathcal{D}$)

3. "$\mathcal{D} \oplus \psi$" is a derivation of φ, provided $\neg\neg\psi \in$U($\mathcal{D}$)
4. "$\mathcal{D} \oplus \neg\neg\psi$" is a derivation of φ, provided $\psi \in$U($\mathcal{D}$)
5. "$\mathcal{D} \oplus \alpha_i$" is a derivation of φ, provided $\alpha \in$U($\mathcal{D}$)
6. "$\mathcal{D} \oplus \alpha$" is a derivation of φ, provided $\{\alpha_1, \alpha_2\} \subseteq$U($\mathcal{D}$)
7. "$\mathcal{D} \oplus \beta_j$" is a derivation of φ, provided $\{\beta, -\beta_i\} \subseteq$U($\mathcal{D}$)
8. "$\mathcal{D} \oplus \beta$" is a derivation of φ, provided $\beta_i \in$U($\mathcal{D}$)
9. "$\mathcal{D} \oplus -\beta_1$" is a derivation of φ, provided $\mathcal{D} = \mathcal{D}'\oplus$SHOW:$\beta$
10. "$\mathcal{D} \oplus -\psi$" is a derivation of φ, provided $\mathcal{D} = \mathcal{D}'\oplus$SHOW:$\psi$
11. If $\mathcal{D} =$ "$\mathcal{D}'\oplus$SHOW:$\psi\oplus\mathcal{D}''\oplus\bot$" is a derivation of φ, then "$\mathcal{D}'\oplus$SHØW:$\psi\oplus$ [$\mathcal{D}'' \oplus \bot$]" is a derivation of φ, provided S($\mathcal{D}''$)=$\varnothing$
12. If $\mathcal{D} =$ "$\mathcal{D}'\oplus$SHOW:$\beta\oplus\mathcal{D}''\oplus\beta_2$" is a derivation of φ, then "$\mathcal{D}\oplus$SHØW:$\beta\oplus$ [$\mathcal{D}'' \oplus \beta_2$]" is a derivation of φ, provided S($\mathcal{D}''$)=$\varnothing$

C. Nothing more counts as a derivation of φ.

$\mathcal{D} \oplus \mathcal{D}'$ stands for the concatenation of two parts of a derivation, [$\mathcal{D}$] means that $\mathcal{D}$ is a subproof (is boxed); ψ denotes any formula, $\mathcal{D}'$ and $\mathcal{D}''$ may be empty sequences. Points 2.–8. state the conditions for the application of inference rules, points 9. and 10. – for entering assumptions, points 11. and 12. – for closure of subproofs.

We are now in a position to define a proof of φ.

Definition 2.3 (Proof of φ) A derivation $\mathcal{D}$ of φ with S($\mathcal{D}$) = $\varnothing$ and U($\mathcal{D}$) = $\{\varphi\}$ is a *proof* of φ, otherwise a derivation is open.

It is easy to extend the definition of a derivation to cover also deducibility of conclusions from premises. It is enough to replace the phrase "a derivation of φ" by "a derivation of φ from Γ" throughout all the definition and formulate part A. as follows:

A'. "$\chi_1 \oplus ... \oplus \chi_k\oplus$SHOW:$\varphi$" is a derivation of φ from Γ, where $\Gamma = \{\chi_1, ..., \chi_k\}$

Clearly in such a case the proof of φ from Γ has $\mathrm{U}(\mathcal{D}) = \Gamma \cup \{\varphi\}$. The alternative (and more general because finiteness of Γ is not assumed) solution is to introduce the new point 1. in part B. (and change the numbering of the remaining points):

1. "$\mathcal{D} \oplus \chi$" is a derivation of φ from Γ, provided $\chi \in \Gamma$

Below there is a sample of a proof in KM. It is a particular case of the schema of a thesis which was analyzed in Section 2.4.1 in the context of tree-proofs. Let $\varphi_i := p_i$, $\psi := q$, $\chi_i := r_i$, then for $n = 3$ we have the following formula: $\varphi \to (q \to r_1 \wedge r_2 \wedge r_3)$, where $\varphi := (p_1 \wedge (q \wedge p_1 \to r_1)) \wedge (p_2 \wedge (q \wedge p_2 \to r_2)) \wedge (p_3 \wedge (q \wedge p_3 \to r_3))$

1	SHØW: $\varphi \to (q \to r_1 \wedge r_2 \wedge r_3)$	$[12, COND]$
2	φ	*ass.*
3	$p_1 \wedge (q \wedge p_1 \to r_1)$	$(2, \alpha E)$
4	$p_2 \wedge (q \wedge p_2 \to r_2)$	$(2, \alpha E)$
5	$p_3 \wedge (q \wedge p_3 \to r_3)$	$(2, \alpha E)$
6	p_1	$(3, \alpha E)$
7	$q \wedge p_1 \to r_1$	$(3, \alpha E)$
8	p_2	$(4, \alpha E)$
9	$q \wedge p_2 \to r_2$	$(4, \alpha E)$
10	p_3	$(5, \alpha E)$
11	$q \wedge p_3 \to r_3$	$(5, \alpha E)$
12	SHØW: $q \to r_1 \wedge r_2 \wedge r_3$	$[20, COND]$
13	q	*ass.*
14	$q \wedge p_1$	$(6, 13, \alpha I)$
15	$q \wedge p_2$	$(8, 13, \alpha I)$
16	$q \wedge p_3$	$(10, 13, \alpha I)$
17	r_1	$(7, 14, \beta E)$
18	r_2	$(9, 15, \beta E)$
19	r_3	$(11, 16, \beta E)$
20	$r_1 \wedge r_2 \wedge r_3$	$(17, 18, 19, \alpha I)$

The example illustrates two things: the convention of setting proofs in KM and our previous remarks concerning the economy of linear proofs (when compared with tree ones). As for the first, in the rightmost column (justification column) we state in every line the numbers of premises and the name of the applied rule; *ass.* means assumption. As for the second question, one may easily note that despite the value of n, the length of a proof will be $5 + 5n$, and its depth (=2) is rigid.

2.5.4 The Original Formulation of KM

In our presentation of KM we departed in many ways from the original system of [159].[14] We proposed a bit different set of rules which is redundant but the original KM system is in some respects even richer. It contains additional rule of direct proof construction $[DIR]$ which allows us to close current subproof if it contains U-formula identical with the last (opening this subproof) S-formula. We dispense with $[DIR]$ as a rule specific for KM but not very characteristic for ND systems in general. It is quite obvious that every application of $[DIR]$ is eliminable. There are three cases:

- Subproof closed by $[DIR]$ has no assumption; we may rewrite it adding as the first line an indirect assumption warranting inconsistency with the last formula, and change the justification of a subproof completion on $[RED]$.
- Subproof closed by $[DIR]$ contains an indirect assumption; everything remains uchanged except the justification on $[RED]$.
- Subproof closed by $[DIR]$ contains a conditional assumption and S-formula is some β; since the last U-formula is β again, and the assumption is $-\beta_1$, then we may add β_2 as the additional line by the application of (βE) and change the justification on $[COND]$.

Although our description of the realization of KM slightly differs from the original version, its two important features are preserved. First, it is a characteristic feature of KM that introduction of assumptions in KM is optional; we are not forced to do that. Second, the way of completing a subproof is independent of the form of its introduction, e.g. we may start with a conditional assumption (trying to do conditional proof) but finish with $\bot$ and close a subproof by $[RED]$. That is why, when showing the eliminability of $[DIR]$, we had to consider three cases of its application instead of one.

Separating, on the level of realization, the elements rigidly connected on the level of calculus, makes KM a particularly flexible and user's friendly tool of proof search. Other systems of ND in Jaśkowski's format are usually formulated in such a way that the manner we open and close a subproof must be specified in advance. It is one more argument for introducing a

[14]The version from [158] differs in even more respects; e.g. there are no boxes, no specified inference rules for connectives, no $[RED]$.

distinction between a calculus and its realization. Both, our description of a calculus and the diagrams of realization of $[COND]$ and $[RED]$ were more rigid (e.g. explicit occurrence of assumptions in both diagrams) than KM indeed requires. In fact, when we start rather with the description of realization of KM (even without $[DIR]$) and then try to precisely reconstruct the calculus involved, we obtain a richer set of proof construction rules containing additionally:

$[COND']$	If $\Gamma \vdash \beta_2$, then $\Gamma \vdash \beta$
$[COND'']$	If $\Gamma, -\beta \vdash \beta_2$, then $\Gamma \vdash \beta$
$[RED']$	If $\Gamma \vdash \bot$, then $\Gamma \vdash \varphi$
$[RED'']$	If $\Gamma, -\beta_1 \vdash \bot$, then $\Gamma \vdash \beta$

The eliminability of these rules by means of rigid forms of $[COND]$ and $[RED]$ may be easily established.

Virtues of KM

Before proceeding further and introducing a variety of ND systems based on KM, it is worthwhile pointing out these features of the realization that make it, in our opinion, superior to others. To be brief: KM is more dynamic, heuristically oriented and flexible. All these features together greatly improve the didactic value of the system.

The distinction of the two types of lines/formulae with changing status makes KM more dynamic than other systems. Proof search process is divided into a sequence of partially isolated fragments which is similar to usual mathematical practice of proving a theorem by means of auxiliary lemmata. The hierarchy of parts is well controlled by the cancelling/boxing technique. Moreover, after completion, the structure of the whole proof is represented better than in other L-systems (perhaps not worse than in T-proofs).

The application of S-lines really helps in proof search. First, it augments the control over what is the current goal of a derivation. Second, it resembles usual practice of argumentation, where a conclusion (thesis) is usually stated first and then a justification is provided. Finally, it has a great heuristic value in the analysis of ordinary arguments which are usually enthymematic, and an honest interpretation should elucidate what is missing. The apparatus of KM helps to provide the reconstruction of missing premises in the proof search. In case we fail at some stage of the construction of a proof for some argument, the last S-line suggests what additional premise we need to complete a (sub)proof. One may compare such a premise with original data provided by an author of argument and evaluate whether addition of extra information is compatible with the presented set of beliefs. So, such additional data provided by S-lines may help either to show that the analysed argument is valid or to provide a counterexample to it.[15]

Finally, the flexibility of proof construction mentioned above is a consequence of independence of assumption introduction and the form of derivation closing. Additional redundancy of primitive rules strengthens this feature. As a result, in KM one may freely change the strategy of proof search during the construction of a derivation. If we fail with one strategy (signalled by the shape of S-formula or a type of assumption), we may continue with another and not necessarilly start a new subderivation for that.

2.6 Adequacy of KM

We are not going to present the completeness proof of KM for **CPL**. It is enough to prove all axioms of any complete set with (MP) as the only rule, and we have a result indirectly, since (MP) is covered by (βE). We leave this task as a simple exercise to the reader.

The difficulty ND systems usually give rise, particularly in Jaśkowski's format, lies in their soundness proof which require some "translation" of all this additional machinery into the semantics of a suitable logic. Jaśkowski [157] established some standard form of soundness proof extensively used by many logicians in the respective proofs for ND-systems. Shortly, for each prefixed formula we build its development, which is a descending implication with suppositions for each number in the prefix as antecedents and formula itself as the succedent. For example, the development of a prefixed formula $i_1....i_n.\varphi$ is $\psi_1 \to (\psi_2 \to \ldots(\psi_n \to \varphi)\ldots)$, where each ψ_k, $1 \leq k \leq n$ is an assumption introduced with addition of successive i_k to already existing prefix, i.e. we have $i_1....i_k.S\psi_k$ above $i_1....i_n.\varphi$ in the proof. Then Jaśkowski has proved that the development of a formula in each line is a thesis of axiomatic system, obtaining indirectly the proof of soundness of his ND. On the basis of such a translation we can also prove soundness directly, showing that the first line of a proof is valid $(\varphi \to \varphi)$ and that all rules expressed in terms of developments are validity preserving.

This manner of showing soundness is very popular. There are many variants of it (cf. e.g. Fitting [95], or the original proof of Kalish and

[15] One may notice that in some ND systems without explicit use of S-lines, their effect is simulated by some meta-system devices, e.g. in Pollock's [218] or in Gabbay [99], where "almost" KM system is used.

Montague from [158]) but essentially in such a proof we proceed by turning formulae of any proof into sequents (we add a record of active assumptions), and then by showing that (such modified) rules are normality preserving. This is obvious and natural for S-systems in Suppes' format, but for F-systems it is not very natural. Instead of showing directly that F-system S is sound, we first change it into some equivalent S-system S', and then show soundness of S'. Below we propose a general schema of proving soundness for any ND system in Jaśkowski's format which is in a sense more direct. It allows separating of this part of soundness proof which is concerned with proof construction as such, from showing the correctness of inference rules. Generality means that in case of extensions we need only to check that additional rules are correct.

We start with some auxiliary results characterizing KM rules in semantic terms introduced in Chapter 1. Standard and easy proof is left to the reader.

Lemma 2.1 *Every inference rule is* **CPL**-*normal*

Lemma 2.2 *Every proof construction rule is normality preserving in* **CPL**

We introduce two notions concerning correctness of subderivations.

Let $\mathcal{D}$ be any proof and consider any subproof $\mathcal{D}'$ of degree $n > 0$ contained in it. We say that this subproof is *justified* iff, $\Gamma, \psi_1 \models \psi_n$, where ψ_1 is the first and ψ_n is the last formula of $\mathcal{D}'$, and Γ is the set (possibly empty) of all formulae of $\mathcal{D}$ above this subproof that were U-formulae at the stage immediately before $\mathcal{D}'$ was closed. For any subproof of degree i, let $\langle\psi_1, ..., \psi_n\rangle$ denote the sequence of all formulae inside the box that were U-formulae of this subproof immediately before it was closed.

The proof proceeds by double induction: on the depth of $\mathcal{D}$ and on the length of its subproofs. Let the depth of $\mathcal{D}$ be k; we show by reverse induction on k that every subproof is justified. In the basis we consider all subproofs of degree k, i.e. with no subproofs inside. By induction on the length of any such subproof we show that it is justified. Let $\langle\psi_1, ..., \psi_n\rangle$ be the sequence of all formulae inside the box, and Γ be the set of all U-formulae above this subproof. By induction on n, we may show that $\Gamma, \psi_1 \models \psi_i$ $(1 \leq i \leq n)$.

Basis: $i = 1$, so ψ_1 is an assumption, and the thesis follows by reflexivity and monotonicity of $\models$. Assume the thesis holds for any i such that $i < k \leq n$; we will show that $\Gamma, \psi_1 \models \psi_k$. Either all premises used in the deduction of ψ_k belong to $\Gamma \cup \{\psi_1\}$, or at least one of them is ψ_i. In the former case it holds by monotonicity and Lemma 2.1, in the latter case by the same lemma, induction assumption and transitivity of $\models$. So every subderivation of degree k is justified.

Now assume that every subproof of degree $i + 1 \leq k$ is justified, we show that every subproof of degree i is justified. The proof goes as for basis, by induction on the length of the considered subproof, but this time as elements of this subproof we may also have some formulae that were previously S-formulae initiating subproofs of degree $i + 1$. Let ψ_i be the first such formula. By induction assumption we have $\Gamma', \chi_1 \models \chi_n$, where χ_1 is an assumption and χ_n the last line of this subproof of degree $i + 1$. By Lemma 2.2, $\Gamma' \models \psi_i$ because this subproof was completed by some of the proof construction rules, and both rules are normality preserving. Γ' is just $\Gamma \cup \{\psi_1, ...\psi_{i-1}\}$, but we have already proved that $\Gamma, \psi_1 \models \psi_j$, for each $j \leq i - 1$, so by transitivity we have $\Gamma, \psi_1 \models \psi_i$ which guarantees that theorem holds in this case, as well. This step is repeated for subsequent ψ_i that were previously S-formulae, so every subproof containing nested subproofs is also justified. In particular, it holds for the only subproof of degree 1 and thus we have:

Theorem 2.1 (Soundness of KM for CPL) *If* $\Gamma \vdash \varphi$, *then* $\Gamma \models \varphi$

2.7 ND for First-Order Logic

2.7.1 Gentzen Systems

Both the systems of Jaśkowski and Gentzen contained also the rules for quantifiers. In fact, Jaśkowski [157] provided the first formalization of inclusive logic, so we describe it when ND for free logic will be considered. Gentzen [109] gives an adequate set of rules for **CQL** (and for intuitionistic **QL**) which is duplicated in many ND systems nowadays.

($\forall E^p$) $\forall x\varphi$ / $\varphi[x/a]$, where a is any parameter (or term)
($\forall I^p$) $\varphi[x/a]$ / $\forall x\varphi$, provided a is a parameter with no occurrence in undischarged assumptions and premises
($\exists I^p$) $\varphi[x/a]$ / $\exists x\varphi$, where a is any parameter (or term)
[$\exists E^p$] If $\Gamma \vdash \exists x\varphi$ and $\Delta, \varphi[x/a] \vdash \psi$, then $\Gamma, \Delta \vdash \psi$, provided a is a parameter new to φ, ψ and the set of undischarged assumptions Γ, Δ

The last rule in Gentzen's tree realization looks as follows:

$$\dfrac{\begin{matrix}\Gamma \\ \vdots \\ \exists x\varphi\end{matrix} \qquad \begin{matrix}[\varphi[x/a]], \Delta \\ \vdots \\ \psi\end{matrix}}{\psi}$$

One of the characteristic features of Gentzen's rules is the asymmetry of $(\forall I^p)$ and $[\exists E^p]$ which is not present in his SC (cf. a presentation of SC in the next Chapter). One might expect that either both should be proof construction or inference rules. We will say more about it later.

The next feature is that instead of free variables, the special category of quasi- or arbitrary- or parametric names[16] is introduced. This simplifies the matter as one is not bound to make troublesome distinctions while dealing with proper substitution (cf. Chapter 1), which is certainly an advantage in the context of teaching logic. Such rules also correspond well to substitutional semantics. But sometimes using free variables can make the formulation of the set of rules even easier, as is the case with rules of KM. Moreover, the category of such names is somewhat ambiguous and probably, as e.g. Pelletier [205] has noticed, not very consequently used. There are different senses of arbitrariness involved in $(\forall I^p)$ and in $[\exists E^p]$. In the former we mean any object with no established properties except satisfaction of φ, whereas in the latter we refer rather to some specific but unidentified (unnamed) object satisfying φ.

Because of all that we prefer to use rules with no parameters in the main system, but we will also present variants with rules based on parameters. The names of such rules will be distinguished by a superscript p; exactly as in the original rules of Gentzen formulated above.

In [110] Gentzen introduced suitable rules for S-system of ND:

$(\forall E^p_S)$ $\Gamma \Rightarrow \forall x\varphi \;/\; \Gamma \Rightarrow \varphi[x/\tau]$, where τ is any term
$(\forall I^p_S)$ $\Gamma \Rightarrow \varphi[x/a] \;/\; \Gamma \Rightarrow \forall x\varphi$, provided a is a parameter with no occurrence in undischarged assumptions and premises
$(\exists I^p_S)$ $\Gamma \Rightarrow \varphi[x/\tau] \;/\; \Gamma \Rightarrow \exists x\varphi$, where τ is any term
$(\exists E^p_S)$ $\Gamma \Rightarrow \exists x\varphi;\; \Delta, \varphi[x/a] \Rightarrow \psi \;/\; \Gamma, \Delta \Rightarrow \psi$, provided a is a parameter new to $\Gamma, \Delta, \varphi, \psi$

[16] There is no risk of confusion with the notion concerning formulae in sequents so, in this book, we will call them shortly parameters in accordance with the terminology of Fitting.

Also in this system we have an asymmetry in treating $\forall$ and $\exists$, which is not present in his SC. Two rules admit more general category of terms τ as the system is defined for arithmetic and there are individual constants representing natural numbers in the language.

Basically this set of rules is commonly applied in textbooks using Suppes' format of ND. But when linear proofs are introduced it is more convenient to state a suitable rule of $\exists$ elimination in the calculus as the three-premises rule of the form:

$(L\exists E^p_S)$ $\Gamma \Rightarrow \exists x\varphi;\ \varphi[x/a] \Rightarrow \varphi[x/a];\ \Delta, \varphi[x/a] \Rightarrow \psi \ /\ \Gamma, \Delta \Rightarrow \psi$, provided a is a parameter new to $\Gamma, \Delta, \varphi, \psi$

In fact, Suppes in [266] applies a different form of elimination of $\exists$ which does not correspond to Gentzen's solution because in terms of F-systems Gentzen's variant is a proof construction rule, whereas Suppes' variant is an inference rule. The above rule is probably due to Lemmon [174] (hence the prefix L in the name); the original rule of Suppes was:

$(S\exists E^p_S)$ $\Gamma \Rightarrow \exists x\varphi \ /\ \Gamma \Rightarrow \varphi[x/a_{y_1,...,y_k}]$, provided a is a parameter new to Γ, φ, and $y_1, ..., y_k$ are all free variables of φ

Structurally, this rule is simpler but it introduces a form of skolemization and additionally leads to considerable complications in formulation of the correct rule of $\forall$ introduction in Suppes' system. We do not present his variant of $(\forall I^p_S)$ but postpone a discussion of troubles connected with $\exists$ elimination as an inference rule to the next subsection.

2.7.2 Kalish/Montague Rules for CQL

The original set of quantifier rules due to Kalish and Montague is in a sense opposite to Gentzen rules. First, it is defined on pure language (only variables – including free occurrences). Secondly, the elimination of $\exists$ is an inference rule, whereas the introduction of $\forall$ is a proof construction rule. We also admit substitution of individual names for variables, so by $\varphi[x/\tau]$ we mean either a proper substitution of variable τ or of any name τ for x in φ. Let us display the rules below:

$(\forall E)$ $\forall x\varphi \ /\ \varphi[x/\tau]$
$(\exists E)$ $\exists x\varphi \ /\ \varphi[x/y]$, provided y is a new variable in a derivation
$(\exists I)$ $\varphi[x/\tau] \ /\ \exists x\varphi$
$[UNIV]$ If $\Gamma \vdash \varphi$, then $\Gamma \vdash \forall x\varphi$, provided $x \notin VF(\Gamma)$

Note that in $(\exists E)$ a variable y is not at all admitted; neither as free nor as bound, neither in U- nor in S-formulae stated in a derivation. Condition for $[UNIV]$ is more relaxed; we make a constraint only with respect to free occurrences of x and only in U-formulae but Γ covers not only assumptions but all U-formulae above S-line with $\forall x\varphi$. Although in practice KM admits subderivations without assumptions, this last rule does not require an assumption even in theory. Its formulation for KM may be stated as follows:

$[UNIV]$ Let $\forall x\varphi$ be a show-formula of k-degree subderivation, then we can close this subderivation, if φ has appeared as an usable-formula in this subderivation, provided $x \notin VF(\Gamma)$, where Γ is the set of all U-formulae above S-formula $\forall x\varphi$.

It is a simple matter to enrich the definition of a derivation stated in Section 2.5.3 with additional 4 clauses corresponding to the respective rules so we leave it to the reader. When extending the definition of a derivation to the case of deducibility from premises we follow the first solution stated therein, i.e. we make a stipulation that all premises must be stated before the first show-line is introduced. Note that if in the formulation of derivation from premises we will apply the second approach, i.e. a possibility of introducing a premise as any line in a derivation, we could prove $Ax \vdash \forall x Ax$, something which we certainly do not want to obtain. Fortunately, such deductions are impossible if the premise is stated above the show-line.

It is routine to prove all axioms and show derivability of rules of H-**CQL** except perhaps $(\exists)$. In this case we have the following proof of $\exists x\varphi \rightarrow \psi$ on the basis of the assumption that $\varphi(x) \rightarrow \psi$ is already a thesis and $x \notin VF(\psi)$

1	SHØW: $\exists x\varphi \rightarrow \psi$	$[7, COND]$
2	$\exists x\varphi$	$ass.$
3	$\varphi[x/y]$	$(2, \exists E)$
4	SHØW: $\forall x(\varphi(x) \rightarrow \psi)$	$[5, UNIV]$
5	$\varphi(x) \rightarrow \psi$	$thesis$
6	$(\varphi(x) \rightarrow \psi)[x/y]$	$(4, \forall E)$
7	ψ	$(3, 6, \beta E)$

In the proof y must be new so $y \notin VF(\psi)$ exactly as x. In result $(\varphi(x) \rightarrow \psi)[x/y] = \varphi[x/y] \rightarrow \psi$ and the inference in line 7 is correct. Universal closure of $\varphi(x) \rightarrow \psi$ in line 4 is justified since the formula with free x is a thesis, not an arbitrary premise which must be stated before the first show-line.

So, on the basis of completeness of H-**CQL**, we obtain:

Theorem 2.2 (Completeness) *If* $\Gamma \models_{CQL} \varphi$ *then* $\Gamma \vdash_{KM-CQL} \varphi$

Remark 2.1 As for the question of asymmetry of the rules for introduction of $\forall$ and elimination of $\exists$ which is characteristic for systems of Gentzen and for KM as well, one should note that there are also systems where it is avoided. For example, Suppes' S-system mentioned in the preceding subsection is of this kind. In case of F-systems, ND of Quine [226], or of Słupecki, Borkowski [54] have all the rules for quantifiers formulated as inference rules. In fact, Gentzen's rule for elimination of $\exists$ may be also transformed into inference rule, if we replace a subderivation by suitable implication; the result is:

$\exists x\varphi, \varphi[x/a] \rightarrow \psi \ / \ \psi$, provided a is a new parameter to φ, ψ and to all undischarged assumptions.

On the other hand, systems of Fitch [89] or Thomason [274] offer two proof construction rules: Gentzen's-like $[\exists E]$ and a kind of $[UNIV]$. Yet another combination is present in ND system due to Negri and von Plato [193], where introduction rules are inference rules and elimination rules are proof construction rules for both quantifiers, cf. Section 4.1.1. ♣

The interplay of $\exists$ elimination and $\forall$ introduction leads to troubles in the formulation of correct rules, particularly evident if the first is proposed as an inference rule. It is not possible to combine any known rules for quantifiers because in most cases the result would be unsound. No surprise that proper statement of $(\exists E)$ was delayed; first formulations of Copi [72] were incorrect as well as versions from later editions,[17] and even correct ones like in Quine [226] were so complicated that one can hardly call them practical and natural. For example, in Słupecki, Borkowski system one must in fact introduce hidden form of skolemization (similar as in Suppes' rule) and additional global side conditions (i.e. constraints put on the whole proof). Seen in this light, the rules of KM are really simple, which is possible due to the whole structure of the system.

[17] Actually, the problems was to state properly the restriction for $\forall$ introduction, in the presence of $(\exists E)$; a detailed account of it can be found in Pelletier [204] and in Fine [87].

2.7.3 Gentzen's Variant of KM

Systems with inference rule of $\exists$ elimination suffer from difficulty with proving soundness. It is the only inference rule which is not normal, since the conclusion does not follow from the premise. Because of that, neither standard forms of soundness proof (i.e. by transformation of a formula into a sequent with all active assumptions displayed in antecedent), nor the proof introduced in the preceding section works. One should rather use global strategies characteristic for soundness proofs for tableaux where in order to justify every line we refer to satisfiability of all formulae above (cf. Fitting [93] for application to modal ND). Yet another proof, which we omit here, is due to Kalish and Montague in [158]. For keeping KM format open for several modifications we proceed indirectly. Let us call KMG (from Gentzen) a variant of KM where $(\exists E)$ is replaced by $[\exists E]$ of the form:

$[\exists E]$ If $\Gamma, \exists x\varphi, \varphi[x/y] \vdash \psi$, then $\Gamma, \exists x\varphi \vdash \psi$,
provided y is a variable new in a derivation

On the level of realization in KMG it splits into two rules: for introduction of existential assumption, and for closing a subderivation:

(c)[18] if $\exists x\varphi$ is a U-formula and the last line is a show-line, then we may add $\varphi[x/y]$ as an existential assumption of this subderivation, provided y is new in a derivation.

$[\exists E]$ Let φ be a show-formula of k-degree subderivation, then we can close this subderivation provided φ has appeared in it as a usable-formula.

Let us emphasize again the freedom KM format admits for its proof construction. We do not have to require the presence of existential assumption in the last rule; even if it is not present, the closure is correct, being just a case of $[DIR]$. On the other hand, introduction of existential assumption does not forces us to close a subderivation by $[\exists E]$ – one may equally well close it by $[RED]$, $[COND]$ or $[UNIV]$. Such forms of completion are based on the following rules:

[18] Here and then we follow the convention of Section 2.5, where two instructions for assumption introduction were signed by (a) and (b).

1. If $\Gamma, \exists x\varphi, \varphi[x/y] \vdash \bot$, then $\Gamma, \exists x\varphi \vdash \psi$, provided y is a variable new in a derivation
2. If $\Gamma, \exists x\varphi, \varphi[x/y] \vdash \beta_2$, then $\Gamma, \exists x\varphi \vdash \beta$, provided y is a variable new in a derivation
3. If $\Gamma, \exists x\varphi, \varphi[x/y] \vdash \psi$, then $\Gamma, \exists x\varphi \vdash \forall z\psi$, provided y is a variable new in a derivation

All these variants are eliminable: 1. If we have a contradiction, then we may by $(\bot E)$ deduce a formula identical to S-formula ψ and close a subproof by $[\exists E]$. 2. If we have deduced β_2 of S-formula β, then β is also derivable by (βI) and again we close a subproof by $[\exists E]$. 3. In the last case we must insert a new S-formula $\forall z\psi$ immediately after existential assumption $\varphi[x/y]$; then the new subproof closes by $[UNIV]$ since the conditions on z are satisfied (y must be new so $y \neq z$), and the outer one closes by $[\exists E]$ since S-formula $\forall z\psi$ became usable and identical to current S-formula.

So KMG keeps a flexibility of construction characteristic for this format of ND system. In practice, when dealing with existential assumption, one may postpone the introduction of S-formula, write just "SHOW:" and complete the line later remembering simply about the constraint on variable in assumption. It is a handy way to proceed if one has no idea of what to search for at a current stage but there is an unused existential formula.

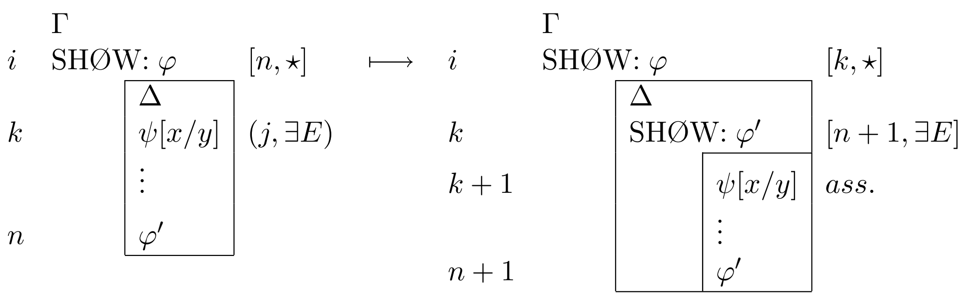
Soundness of KMG-**CQL** can be easily demonstrated. It is enough to prove that $(\forall E)$ and $(\exists I)$ are normal, and that $[UNIV]$ and $[\exists E]$ preserve normality and then to make suitable adjustments in the proof from the preceding section. We show, as an example, that $[\exists E]$ is correct and leave the rest to the reader.

Assuming that $\Gamma, \exists x\varphi, \varphi[x/y] \models \psi$ and y is not in Γ, φ and ψ, we must show that $\Gamma, \exists x\varphi \models \psi$. Consider $\mathfrak{M}, a \vDash \Gamma, \exists x\varphi$, hence $\mathfrak{M}, a \vDash \Gamma$ and $\mathfrak{M}, a \vDash \exists x\varphi$. Since $\models \exists x\varphi \leftrightarrow \exists y\varphi[x/y]$, then $\mathfrak{M}, a \vDash \exists y\varphi[x/y]$. It means that $\mathfrak{M}, a^y_o \vDash \varphi[x/y]$ for some o in the domain of $\mathfrak{M}$. But then, since y is not free in Γ and $\exists x\varphi$ it follows that $\mathfrak{M}, a^y_o \vDash \Gamma, \exists x\varphi$ (cf. the fact stated in Section 1.1.4). Hence $\mathfrak{M}, a^y_o \vDash \psi$ by assumption and, since y is not free in ψ either, $\mathfrak{M}, a \vDash \psi$.

We are entitled to conclude:

Lemma 2.3 (Soundness of KMG) *If* $\Gamma \vdash_{KMG} \varphi$ *then* $\Gamma \models_{CQL} \varphi$

Now, it is also easy to prove the soundness of KM by showing that every proof in KM may be transformed into a proof in KMG. In fact, it is sufficient to demonstrate that $(\exists E)$ is admissible in KMG since other rules are the same in both systems. We do it by induction on the number of $(\exists E)$ applications in a proof in KM. The case of 0 applications is trivial. Let us assume that we have just one application of this rule in some subproof. We may transform this part of a proof dispensing with $(\exists E)$ in favor of $[\exists E]$. This is displayed in the following schema:



where: $\star \in \{COND, RED, UNIV\}$, φ is a suitable S-formula and φ' suitable formula closing a subproof, $\exists x\psi \in \Gamma \cup \Delta \neq \varnothing$ (although Γ or Δ alone may be empty) in line j $(1 \leq j < i$ or $i < j < k)$.

It is enough to check that if the input-subproof is correct then the transformed output-subproof is also correct. Note that by the definition of proof construction rules y cannot occur in φ' as it would also occur in φ while it must be new when introduced by $(\exists E)$. Hence the application of $[\exists E]$ in output-subproof is correct in KMG. For induction step we apply exactly the same transformation. So by the above schema we have that:

Lemma 2.4 *If* $\Gamma \vdash_{KM} \varphi$ *then* $\Gamma \vdash_{KMG} \varphi$

This lemma implies both completeness of KMG (by Theorem 2.2.) and soundness of KM (by Lemma 2.3).

2.7.4 KM for Free Logic

The apparatus of KM, or KMG, is very convenient for the easy-going formalization of free logic. The first ND system for inclusive logic was in fact formulated by Jaśkowski [157]. He used a convention of introducing as assumptions not only formulae but also variables (the only terms of his system). So he had a rule to the effect of starting a subproof with the so-called term supposition Tx and two rules for $\forall$

$(J\forall E)$ $\forall x\varphi, Ty \;/\; \varphi[x/y]$
$[JUNIV]$ If $\Gamma, Tx \vdash \varphi$, then $\Gamma \vdash \forall x\varphi$, provided $x \notin VF(\Gamma)$

Most of the ND systems for **FQL** proposed later are based on the application of proof construction rule for elimination of $\exists$ and use parameters. We rather introduce a system being a minimal modification of the original KM from Section 2.7.2. The following free variants of KM rules were first introduced in Indrzejczak [141]; they look as follows:

$(F\forall E)$ $\forall x\varphi, E\tau \;/\; \varphi[x/\tau]$
$(F\exists E)$ $\exists x\varphi \;/\; Ey \wedge \varphi[x/y]$,
provided y is a new variable in a derivation
$(F\exists I)$ $\varphi[x/\tau], E\tau \;/\; \exists x\varphi$
$[FUNIV]$ If $\Gamma, Ex \vdash \varphi$, then $\Gamma \vdash \forall x\varphi$, provided $x \notin VF(\Gamma)$

Let us call KM' a system obtained by the addition of these rules to the propositional basis of KM. On the level of the realization the statement of $[FUNIV]$ is the same as for $[UNIV]$; the assumption Ex need not be mentioned at all (the reasons for that will be explained below). But we must add an explicit instruction for the introduction of it into a proof:

(d) if the last S-formula is $\forall x\varphi$ and x has no free occurrence in U-formulae above this line, then we may add Ex as an existential assumption of this subderivation.

One may observe in this context that proof construction rules for $\forall$ introduction, like $[UNIV]$, are very much in the spirit of free logic. It is a natural composition of conditional and universal proof, whereas any form of inference rule for $\forall$ introduction forces us to use an implication as a premise. Moreover, in free logics we avoid this way introducing subproofs without assumptions, a practice which may seem odd to some practitioners of ND systems.

Completeness of this system may be shown in a similar way as we did for KM-**CQL** by showing that all axioms are provable and all rules are derivable. We leave it to the reader. In showing soundness we also encounter the same problem as in the classical setting; non-normality of $(F\exists E)$. It is not surprising that we apply the same strategy and introduce KMG' variant with the suitable proof construction rule:

$[F\exists E]$ If $\Gamma, \exists x\varphi, Ey \wedge \varphi[x/y] \vdash \psi$, then $\Gamma, \exists x\varphi \vdash \psi$,
provided y is a new variable in a derivation

All the results concerning soundness of KMG' with respect to **FQL** and admissibility of $(F\exists E)$ in this system, which together yield adequacy of both KM' and KMG' with respect to **FQL**, follow the same pattern as in the classical case. The reader is asked to check it for himself.

2.7.5 Introduction of Parameters

Finally, we consider how KM (or KMG) should be modified to obtain a system involving rules with parameters. The modification of $(\forall E)$ and $(\exists I)$ is straightforward; we just take Gentzen's $(\forall E^p)$ and $(\exists I^p)$ instead. Normality of both rules is provable in exactly the same way, provided parameters are semantically treated as free variables. Similarly, instead of $(\exists E)$ we use $(\exists E^p)$ which introduces a new parameter. Now consider a variant of $[UNIV]$:

$[UNIV^p]$ If $\Gamma \vdash \varphi[x/a]$, then $\Gamma \vdash \forall x\varphi$,
where a is a new parameter to Γ and φ

Unfortunately, it does not work. Although we do not have to make a constraint on free occurrences of x, since there are no free variables (only parameters) in such a system, we may easily deduce $\exists x\varphi \rightarrow \forall x\varphi$. It is enough to have $\exists x\varphi$ in Γ, introduce $\forall x\varphi$ as a show-line, then apply $(\exists E^p)$ inside this subderivation for a as a new parameter, which is sufficient to close a box. In order to avoid such a situation one must take care of the origin of a new parameter in $\varphi[x/a]$ that closes a subderivation by $[UNIV^p]$. Let us consider some techniques that may be applied:

1. One may use a new parameter a as flagging a subderivation under S-formula $\forall x\varphi$ and require that U-formula closing this subderivation must contain exactly this parameter substituted for x. It is just a technique of Jaśkowski and many ND systems use similar solutions.

2. Bonevac [52] has made a stipulation introducing, immediately after show-line with $\forall x\varphi$, the next show-line with $\varphi[x/a]$, where a is a new parameter. If the innermost subderivation is closed (by any rule) we automatically obtain a formula closing the parent subderivation by $[UNIV^p]$. In this way we avoid additional bookkeeping device introduced in the preceding solution however at the expense of making two connected subderivations instead of one. Note that it is a modification made rather on the level of strategy of a proof search.

3. In the system for **QLI** we may, instead of flagging a subderivation by a new a, introduce $a = a$ as the first line (universal assumption) establishing which "new" parameter should be substituted for x in U-formula closing this subderivation.

4. Likewise, but instead of $a = a$ one may introduce Ea with a new a as existential assumption of this subderivation. Note that this is a variant (with parameter) of $[FUNIV]$ we have stated for free logic. But nothing changes if other rules are as for the classical first-order logic, because our free variant of $\forall$ introduction (as well as other free logic rules) is also correct in **CQL** and sufficient for making all the classical deductions. It suffices to observe that if the rule for the elimination of $\forall$ is classical, then Ea is not needed as a premise for deduction but only to fix a new parameter for universal subderivation.

5. Finally, one may introduce, immediately after S-formula $\forall x\varphi$, an (indirect) assumption $\neg\varphi[x/a]$ with a new a. It has some advantage over 3. and 4. – we may use it in **CQL** with no extension of a language or technical machinery, similarly as in 2. But it is a special form of indirect proof rather than universal derivation.

In what follows we will no longer consider solution 1. as bringing dispensable element into the technical machinery, difficult to state on the calculus level. Solution 2. has some merits since $[UNIV^p]$ is a correct rule in itself, and Bonevac's solution allows keeping it together with $(\exists E^p)$ on the level of calculus. But we must carefully introduce strategic constraints for making universal proof on the level of the realization. We are not going to enter into details (one may consult [52]) but rather focus on the remaining solutions.

Solutions 3, 4, and 5. may be grouped together as all depending on the introduction of an assumption that makes a new parameter a fixed for universal subderivation. So on the level of calculus the correct rule of $\forall$ introduction has the following form:

$[UNIV^{p\star}]$ If $\Gamma, \psi(a) \vdash \varphi[x/a]$, then $\Gamma \vdash \forall x\varphi$,
where a is a new parameter to Γ and φ

Here $\psi(a)$ is $Ea, a = a$ or $\neg\varphi[x/a]$ according to the chosen solution. The version of $[UNIV^{p\star}]$ with Ea will be called $[FUNIV^p]$ for future use (in the context of free logic). KM system based on inference rules with parameter and on some form of $[UNIV^{p\star}]$ will be called KMP.

Remark 2.2 Note that although $[FUNIV^p]$ and $[FUNIV]$ are structurally similar there is a significant difference. Existential assumption in $[FUNIV]$ is in fact not necessary for completing universal subproof. Constraints on x are stated as an extra side condition and we need this assumption only because we are in free logic and the rule of $\forall$ elimination requires such formulae as additional premises. In case of $[FUNIV^p]$ (and $[UNIV^{p\star}]$ in general) an assumption is essential just for closing universal subproof because it fixes some arbitrary parameter for this subproof. Therefore we need it (or something similar) even in a system for classical logic, where the elimination of $\forall$ does not involve existential formulae as an additional premise. It shows that although using parameters instead of free variables seems to be quite an innocent deviation, it can make serious differences when concrete systems are defined. ♣

Again on the level of realization of KMP we have a division into two rules: for introduction of suitable assumption, and for completing a subderivation:

(e) if the last S-formula is $\forall x\varphi$, then we may add $\psi(a)$ (Ea or $a = a$ or $\neg\varphi[x/a]$) as an (existential, universal, indirect) assumption of this subderivation, provided a is new in a derivation.

$[UNIV^{p\star}]$ Let $\forall x\varphi$ be a show-formula of k-degree subderivation, then we can close this subderivation provided $\varphi[x/a]$ has appeared in it as a usable-formula, where a is a parameter fixed by the assumption of this subderivation.

Note that although the introduction of suitable assumption under universal S-formula is not necessary in general, it must be introduced if we want to apply closure by $[UNIV^{p\star}]$. Completeness of KMP follows from the fact that every proof in KM-**CQL** is simulated in KMP just by suitable renaming of free variables on parameters and addition of respective assumption in every case of universal (sub)proof. We leave the details to the reader. Again, for showing soundness (but not only for that) we will introduce one more variant based on proof construction rule for the elimination of $\exists$.

2.7.6 Gentzen's Variant of KMP

It seems that in KMG with parameters (hence called KMGP) we may keep $[UNIV^p]$ and avoid nonvalid deductions. But in fact the same difficulties may appear in KMGP but on the level of representation. Let us explain the problem. If we replace $(\exists E^p)$ with a variant of $[\exists E]$ with a new parameter:

$[\exists E^p]$ If $\Gamma, \exists x\varphi, \varphi[x/a] \vdash \psi$, then $\Gamma, \exists x\varphi \vdash \psi$, provided a is a new parameter to Γ, φ, ψ

Then on the level of calculus everything is correct, i.e. both inference rules are normal and both proof construction rules are normality preserving. But now the flexibility of KM (and KMG) realization may cause troubles. The point is that the introduction of assumptions in KM is independent of the form of completing a subderivation. For example, an introduction of existential assumption does not force us to make a closure by $[\exists E^p]$. So, if we just put a parameter a instead of a variable y in a clause concerning the introduction of existential assumption, and do not make any additional restriction for closing a subderivation by $[UNIV^p]$ except a requirement of a in $\varphi[x/a]$ being new, then we have problems again. The only difference is that now $\varphi[x/a]$ would appear as an assumption inside the respective subproof, not as a conclusion of $(\exists E^p)$ application. But one should be aware that such a formulation of completion by $[UNIV^p]$ on the level of realization does not agree with respective proof construction rule from a calculus. If $\varphi[x/a]$ closing a subderivation is an assumption it does not satisfy the condition $\Gamma \vdash \varphi[x/a]$. So, the correct formulation of closure by $[UNIV^p]$ in a parametric version of KMG should read:

$[UNIV^p]$ Let $\forall x\varphi$ be a show-formula of k-degree subderivation, then we can close this subderivation, if it has $\varphi[x/a]$ as a usable-formula but not an assumption, and a parameter a does not occur in current S-formula $\forall x\varphi$ and U-formulae above it.

Although such a formulation forbids incorrect inferences it is unnecessarily complicated; much weaker restriction is sufficient. The suitable rule for introduction of $\forall$ on the level of calculus is:

$[UNIV^{p'}]$ If $\Gamma \vdash \varphi[x/a]$, then $\Gamma \vdash \forall x\varphi$, where a is a parameter with no occurrence in φ and undischarged assumptions

On the level of realization we formulate a condition for closure:

$[UNIV^{p'}]$ Let $\forall x\varphi$ be a show-formula of k-degree subderivation, then we can close this subderivation, if if it has $\varphi[x/a]$ as a usable-formula, and a parameter a does not occur in current S-formula $\forall x\varphi$ and in any undischarged assumption.

In fact, the constraints on $[\exists E^p]$ may be relaxed in a similar way, yielding the following rule:

$[\exists E^{p'}]$ If $\Gamma, \exists x\varphi, \varphi[x/a] \vdash \psi$, then $\Gamma, \exists x\varphi \vdash \psi$, where a is a parameter with no occurrence in φ, ψ and undischarged assumptions

So in both proof construction rules with parameters we have slightly weaker side condition on a. A parameter a is not required to be completely new; it must be only new to the set of active assumptions. Note that these restrictions were originally formulated by Gentzen for his T-system and followed by many authors of textbook's ND systems. Although it is difficult to show that this modification bring about some substantial profits with respect to the formalization of **QL**, we will see that in modal logics this is an essential difference between KMGP and KM. Thus eventually by KMGP we mean the system based on $[UNIV^{p'}]$, $[\exists E^{p'}]$, $(\forall E^p)$ and $(\exists I^p)$. In fact it is a set of rules characteristic for many systems, e.g. in Garson [105], with one difference only. We still keep proof construction rule for introduction of $\forall$, whereas in [105] it is an inference rule but with the same proviso (which is actually the original Gentzen's rule – see Section 2.7.1), namely:

$(\forall I^p)$ $\varphi[x/a]$ / $\forall x\varphi$, provided a is a parameter not occurring in φ and undischarged assumptions

It is straightforward that the two rules yield the same results, provided other quantificational rules are the same. In one direction: Let $\mathcal{D}$ be a derivation with at least one application of $(\forall I^p)$, we apply to every case a simple modification. If in line n, U-formula $\forall x\varphi$ was deduced by $(\forall I^p)$ from $\varphi[x/a]$ in line $k < n$, then turn n into show-line, repeat $\varphi[x/a]$ in $n+1$ (by Jaśkowski's repetition rule), close this one-line subderivation by $[UNIV^{p'}]$. Then change the numbers of later lines (if there are any) of this derivation ($n+1$ of $\mathcal{D}$ for $n+2$ in the "new" derivation, e.t.c.). Proceeding systematically we will obtain a proof which is i-lines longer, where i is the number of $(\forall I^p)$ applications in $\mathcal{D}$.

In the other direction: Let $\mathcal{D}$ be a derivation with at least one application of $[UNIV^{p'}]$ and let $\forall x\varphi$ with canceled SHOW be in line k and boxed occurrence of $\varphi[x/a]$ justifying this application of a rule be in line $n > k$. Due to the possibility of having some assumption in line $k+1$ we have three cases to consider.

- There is no assumption under S-formula $\forall x\varphi$. We simply delete show-line k and the box below. All formulae in the boxed subproof must be renumbered ($k+1$ for k, ..., n for $n-1$) and $\forall x\varphi$ added as a new line n inferred from $\varphi[x/a]$ by ($\forall I^p$). The new proof has the same length as $\mathcal{D}$ but its depth may be smaller since we eliminate some boxes.[19]

- There is an indirect assumption ($\neg\forall x\varphi$) in line $k+1$. This time we just add new lines to the box: $n+1$ contains $\forall x\varphi$ inferred by ($\forall I^p$) from line n, $n+2$ contains $\bot$ by $k+1, n+1$. The justification for the completion of the box is changed into $[RED]$ and the numeration of the remaining lines is altered accordingly (+2). The number of boxes in a proof is intact.

- In some line $i < k$ there is $\exists x\psi$ and in line $k+1$ we have an existential assumption $\psi[x/b]$. Clearly, $b \neq a$, otherwise a would be present in the undischarged assumption and the application of $[UNIV^{p'}]$ would be incorrect. We must add only one line: again, a new $n+1$ contains $\forall x\varphi$ inferred by ($\forall I^p$) from line n. It is identical to current S-formula and does not contain b so the justification for closure of the box is changed into $[\exists E^{p'}]$ and the numeration of the remaining lines is altered accordingly (+1). The number of boxes in a proof is also intact.

Thus, because KMGP is equivalent to Garson's ND system which is adequate, we have:

Theorem 2.3 (Adequacy of KMGP) *KMGP is adequate for* **CQL**

Now we may show the soundness of KMP by simulation of every proof in KMGP. This is apparently harder than in case of KM and KMG, as the proof construction rules of KMGP have essentially weaker restrictions. Fortunately, additional complications are not very hard. First, we change every application of $[UNIV^{p\star}]$ in some $\mathcal{D}$ into $[UNIV^{p'}]$. This is simple, it is enough to delete an assumption $\psi(a)$ under show-line. The result is the correct application of $[UNIV^{p'}]$ because, if a was declared as having completely fresh occurrence in $\psi(a)$, then a fortiori it is not present in

[19] But it may be the same since the depth is measured not by the number of boxes in general but by the maximal number of nested boxes, and the eliminated box may not be a member of this maximal sequence.

undischarged assumptions of the new proof. Now, we have a proof $\mathcal{D}'$ where every application of $[UNIV^{p\star}]$ is already replaced by $[UNIV^{p'}]$. Now we eliminate from $\mathcal{D}'$ all applications of $(\exists E^p)$ in favor of $[\exists E^{p'}]$. The schema of elimination is very much the same as the schema presented in Section 2.7.3 for KM and KMG, only with a instead of y. We must only check that after the transformation the new proof $\mathcal{D}''$ is really a proof in KMGP. It is enough to check that every application of $[\exists E^{p'}]$ is correct. Let us emphasize that a cannot occur in φ' (we mean φ' in the schema of elimination displayed in Section 2.7.3), which is a necessary condition for a modified proof to be correct. In cases where completion of a subproof containing the application of $(\exists E^p)$ was by $[RED]$ or $[COND]$ it is obvious. In case it is obtained by $[UNIV^{p'}]$ it is likewise impossible, since in φ' there must be some $b \neq a$ which was fixed as new by the (already deleted) assumption in the original $\mathcal{D}$. If $b = a$, then the application of $(\exists E^p)$ in the transformed proof would be incorrect because the assumption with b was above $\psi[x/a]$. So, in two steps we may transform every KMP proof into KMGP proof and by Theorem 2.3. we have:

Lemma 2.5 (Soundness of KMP) *KMP is sound for* **CQL**

Remark 2.3 One should note that the possibility of introducing more liberal restrictions for elimination of ∃ and introduction of ∀ does not depend on the presence of parameters. The necessary condition for weakening the side conditions of both rules is the fact that the elimination of ∃ is of Gentzen's type, i.e. a proof construction rule; whether we use parameters or just free variables is inessential. In fact, we are even not obliged to introduce a variable different from quantified x as long as a constraint on undischarged assumptions (and inferred formula ψ, in case of ∃ elimination) is satisfied. It seems that the first ND system of this kind (but with suitable rules defined on free variables not on parameters) may be found already in Anderson and Johnstone [6]. It follows that we could also introduce a version of KMG of this sort. We did not because we needed KMG only as a supporting system for proving soundness of KM. To this aim the replacement of only one rule (namely $(\exists E)$) and by $[\exists E]$ of such form as we stated, yields the simplest solution. ♣

2.7.7 KM with Parameters for Free Logic

For free logic we may also obtain two versions based on rules with parameters analogically as we did for **CQL**. The first one KMP' consists of the following rules:

$(F\forall E^p)$	$\forall x\varphi, E\tau \ / \ \varphi[x/\tau]$
$(F\exists E^p)$	$\exists x\varphi \ / \ Ea \wedge \varphi[x/a]$, provided a is a new parameter in a derivation
$(F\exists I^p)$	$\varphi[x/\tau], E\tau \ / \ \exists x\varphi$
$[FUNIV^p]$	If $\Gamma, Ea \vdash \varphi[x/a]$, then $\Gamma \vdash \forall x\varphi$, provided a is a new parameter to Γ and φ

All the rules are obvious parameter-counterparts of the rules stated for KM'-**FQL** in Section 2.7.4. In particular, $[FUNIV^p]$ is an instance of general $[UNIV^{p\star}]$ with Ea in place of $\psi(a)$.

KMGP' is obtained from KMP' by replacing $(F\exists E^p)$ and $[FUNIV^p]$ by the following:

$[F\exists E^{p'}]$	If $\Gamma, \exists x\varphi, Ea \wedge \varphi[x/a] \vdash \psi$, then $\Gamma, \exists x\varphi \vdash \psi$, provided a is a parameter with no occurrence in φ, ψ and undischarged assumptions
$[FUNIV^{p'}]$	If $\Gamma, Ea \vdash \varphi[x/a]$, then $\Gamma \vdash \forall x\varphi$, where a is a parameter with no occurrence in φ and undischarged assumptions

This is basically the solution of Garson [105], but again, with the introduction of $\forall$ as a proof construction, not as an inference rule, which seems to be more natural in case of free logic. So, similarly as for **CQL**, we provide a version of KMGP' which differs from KMP' not only by changing inference rule for $\exists$ elimination into proof construction rule, but also by the introduction of less demanding side conditions.

These systems bring about no specific problems except those which were discussed for versions adequate for **CQL** in the two preceding subsections. So by the analogous reasoning but tailored for **FQL** one may obtain:

Theorem 2.4 (Adequacy) *KMP' and KMGP' is adequate for* **FQL**

2.7.8 Identity

In contrast to the multiplicity of rules presented for several versions of ND for **QL** the addition of identity is unproblematic. We need an axiom schema *ID* and the rule:

(LL) $\tau_1 = \tau_2, \ \varphi \ / \ \varphi[\tau_1//\tau_2]$, where φ is atomic

It is an obvious counterpart of axiom LL from Section 1.1.4. Thus showing that strengthening of some ND system S with ID and (LL) is equivalent to a suitable axiom system S' with ID and LL is trivial if we have shown that S and S' are equivalent. The latter holds for all ND systems considered above. It is convenient to recall in this place all variants of quantificational KM systems we have introduced:

1. KM = $(\forall E) + (\exists I) + (\exists E) + [UNIV]$
2. KMG = $(\forall E) + (\exists I) + [\exists E] + [UNIV]$
3. KM' = $(F\forall E) + (F\exists I) + (F\exists E) + [FUNIV]$
4. KMG' = $(F\forall E) + (F\exists I) + [F\exists E] + [FUNIV]$
5. KMP = $(\forall E^p) + (\exists I^p) + (\exists E^p) + [UNIV^{p\star}]$
6. KMGP = $(\forall E^p) + (\exists I^p) + [\exists E^{p'}] + [UNIV^{p'}]$
7. KMP' = $(F\forall E^p) + (F\exists I^p) + (F\exists E^p) + [FUNIV^p]$
8. KMGP' = $(F\forall E^p) + (F\exists I^p) + [F\exists E^{p'}] + [FUNIV^{p'}]$

An addition of ID and (LL) to any of the above systems provides an adequate ND formalization of suitable logic with identity. Moreover, in case of four systems for free logic (KM', KMG', KMP', KMGP') one may also add an axiom (1.4) $\exists xEx$ which yields formalizations of free logics excluding empty domains.

Remark 2.4 For those who dislike axioms in ND systems, the original solution of Kalish and Montague from [158] may seem more adequate. They use two inference rules for identity of the form:

$(ID1)$ $\forall x(x = \tau \to \varphi) \;/\; \varphi[x/\tau]$
$(ID2)$ $\varphi[x/\tau] \;/\; \forall x(x = \tau \to \varphi)$

Also, instead of the axiom (1.4), one may add to ND for **FQL** the following rule:

$(\forall\exists)$ $\forall x\varphi \;/\; \exists x\varphi$

One may easily prove for KM' (or any other free version of the four considered) that $(\forall\exists)$ is derivable by (1.4) and that (1.4) is provable by $(\forall\exists)$ so these two solutions are equivalent. ♣

Chapter 3

Other Deductive Systems

The Chapter provides a set of preliminary notes to the next one, where several forms of extended ND systems are discussed. These nonstandard forms of ND are strongly based on solutions occuring in different kinds of deductive systems. Therefore we need to recall some basic information concerning them, which is taken up successively in two sections: the first presents sequent and tableau calculi, systems strongly connected with ND; the second deals with systems popular in automated theorem proving like resolution and Davis/Putnam procedure. It happens that the latter systems are based on the application of cut, whereas the former rather tend to eliminate this rule in practice.

It should be noted that the presentation of different types of deductive systems has an elementary character and is limited in two senses. From the variety of systems we have selected only those that are used further as the source of inspiration in building the enriched versions of ND, particularly in the setting of formalization of modal logic. In result many important kinds of deductive systems like connection calculi, goal oriented proof systems or refutation calculi, are not taken into account. Either we do not know how to take advantage of them for the needs of ND (which is not to say that it is not possible!), or they were not used in modal logic, at least not in the way suitable for our purposes. Moreover, we focus only on some theoretical aspects of the discussed systems that are vital for us. In particular, we focus upon the cut rule and its importance for strategies of proof search, and to some related properties of rules like: *subformula property, analyticity, confluency.*

The last section of this Chapter contains a discussion of some complexity problems connected with cut and its elimination or bounded application.

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*,

Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_3,

Again, because of the rudimentary character, much of this Chapter may be skipped in the first reading and consulted when necessary in further lecture.

3.1 Sequent Systems and Tableaux

For an analysis of the range of possible extensions and applications of ND a comparison with other kinds of DS's is required. In order to facilitate the reader a study of more general forms of ND introduced in the next Chapter, we recall here the basic information about systems that are the source of inspiration for these extensions. They fall into two groups.

At first, let us take a look at systems being closely related to ND, i.e.: sequent calculi (SC) and tableau systems (TS). Currently, SC's are rather more popular in theoretical considerations on proof theory, whereas TS's seem to be more often used in practical applications. But, since 80s, they have been even treated as serious rivals to the resolution on the field of automated theorem proving. In fact, these calculi have much longer history than the resolution, and also the earliest working examples of automated theorem provers were based on them.

The first form of SC was invented by Gentzen in the 30s as a result of the research on the alternative to Hilbert systems. His seminal paper [109] contains two such systems (for classical and intuitionistic logics): ND and SC. The proper aim of his investigation was, in fact, the construction of ND-system. He defined SC as a kind of supplementary formalization of only theoretical character, needed to prove the equivalence of ND with axiom systems, and to establish some properties of proofs in the former system. But his famous Hauptsatstheorem, better known as cut elimination theorem, opened the way to define direct proof search procedures (Hintikka [131] and Beth [27] – cf. the next subsection) and the construction of first automated theorem provers (Wang [279] and Prawitz [221]).

Below we briefly describe some variant of SC, and then we show how different forms of tableau systems may be introduced as kinds of refinements of SC. In particular, we will focus on some important (from the proof search perspective) properties of these systems, like analyticity or confluency.

3.1.1 Sequent Calculus

The name of SC refers to the items on which the rules of the system are defined. We have already introduced the concept of a sequent as a pair of sets of formulae, but one should remember that, very often, these are conceived rather as multisets or even sequences of formulae, as in the original Gentzen's solution.[1] To our aim sequents built from sets (or rather from C-set and D-set) are sufficient and this, additionaly makes the comparison with TS's easier. Consequently, the version of SC presented below differs from the original version of Gentzen, in particular, there is no ordinary Gentzen's structural rules of contraction and permutation. The detailed presentation of the evolution of Gentzen system and of several variants of SC's may be found in many places, e.g. in Kleene [162].

The calculus consists of one schema of axiomatic sequent and the set of rules enabling deduction of a new sequent (conclusion-sequent) from one or two sequents (premise-sequents). Here is a list of rules adequate for **CQL**:

$(AX)\quad \varphi \Rightarrow \varphi$

$$(Cut)\quad \frac{\Gamma\Rightarrow \Delta, \varphi \qquad \varphi, \Gamma\Rightarrow \Delta}{\Gamma\Rightarrow \Delta}$$

$$(W\Rightarrow)\quad \frac{\Gamma\Rightarrow \Delta}{\varphi, \Gamma\Rightarrow \Delta} \qquad\qquad (\Rightarrow W)\quad \frac{\Gamma\Rightarrow \Delta}{\Gamma\Rightarrow \Delta, \varphi}$$

$$(\neg\Rightarrow)\quad \frac{\Gamma\Rightarrow \Delta, \varphi}{\neg\varphi, \Gamma\Rightarrow \Delta} \qquad\qquad (\Rightarrow\neg)\quad \frac{\varphi, \Gamma\Rightarrow \Delta}{\Gamma\Rightarrow \Delta, \neg\varphi}$$

$$(\wedge\Rightarrow)\quad \frac{\varphi, \psi, \Gamma\Rightarrow \Delta}{\varphi\wedge\psi, \Gamma\Rightarrow \Delta} \qquad\qquad (\Rightarrow\wedge)\quad \frac{\Gamma\Rightarrow \Delta, \varphi \qquad \Gamma\Rightarrow \Delta, \psi}{\Gamma\Rightarrow \Delta, \varphi\wedge\psi}$$

$$(\vee\Rightarrow)\quad \frac{\varphi, \Gamma\Rightarrow \Delta \qquad \psi, \Gamma\Rightarrow \Delta}{\varphi\vee\psi, \Gamma\Rightarrow \Delta} \qquad\qquad (\Rightarrow\vee)\quad \frac{\Gamma\Rightarrow \Delta, \varphi, \psi}{\Gamma\Rightarrow \Delta, \varphi\vee\psi}$$

$$(\rightarrow\Rightarrow)\quad \frac{\Gamma\Rightarrow \Delta, \varphi \qquad \psi, \Gamma\Rightarrow \Delta}{\varphi\rightarrow\psi, \Gamma\Rightarrow \Delta} \qquad\qquad (\Rightarrow\rightarrow)\quad \frac{\varphi, \Gamma\Rightarrow \Delta, \psi}{\Gamma\Rightarrow \Delta, \varphi\rightarrow\psi}$$

$$(\forall\Rightarrow)\quad \frac{\varphi[x/a], \Gamma\Rightarrow \Delta}{\forall x\varphi, \Gamma\Rightarrow \Delta} \qquad\qquad (\Rightarrow\forall)^1\quad \frac{\Gamma\Rightarrow \Delta, \varphi[x/a]}{\Gamma\Rightarrow \Delta, \forall x\varphi}$$

$$(\exists\Rightarrow)^1\quad \frac{\varphi[x/a], \Gamma\Rightarrow \Delta}{\exists x\varphi, \Gamma\Rightarrow \Delta} \qquad\qquad (\Rightarrow\exists)\quad \frac{\Gamma\Rightarrow \Delta, \varphi[x/a]}{\Gamma\Rightarrow \Delta, \exists x\varphi}$$

Side condition:

1. a is a parameter which does not occur in Γ, Δ and φ.

Usually the rules like (Cut) or both rules of weakening ($(W \Rightarrow)$ and $(\Rightarrow W)$) are called *structural* since they do not exhibit any presence of logical

[1] In fact, the original system of Gentzen is defined on sequents formed from finite lists of formulae, however this is not of importance here.

constants, whereas the remaining ones are called *logical*. Some terminology concerning occurrences of formulae in rules schemata is also convenient. A formula exhibited in the conclusion-sequent is called the *principal formula* of this rule application, formula(e) exhibited in premise-sequent(s) is (are) *side formula(e)*, whereas the elements of Γ and Δ are called *parametric formulae*. In case of (Cut) a formula, exhibited in both premises is called *cut-formula*.

A proof of a sequent S in the system ($\vdash_{SC} S$) is defined in the standard way, as a binary tree of sequents (i.e. each node is decorated with a sequent), where: (i) S is the root (ii) all leaves are axioms (iii) all other nodes are constructed by means of the rules, i.e. premise-sequent(s) is (are) a child(ren) and conclusion-sequent is a parent. Usually, actual proof search is performed in the reverse order; we start with the root-sequent and systematically add above the premise-sequents of suitable rules.

Of course, the notion of a proof may be generalized in order to introduce the relation of deducibility between sequents. It allows the notions of derivable and admissible rules to be easily redefined for SC. Formally:

Definition 3.1 (Deducibility, Rules) Let S_i denote a sequent, then:

1. $S_1, ..., S_k \vdash_{SC} S_{k+1}$, iff there is a proof in SC of S_{k+1}, where leaves are not only axioms but also sequents $S_i, i \leq k$ (if $k = 0$ we have ordinary SC-proof)

2. $S_1, ..., S_k \ / \ S_{k+1}$ is SC-derivable, iff $S_1, ..., S_k \vdash_{SC} S_{k+1}$

3. $S_1, ..., S_k \ / \ S_{k+1}$ is SC-admissible, iff, if $\vdash_{SC} S_1, ..., \vdash_{SC} S_k$, then $\vdash_{SC} S_{k+1}$.

In particular, proofs of admissibility of several forms of cut in the context of SC are usually called cut elimination proofs.

We record below, without a proof, a very useful lemma which helps building SC-theories by addition of sequents or rules of different types, according to the needs.

Lemma 3.1 *If one of the following rules (or a sequent) is added to SC, then the rest is derivable:*

1. $\varphi, \psi \Rightarrow \chi$
2. $\chi, \Gamma \Rightarrow \Delta \;/\; \varphi, \psi, \Gamma \Rightarrow \Delta$
3. $\Gamma \Rightarrow \Delta, \varphi \;/\; \psi, \Gamma \Rightarrow \Delta, \chi$
4. $\Gamma \Rightarrow \Delta, \psi \;/\; \varphi, \Gamma \Rightarrow \Delta, \chi$
5. $\Gamma \Rightarrow \Delta, \varphi$ *and* $\Gamma' \Rightarrow \Delta', \psi \;/\; \Gamma, \Gamma' \Rightarrow \Delta, \Delta', \chi$
6. $\Gamma \Rightarrow \Delta, \varphi$ *and* $\chi, \Gamma' \Rightarrow \Delta' \;/\; \psi, \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$
7. $\Gamma \Rightarrow \Delta, \psi$ *and* $\chi, \Gamma' \Rightarrow \Delta' \;/\; \varphi, \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$
8. $\Gamma \Rightarrow \Delta, \varphi$ *and* $\Gamma' \Rightarrow \Delta', \psi$ *and* $\chi, \Pi \Rightarrow \Sigma \;/\; \Gamma, \Gamma', \Pi \Rightarrow \Delta, \Delta', \Sigma$

The lemma establishes interderivability (mutual derivability) of several forms of rules. For simplicity, we have stated it for the case of three formulae; one can easily establish suitable forms for less or more formulae. For example, in case of two formulae we have only cases 1, 2, 3, and 6 with deleted ψ.

The concepts of satisfiability and validity of a sequent $\Gamma \Rightarrow \Delta$ may be reduced to satisfiability (validity) of the corresponding implication $\wedge\Gamma \rightarrow \vee\Delta$. One can easily check that this form of SC is sound; in order to prove completeness it is enough to prove all the axioms of H-**CQL**.

In the version of SC considered here all the rules except $(Cut), (W \Rightarrow)$ and $(\Rightarrow W)$ satisfy the so called *subformula property.* In SC it means that in premise-sequents there are only subformulae of the conclusion-sequent. The presence of three rules that lack this property is a serious drawback from the point of view of practical applicability, if we treat SC as a tool for proof search. But, fortunately, this inconvenience may be easily excluded. The rules of weakening may be eliminated by introduction of more general form of axioms. Notice that every sequent $\Gamma \Rightarrow \Delta$, where $\Gamma \cap \Delta \neq \varnothing$, is easily deduced from original axioms just by repeated application of weakening. The elimination of cut is not so easy and the original proof is rather involved. It is the main result of Gentzen in [109] that every proof in SC may be transformed into the proof in SC without any use of (Cut).

After these changes (the elimination of the three rules in question, and the replacement of axioms by their generalized forms) we may define the following proof-search procedure which, in case of **CPL**, is in fact a decision procedure. We always start with the sequent we want to prove and construct proof-tree applying rules in the reverse order, from conclusion to premises.

Because the number of choices is always finite for **CPL** (finite number of compound formulae in the conclusion-sequent that may be chosen as principal formulae) so we have a procedure with bounded indeterminacy, which may be easily transformed into fully deterministic algorithm, by adding some order of choices (e.g. from left to right). In **CPL** this procedure, due to the subformula property, always terminates yielding a finite tree which is either a proof or a deduction of the sequent from some atomic sequents. In the latter case we obtain a falsification of tested sequent by consideration of any branch with nonaxiomatic atomic sequent as the leaf. It is sufficient to define a valuation that makes all elements of the antecedent of the leaf true, and all elements of the succedent false.

In case of **CQL** it is not so straightforward. One must additionally redefine rules (∀⇒) and (⇒∃):

$$(\forall\Rightarrow') \ \frac{\forall x\varphi, \varphi[x/a], \Gamma \Rightarrow \Delta}{\forall x\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\exists') \ \frac{\Gamma \Rightarrow \Delta, \exists x\varphi, \varphi[x/a]}{\Gamma \Rightarrow \Delta, \exists x\varphi}$$

An important feature of SC after these changes is invertibility of rules. It holds for every rule that, if a conclusion is provable, then every premise is also provable. It is very important for automated deduction, even for undecidable **CQL**, where we have no guarantee for termination. In particular, the number of choices of a when the application of the above rules is taken into consideration, is not bounded in general.

To sum up; the version of SC without (*Cut*) and weakening, and with axioms and two rules for quantifiers suitably redefined has two important properties called *analyticity* and *confluency*. Let us explain them briefly.

Analyticity

The term *analyticity* is applied in at least two senses in the field of investigation on proof methods.[2] In one sense analytic systems are opposed to synthetic ones. In analytic systems rules allow us only to decompose formulae into their components, whereas in synthetic systems we have just an opposite operation. In this understanding SC would be classified as a synthetic system, if we follow the statement of definition of rules. But if we follow the way we actually proceed when building proof-trees ("upside-down" construction as described above), we should rather treat SC as an analytic system. In this perspective ND-systems should be classified as of mixed character, since their rules allow both decomposing and building up

[2] Poggiolesi [214] contains an interesting historical exposition of this concept.

formulae from components.

We do not use this term in such a sense but rather say that a system is analytic if it satisfies subformula property or some reasonable generalization of it. But what does it mean for the system to satisfy this property; after all, we have defined it only for the rules. In an obvious sense we say that a system satisfies this property if all its rules have it. In this sense our starting version of SC is not analytic but our final (cut-free) version is. Very often, especially in earlier works concerning SC's, analyticity in this sense is identified with cut elimination. It is not true in general; there are lot of systems, where cut is admissible but some other rules do not satisfy subformula property and they are not eliminable.[3]

This sense of analyticity, called further *strict analyticity*, may be liberalised in order to cover also some systems that, on the level of calculus, are not (strictly) analytic. Even if not all rules satisfy subformula property, we may require, on the level of realization, that it holds for all admissible proofs. It means that every application of a rule in a proof, even if it is building up (synthetic) rule, is restricted to use only elements of some well defined finite set; typically subformulae of premises and their negations. Analyticity in this sense implies the elimination of full indeterminacy in looking for the next steps of proof construction; possible choices are strongly limited to predetermined finite set. Obviously, every strictly analytical system is analytical, but the opposite does not hold.

Confluency

The notion of confluency of the system makes sense only with respect to such types of formalizations that allow for the definition of proof search procedures. In case of SC it may be defined as follows:

Definition 3.2 SC system is confluent iff, if a sequent S is provable, then any proof-tree with this sequent as the root may be extended in such a way that we obtain a proof of S.

It is in fact a consequence of invertibility of rules. In practice it means that no matter what choices we have made during proof-search, we finally obtain a proof, if the input is provable. On the other hand, if some branch ends with atomic but not axiomatic sequent, we are done negatively, we

[3] Cf. *Display Calculus* [280], or SC of Mints [190] for **S5**.

know for sure that there is no proof. That's why, confluent systems are very convenient from the point of view of automated theorem proving. We are not forced at some stage of proof-search for backtracking to earlier stages, if we made "wrong" choices. Consequently, confluent systems are less expensive for the program memory. From the point of view of the result, in confluent systems there are no wrong choices (although our choices may have strong influence on the length of the obtained proof or the time needed for performance).

3.1.2 Tableau Systems

If we think mainly on practical applications of a deductive system, then SC seems to be rather complicated. We may treat as an obvious simplification a redefinition of rules (and proof) implied by the actual practice of proof search in SC: from conclusion-sequent to premise(s)-sequent(s). In this case premises and conclusions exchange their roles, and a proof is defined as an inverted tree; some authors prefer such formulation. One may also eliminate sequents and introduce sets of formulae as basic items. In this case we have two options: If in every sequent we move all formulae from antecedent to succedent (by $\Rightarrow \neg$), then we obtain right-sided sequents of the form $\Rightarrow \neg\Gamma, \Delta$. Converse operation gives left-sided sequents of the form $\Gamma, \neg\Delta \Rightarrow$. In both cases $\Rightarrow$ is superfluous; we operate on D-sets in the first case, and on C-sets in the second. The first solution was introduced by Schütte [244] as a simplified form of SC, and independently by Rasiowa and Sikorski [228] as a form of TS. The second solution was developed by Hintikka [131]. From now on we will say, for simplicity, about Schütte (type) system (or format) and Hintikka (type) system. With the help of generalized notation introduced in Section 1.1., we may formulate the rules of both systems in the following way.

Schütte system contains two rules:

$$(\alpha S)\ \ \frac{\Gamma, \alpha}{\Gamma, \alpha_1 \ \mid\ \Gamma, \alpha_2} \qquad\qquad (\beta S)\ \ \frac{\Gamma, \beta}{\Gamma, \beta_1, \beta_2}$$

Hintikka system contains two rules:

$$(\alpha H)\ \ \frac{\Gamma, \alpha}{\Gamma, \alpha_1, \alpha_2} \qquad\qquad (\beta H)\ \ \frac{\Gamma, \beta}{\Gamma, \beta_1 \ \mid\ \Gamma, \beta_2}$$

Both systems have a rule for double negation elimination:

$(NN)\ \Gamma, \neg\neg\varphi \ /\ \Gamma, \varphi$

In both cases a proof for a set Γ is an inverted binary tree with this set as the root, constructed by the application of rules, where every leaf is a complementary set. Each system is dual to the other; Schütte system is branching on α-formulae, whereas Hintikka system is branching on β-formulae, which is the result of different interpretation of sets. Moreover, Schütte system is direct in the sense that we use only direct proofs. If we search for a proof of φ we start with a set $\{\varphi\}$, and complementary set in each leaf is tautological (a disjunction containing an instance of the excluded middle law). If we are restricted to the propositional part, and we add in realization a condition that the procedure is applied until we get an atomic set (only literals) in every branch, then Schütte system may be seen as a particular form of realization of the procedure for obtaining a conjunctive normal form (CNF) of φ known from the completeness proof of Post. System of Hintikka is an example of refutation calculus, using only indirect proofs. If we search for a proof of φ, we start a tree with $\{\neg\varphi\}$ as the root, complementary set in each leaf is an instance of elementary contradiction. On propositional level every tree with atomic leaves gives a recipe for disjunctive normal form (DNF) of $\neg\varphi$.

Further evolution of TS's was usually connected with the modification of Hintikka approach and led to additional simplifications. From the point of view of practical pen and paper application, especially in teaching logic, tableaux using sets are far from being ideal because it is still connected with tedious and boring practice of rewriting parametric formulae. The solution to this problem is provided by the version of tableaux introduced by Beth, commonly called Beth diagrams [27] and its simplifications due to Lis [178] and Smullyan [261].

The version of Beth is free from rewriting parameters but obtained diagrams are often difficult to read in case of many branchings. A tableau is divided on two parts with formulae assumed as true on the left and formulae assumed as false on the right. Thus, when testing provability of a sequent we write all elements of the antecedent in the left part and all elements of the succedent in the right part. The application of rules leads to writing further formulae to suitable parts of a tableau and, in case of β-formulae, to the introduction of additional divisions of a tableau. Proof is obtained if in every part of a tableau we get the same formula on both sides, which exhibits contradiction. We do not describe this form of representation of TS in detail but rather focus on the solution of Lis/Smullyan. In this version, we build binary (inverted) trees from single formulae (not sets) as nodes. A proof-tree for a sequent $\varphi_1, ..., \varphi_k \Rightarrow \psi_1, ..., \psi_n$ starts with a branch having

$k + n$ nodes – all elements of the antecedent and negated elements of the succedent.[4] We extend a tree applying the following rules:

$$(\alpha)\ \frac{\alpha}{\alpha_i} \quad \text{where } i \in \{1, 2\} \qquad (\beta)\ \frac{\beta}{\beta_1 \mid \beta_2} \qquad (NN)^5\ \frac{\neg\neg\varphi}{\varphi}$$

Generally, the rules of inference applied in tableau systems for extending (and branching) a proof-tree will be called *rules of expansion.* A tree containing a pair of complementary formulae in every branch is a proof of a sequent, whereas a tree with at least one open (no complementary formulae) and finished branch (no application of expansion rules yield new formulae), provides a falsifying interpretation. It is one of the most popular forms of TS for classical logic (and many nonclassical as well), so in the remaining part of the book it will be used as a natural point of reference when talking about tableaux.

Systems of sequents or tableaux described above will be called standard. All of them are analytic and confluent, although in case of tableau systems (except Beth variant) analyticity should be understood as the restriction of choices in proof search not only to subformulae of an input but also to their negations. The qualification "standard" is mainly for contrasting them with some forms of SC's and TS's that were introduced later. Many of these nonstandard systems were developed to obtain the formalizations of nonclassical logics. In these calculi we meet an essential enrichment of the standard formal apparatus obtained in several ways; we will introduce some such "nonstandard" calculi for modal logics in next chapters. Also, an intensive development of automated theorem proving based on tableaux introduced several forms of TS seriously differing from the systems described above. Some of them are in fact good examples of hybrid systems combining TS with resolution or connection calculi. We do not present these refinements since automated deduction is not our aim; one may find excellent surveys in Hähnle [120] or in papers (e.g. Letz' contribution) from [121].

[4] In fact, both authors prefer metalinguistic devices: Lis uses + for assertion and – instead of negation, Smullyan uses prefixes T and F.

[5] In what follows we will sometimes use the same name for different variants of essentially the same rule in related systems if it does not cause misunderstanding.

3.2 Resolution and Davis/Putnam Procedure

Both systems discussed in this section are not very close to ND paradigm, but they are important for automated theorem proving. It is a fact that in this group of systems, the dominant role is played by calculi that use some forms of cut, in contrast to SC or TS. In particular, cut is present explicitly as a (propositional) resolution rule in resolution systems. In other systems from this group, like Davis/Putnam procedure (DP) or connection systems, its presence is less visible but also essential. Although resolution and DP are commonly seen as having nothing in common with ND, we will show in the sequel that this may be easily changed. The important point is that ND is essentially based on the (implicit) application of cut (e.g. by transitivity of inferences but not only), similarly as systems discussed in this section. The main difference lies in the fact that both resolution and DP is working on clauses. But why not to think that ND can be defined on clauses, as well?

3.2.1 Resolution

Since Robinson introduced the first form of resolution in 1965, it has become almost an industrial standard in the field of automated theorem proving. An excellent and detailed exposition of this method may be found elsewhere,[6] so we will be very brief. The popularity of resolution is a consequence of its striking simplicity. In the simplest form for **CPL** it is just the application of a special form of cut, called the resolution rule, to the set of atomic clauses, until we get an empty clause ($\bot$) or irreducible set of literals. It may be described in three steps:

1. Negate tested formula (or make a conjunction of premises and a negation of a conclusion, if testing an argument).

2. Transform an input to conjunctive normal form (or C-set of clauses).

3. Apply successively to the set of clauses a rule of resolution (*Res*) of the shape:

$$\Gamma, \varphi \ ; \ \Delta, -\varphi \ / \ \Gamma, \Delta$$

[6]E.g. a classic Chang and Lee [68]; particularly usefull presentations for our purposes are Gallier [101] and Fitting [95].

If after n applications of (Res) to the set of clauses we obtain an empty clause ($\perp$), then tested formula is tautological (or argument valid); otherwise we can define a falsifying valuation.

Enrichment of propositional resolution with skolemization and unification gave the most popular method of automated theorem proving for first-order logic, despite the undecidability of this logic. In 70s it was an unquestionable leader in this field and almost all provers were based on some form of this system.[7]

But despite these advances, the application of resolution to nonclassical logics, and particularly to modal logics has shown serious limitations.[8] The basic problem is connected with the lack of simple normal forms for modal languages. In modal logic usually normal forms are complex and the resolution must be performed inside the scope of modal operators and even in propositional logic it requires some form of skolemization (cf. e.g. the system of Enjalbert and Fariñas del Cerro in [84]).

Problems with modal normal forms were responsible for great popularity of indirect applications of resolution methods to modal logics. They are based on some form of translation of modal language into first-order language or some other richer language (e.g. relation calculus, see [201]). Schmidt and Hustadt [243] provide the detailed survey of different translation techniques applied for modal logics. Indirect approach offers many advantages since we can apply ready to use provers and plenty of effective strategies tested during the last 40 years of work on the optimization of classical resolution.

On the other hand, the indirect resolution has some drawbacks connected with the fact that decidable modal logics are translated into undecidable first-order (or second-order) logic. This operation requires the introduction of additional specific strategies for termination of the fragment of this rich language that corresponds to the suitable modal language and usually leads to implementations that show worse performance than tableau based provers. But it should be noted that some recent investigations on the indirect methods for some rich multimodal logics [196, 242] open new perspectives. In particular, by smart translation we can not only profit from first-order resolution strategies but even develop tableau calculi for these

[7]Clearly, this success was not only the result of simplicity of the original approach but also of the development of several optimization techniques and refinements of original system. One may mention here the introduction of hiperresolution, linear resolution, ordering of literals, selection function e.t.c.

[8]One can find a useful survey in [84, 10].

logics of the same complexity.

Another sort of the resolution calculi for modal logics belongs to the group of direct resolution methods (without translation), often called non-clausal because they do not need any conversion of the input into a normal form. We prefer to say that they use generalized clauses. Resolution systems operating on generalized clauses are in general more appropriate for nonclassical logics since, as we have remarked, such normal forms are usually quite complicated. In fact, such non-clausal forms of resolution seem to be more efficient even in classical logics, since they reduce complexities connected with the first phase of translation.[9] For example, in many-valued logics systems of this sort were devised by Stachniak [262], in modal logic, such systems were offered by e.g. Fitting [94], Abadi and Manna [1]. But there are only a few systems of this sort, and they are developed for only a few particular logics, so the general direct resolution approach for ordinary modal logic is still to come.

There is but one more problem connected with the standard resolution. Despite its simplicity it is rather not a friendly system for humans: neither the actual search for a proof, nor even reading the results provided by a machine, is easy. This is not a drawback if we are just interested in quick response: is it provable or not? But if we are interested not only in the result obtained for a given input but we want to see the actual derivation we face a problem. It is of particular importance, if we are interested in the construction of a proof itself, or in finding the simplest and most direct deduction. It is also important for different kinds of checkers and other interactive programs applied in teaching logic. In fact, recently some efforts have been undertaken (see e.g. [137, 138, 196]) to make the resolution proofs more readable.

Notwithstanding all the above-mentioned vices, the dominating position of resolution in automated theorem proving remains stable. Fourty years of investigations has brought a huge number of optimizing strategies suitable for resolution. In fact, many of them were also successfully applied in tableau based theorem provers. The possibility of transfer of these results to ND systems seems really vital. We undertake this problem in the next Chapter.

[9] In standard procedures of transformation it may lead to the exponential blow up, but at present much better techniques are applied, cf. e.g. [196, 242] for methods of structure-preserving reduction working in polynomial time.

3.2.2 Davis/Putnam System

One of the first procedures for the automated deduction provided for classical logic is due to Davis and Putnam [77] (shortly called DP). It is still considered as one of the most efficient method for **CPL**; the full version for **CQL** was not as efficient as the resolution because of the lack of unification. As far as I know, DP was not applied in the field of nonclassical logics, which however, does not mean that it is not good for that. We recall briefly this method following the presentation of [2, 95], where a more detailed exposition may be found.

The application of DP also assumes that in the first stage we transform a negation of a formula we want to prove into the set of atomic clauses. The rest of the process has more complex character. We apply to the set of clauses X some pre-processing rules that reduce it to the set of clauses X' that is satisfiable iff X is. These are the following:

1. We delete every tautological clause (i.e. including a pair of complementary literals (*a rule of tautology*).
2. We delete all clauses containing literal φ such that its complement $(-\varphi)$ is not present in any clause (*pure literal rule*).
3. We delete every clause Γ such that there is a clause $\Delta \subseteq \Gamma$ (*subsumption rule*).
4. If in clauses $\Gamma_1, ..., \Gamma_k$ we have a literal φ, in clauses $\Delta_1, ..., \Delta_n$ we have its complement, and there is unit clause $\{\varphi\}$, then we delete all clauses $\Gamma_1, ..., \Gamma_k$ (also $\{\varphi\}$), and all clauses $\Delta_1, ..., \Delta_n$ change into clauses $\Delta'_1, ..., \Delta'_n$, without the complement of this literal i.e. $\Delta'_i = \Delta_i - \{-\varphi\}$ (*unit literal rule*).

All these operations were called rules in accordance with commonly applied terminology although they are rather some strategies or complex techniques of reducing the set of clauses.[10] Only the rule of unit literal has a slightly different character. It is partly a special case of subsumption rule (we delete all Γ_i being supersets of $\{\varphi\}$), but one can easily see that it comprises n-time application of the particular case of resolution:

$$(Res1)\ \varphi\ ;\ \Delta'_i, -\varphi\ /\ \Delta'_i$$

[10] In fact, all DP rules are also used as simplification strategies in resolution or tableau-based provers.

If the set of clauses X is unsatisfiable, then in many cases we may obtain an empty clause as a result of the above "rules", which constitutes a proof, exactly as in resolution systems. But such a system is incomplete for **CPL**, mainly because of the restricted form of resolution rule incorporated in unit literal rule. When the systematic application of 1.–4. results in the similar situation of the set of clauses X' to that one of 4., and moreover there is no unit clause $\{\varphi\}$ that would admit the application of the rule, we may apply instead:

5. *splitting rule*, we distinguish two sets: $X' \cup \big\{\{\varphi\}\big\}$ and $X' \cup \big\{\{-\varphi\}\big\}$.

It is evident that to each of the new sets we may apply a rule of unit clause or other of the aforementioned rules. If it is necessary then we may apply this rule again on the later stage to some of the distinguished sets. Clearly, if we have applied a splitting rule at least once to obtain a proof, we must get an empty clause in each set of clauses.

Our description of DP is informal and neutral with respect to particular forms of realization and implementation. In Chapter 4 we will show that a variant of this procedure may be easily simulated in some ND system thus giving particularly readable derivations.

3.3 Cut and Complexity of Proof

In many respects tableau systems seem to surpass resolution systems. As a matter of fact, their calculi contain many rules in contrast to the resolution, which, from the point of view of implementation, may be seen as a drawback. But these rules have a uniform character, moreover, they are intuitive and easy to use. In consequence, proof search in tableau systems is easy not only for computers but also for humans (that is why tableaux are so popular in teaching logic). Tableau systems do not require also preprocessing step of transformation into normal form.[11] This property perhaps decides about the great popularity of TS's in the formalization of nonclassical logics; good evidence for this claim provide e.g. [121] or [222]. On the other hand, the number of nonclassical logics having direct (i.e. no translation) formalization in terms of resolution systems is – as we have already remarked – pretty small (cf. [84] in case of modal logics).

In fact, the first automated theorem provers were based on sequent and

[11] In fact, modern tableau-based provers often work on clauses which simplifies adaptation of strategies developed in the setting of resolution.

tableau systems; one can mention here, among others, works of Prawitz or Wang (cf. [221, 279]). But in 70s the resolution seemed to win in this field, while SC's and TS's were almost forgotten by the specialists in automation. A look at some popular books on automated theorem proving from these years, like very influential [68, 179] suffices to see that SC's and tableaux are completely neglected. First attempts at comparison of resolution with SC or tableaux are rather late, e.g. Gallier [101] or Avron [15]. In 80s tableau systems begun to be treated seriously and seen as a rival to resolution, but soon development of complexity theory has raised serious doubts. It appeared that from the point of view of the (worst cases of the) lenght of proofs, tableau methods are not serious rivals not only to resolution but even to ordinary truth table method of tautology checking which, in common judgement, is totally inefficient. Paradoxically, the source of the problem is a phenomenon which for years was seen as a basis of their success, namely cut elimination.

In standard SC of Gentzen or tableau system being essentially its inversion,[12] we have a clear situation: all the rules except cut have subformula property, so if we can eliminate cut we obtain an analytic system. This observation has led to popular belief that cut elimination equals analyticity. But this belief is wrong since cut elimination in itself does not guarantee subformula property in systems having richer formal apparatus. It is quite common in case of standard SC-formalizations for nonclassical logics with the set of extra rules, or in case of nonstandard systems using richer language, e.g. display calculi. In such generalized forms of SC there are additional rules without subformula property, or additional complications connected with more complex structure of the calculus. For example, in display calculus cut is eliminable and even logical rules satisfy subformula property, but to obtain a real analyticity of the system we need also something like *substructure property.* So cut elimination is not a sufficient condition for analyticity of the system. As we shall see, it is not even a necessary condition, and even worse – it may be disastrous from the point of view of efficiency of the system.

Since 60s a lot of SC's and TS's for which cut elimination was not provable was devised. For many years it was felt as a great disadvantage of these formalizations. Some positive sides of cut were noticed only recently. Earlier it was only observed that for many sequent or tableau formalizations

[12] Note that this simplification applies to the formalization of classical logics; it is not necessarily true on the ground of nonclassical logics, where many proposed SC's and TS's are not in such direct relationship. In case of modal logics cf. Chapter 7 for details.

of some nonclassical logics we can extract some special applications of cut that are necessary for completeness. (cf. Fitting [93] or Takano [269] for some modal logics). But such compromises were treated as rather bad solution.

The research on the complexity of proof procedures has shown however, that even in systems where cut is completely eliminable, e.g. in classical logic, some controlled applications of this rule may lead to the construction of essentially shorter proofs. The source of the problem lies in the fact that in standard cut-free TS we have often too much branching with the same sequences of steps repeated. This may even lead to the exponential growth of the length of proof tree.[13]

After some reflection it should not be as strange as it seems; cut is a rule-realization of one of the most important properties of consequence relation – its transitivity. We have already mentioned that in several systems this property is expressed by means of several rules, but cut is one of the most important. Abandoning this property seems counterintuitive, and involves important costs on the side of simplicity and the length of obtained proofs. In fact, it was the well known truth in the field of automated theorem proving; resolution on propositional level is just a special form of cut. But as we have remarked, for many years there were no traces of communication between research on resolution (and automation) and research on SC's and TS's (and philosophical logic).

What is really vital for us is the fact that the lack of cut elimination does not exclude analyticity of the system. Here we mean not the strict analyticity but the weaker notion introduced in Section 3.1. which does not require subformula property of all the rules. It is sufficient if this property holds for proofs, and this is possible if we can restrict the application of rules to subformulae of proved sequent.

Definition 3.3 In a proof of $\Gamma \Rightarrow \Delta$ an application of a rule is analytic iff, in premise-sequents there occur only formulae from the set of $\mathrm{Subfor}(\Gamma \cup \Delta)$.

For tableau system we have the following definition:

Definition 3.4 In a proof for Γ an application of a rule is analytic iff, every conclusion belongs to the set of subformulae of premises of this proof (i.e. Γ) closed under single negations.

[13] It was Boolos who for the first time paid an attention to this problem, see [51] and a detailed discussion in D'Agostino [2] and Fitting [95].

The notion of an analytic application of a rule may be defined for any deductive system not only for SC or TS. Any system restricted on the level of realization to such applications of rules will be called analytic. Clearly, every strictly analytic system, i.e. containing only rules with subformula property in its calculus, is analytic.

In particular, we can restrict admissible applications of cut in a system to analytic only. For instance, as we have mentioned, the resolution rule is a special form of cut, but in a system for classical logic its applications are analytic. In fact, it is also a consequence of cut-elimination theorem but in a generalized form (c.f. Avron [15]). Strong cut elimination theorem says that every proof of $\Gamma \Rightarrow \Delta$ from the set of sequents S may be transformed into a proof where all applications of cut are made on formulae which are elements of S. Thus, in particular, for $\Gamma = \Delta = \varnothing$ and S consisting of only atomic sequents, strong elimination theorem guaranties that $\Rightarrow$ is deduced from S only by the application of cuts. But it is just a notational variant of the claim that if $S \vdash \bot$, then it may be proved by the applications of resolution only.

The investigations on sequent and tableau systems for the nonclassical logics have also shown that although cut is not always eliminable we can often restrict its use to the analytic instances and still obtain a complete formalization. Sometimes the conditions for the admissible applications of cut must be more liberal, e.g. the closure of the set of respective subformulae not only under single negations but also under modal operators. The examples of such generalized analytic systems for some modal logics were proposed by Takano [269] and Fitting [93] (cf. also Goré [117]). In the setting of ND systems, proofs of normalization theorems (cf. the next Chapter) show that such formalizations may be also transformed into analytic versions without loss of completeness.

So the absolute cut elimination is for automated theorem proving not only dispensable but may be even troublesome. The recognition of advantages of cut caused that in many TS's, analytically restricted cut is added as an additional rule even if the system is complete without it. Cut in TS's defined on sets of formulae (Hintikka or Schütte systems) has the following form:

$$(R\text{-}Cut) \qquad \frac{\Gamma}{\Gamma, \varphi \quad | \quad \Gamma, -\varphi}$$

Letter "R" means that this is a regressive form of cut in contrast to resolution rule or (Cut) in SC which are progressive.

In Smullyan's system cut has a form of a branching non-premise rule, so it is convenient to display it as follows:

(PB)

$$\begin{array}{ccc} & \mathcal{B} & \\ \swarrow & & \searrow \\ -\varphi & & \varphi \end{array}$$

Where $\mathcal{B}$ is a branch extended by the application of a rule. The name (PB) (the principle of bivalence) is applied on the ground of the system KE introduced by Mondadori and D'Agostino [2]. In KE, cut is a primitive (not eliminable) and unique branching rule in the system. The name (PB) refers to one more (semantical) interpretation of cut, as a formalization of the property of two-valuedness or the law of excluded middle. The remaining branching rules from the tableaux were substituted in KE by their linear (nonbranching) counterparts. Proofs in KE are then binary trees constructed as in Smullyan's system but instead of (β) we apply ND rule:

(βE) β, $-\beta_i$ / β_j, where $i \neq j \in \{1,2\}$

KE-proofs are elegant and simpler than proofs in tableau systems. Proof of completeness of KE with respect to classical logic requires only the analytic applications of cut, moreover of a specified character. [2] contains the description of a proof search procedure that uses (PB) only if there is β-formula on the branch such that neither β_i (for $i = 1$ or 2) nor the other premise needed for the application of (βE) is present on the branch. In this case we apply (PB) for β_i and its complement. It is easy to show that in KE one can p-simulate every proof-tree in Smullyan's TS. If tableau proof-tree has n nodes and k forks (i.e. k uses of β-rules), then its KE-simulation has at most $n + k$ nodes, because even if for every application of (β) we must use (PB) then in every fork we have only one additional formula in one branch, namely the complement of the chosen β_i.

On the other hand, TS without some form of cut or other additional techniques,[14] is not able to p-simulation of KE proof-trees. D'Agostino exhibits an example for which TS-simulation requires exponential increase of the length of KE proof-tree. Since KE may p-simulate resolution, DP, ND (in tree format), and all these systems may p-simulate KE, then from this point of view they are of the same complexity measure (sometimes

[14]For example, the use of lemmata or merging, cf. [2].

called *relative proof length complexity*), whereas standard TS belongs to the "worse" class of systems. It is also evident that exactly the possibility of cut applications is the property allowing shorter proofs.

This is not however, the reason to maintain that TS is generally less efficient than KE, or other systems with cut. There are good reasons to think that relative proof length complexity is not a very good measure for evaluating practical efficiency of a system. One should remember that systems producing "short" proofs may be less efficient because proof search space may be larger, as we remarked in Section 1.2.4.[15] Also, in case of automatization, an implementation may be more involved. In fact, this is the case of KE, when compared with ordinary TS – we will say more about it in the next Chapter. In case of modal logics, the applications of cut may cause even more troubles, as we will see in Chapter 10. Anyway, in ND we cannot get rid of cut and that is why we rather focus on the advantages of its use.

The place is appropriate to recall DP as an example of the system, where cut is applied in both ways – regressively (splitting rule) and progressively (unit literal rule). For instance, in resolution systems cut has only progressive form, whereas in KE it has only regressive form (as (PB)). Hence DP is a system with the maximal exploitation of cut and perhaps this is the source of its efficiency.

Systems with primitive and not eliminable cut will be called cut-systems in the sequel. The intersection of the class of analytic systems and cut-systems is – as we have noticed – not empty. Any axiomatic Hilbert-style system or standard ND-system is a sample of non analytic cut-system, where (MP) is a form of cut. Standard TS system for **CPL** is an example of analytic system without cut. KE and resolution systems belong to the intersection of these two classes; they are analytic but the elimination of cut makes them incomplete.

It is not accidental that we have paid so much attention to these questions. In what follows we will be interested in cut-systems, that are analytic or may be tailored to this form on the level of realization without loosing completeness. In particular, we will show that ND systems for **CPL** and for many modal logics may be constructed in analytic version.

[15]For more about it cf. e.g. Hähnle [120] and especially a study of Plaisted and Zhu [211], where formal treatment of proof search space is provided for many types of calculi.

Chapter 4

Extended Natural Deduction

In this Chapter there is a continual emphasis on the application of ND as a tool of proof search and possibly of automation. In particular, we take up the question of how to make ND a universal system. In order to find satisfactory solutions we compare ND with other types of DS's. Although ND systems are rather rarely considered in the context of automated deduction they presumably accord with each other and ND systems may be turned into useful automatic proof search procedures. Moreover, even if there are some problems with the construction of efficient ND-based provers, it seems that for the widely understood computer-aided forms of teaching logic, ND should be acknowledged. A good evidence for this claim is provided by the increasing number of proof assistants, tutors, checkers and other interactive programs of this sort based on some forms of ND. Section 4.1. is devoted to the general discussion of these questions, whereas the rest of the Chapter takes up successively the presentation of some concrete, universal and analytic versions of ND for classical and free logic.

In Section 4.2. we define the most restrictive system called AND1 (analytic ND1). It is modelled after D'Agostino and Mondadori [2] KE system. Although the adequacy of AND1 with respect to **CPL** is shown indirectly by p-simulation of KE, we also present a proof search procedure that yields a direct completeness proof and decision procedure. In contrast to standard ND, this variant is universal, i.e. we may not only produce proofs but also provide countermodels through the saturation process. On the other hand, purists may contend that AND1 is not a genuine ND system, since it excludes introduction rules and proceeds only by indirect proofs. In Section 4.3. we present more flexible version of analytic ND called AND2. This time we do not provide a mechanical proof search procedure but rather discuss

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*,
Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_4,

some examples showing that it may produce shorter proofs than AND1.

The last section contains the presentation of a generalized form of ND system called RND (resolution based ND) operating on clauses. It is more powerful than both AND systems since it allows us not only to p-simulate KE and tableaux (like AND1) but also to simulate clausal systems like resolution or Davis/Putnam procedure. RND offers many advantages over other ND systems. First, it has strikingly simple structure on the level of calculus with only one general proof construction rule. Moreover, this simple generalization leads to the possibility of building much shorter proofs than standard ND or its analytical versions. Finally, it is possible to apply in RND the efficient strategies of proof search tested on the ground of resolution.

4.1 Analytic and Universal Versions of ND

In this Chapter, as in this book in general, we do not pursue the subject of automated theorem proving but rather focus on the practical pen-and-paper application of ND. Nevertheless, some remarks on the existing programs utilizing ND systems are in order.

ND systems are commonly believed to be relatively seldom considered in the context of automatization of proofs. In the first instance, ND is treated as the most flexible tool of deduction. In order to make derivations short, and easy to construct, a lot of freedom in proof construction is allowed which is realized in different ways, as we noted in Chapter 2. But what is good (in the sense of rich resources) for deduction is not as good when automation is in mind, because such features like flexibility of proof construction or naturalness of rules are not always in accordance with the requirements of automated deduction. In fact, sometimes they are even felt as being in conflict. Mainly for this reason up till now, most provers have been based on resolution and tableau systems. ND was commonly seen as a proof technique not easily adaptable for automatization.

That it is only a prejudice, and ND may be successfully applied in the field of automated deduction, should be obvious by now. Kalish and Montague [159] presented automated procedures for ND already in 1964, Smullyan [259] proposed another one in 1965. The former was based on the Kalmar completeness proof and not very satisfactory, but since then a great progress was made.[1] At present there are a lot of ND-systems of

[1] One may consult in this matter e.g. papers collected in the monographic volume of

sufficiently "mechanical" character, and many programs based on them, including provers, proof assistants and general environments for implementing several logics often called logical frameworks. Generally, by provers we mean programs searching for (dis)proofs with no human help, whereas proof assistants or checkers are interactive programs where humans build derivations with some occasional help of a computer. It seems that in the business of pure provers ND-based programs are rather rare, but their impact on the remaining fields is still growing. In fact, a clear division between different types of programs is difficult to obtain. For example proof checkers are in principle restricted to checking validity of lines introduced by the user constructing a proof, but there are many programs called proof checkers that generally give at least some advice on proof search, and sometimes even fully automated search is provided. An early system EXCHECK/VALID due to Suppes is perhaps the first of such proof assistants; one may mention also Mizar, BERTIE, and many others. Let us, for the sake of example, and with no pretension to the exhaustive presentation, recall some of the proposals:

- THINKER of Pelletier is fully automated prover being an implementation of KM system. [204] contains detailed presentation of the system and proof search procedure implemented in the prover.
- OSCAR of Pollock [216] is an automated theorem prover which works not only in classical first-order logic but also implements some forms of defeasible reasoning.
- Symlog of Portoraro. Although the program is an interactive proof assistant it is also equipped with a theorem prover that helps students to construct proofs. [219] provides a discussion of the set of proof search strategies involved in the program.
- MacLogic of Dyckhoff. It is both proof assistant and proof checker implementing not only classical logic but also modal logics.
- PROVER of Bolotov, Bocharov, Gorchakov and Shangin [46] for first-order logic, recently extended to some propositional temporal logics.
- CMU Proof Tutor of Sieg [254] is an example of automated proof engine showing how to build normal proofs (see below) directly. This is in contrast to standard results concerning normal proofs that are obtained via the transformation of ready non-normal proofs.

Studia Logica (1998) devoted to automated ND.

The number of programs is one thing; the other is their efficiency. Surprisingly enough it appeared that some of the proposed provers may solve many hard problems faster than the resolution based engines.[2] It seems that we have sufficiently strong evidence for claiming that ND is not worse for automatization than tableaux or resolution and, in case we are interested not only in obtaining fast result but also in its readability, it may work even better. In fact, all these methods are closely related, which is sometimes obscured by several details of minor importance, connected rather with the form of presentation than with the content of a system. One of the aims of this Chapter is to show that ND may work equally well as standard resolution or tableau systems because it may step-wise simulate them.

4.1.1 Analyticity

The standard version of ND presented in Chapter 2 is highly redundant, which makes reasonable the expectation that we may apply in it different strategies of proof search. On the other hand, it has at least two serious drawbacks: it is not universal and not analytic.

In standard ND, if we cannot find a proof of a formula we are not able to establish its status: is it not a thesis or are we not clever enough to find a proof? Also the construction of many rules gives no clues for proof search in ND. In fact, these requirements (i.e. universality and analyticity) are independent but it is desirable to have a system having both. We show that ND may be reformulated in such a way that it is universal – we can do both: prove theses and extract falsifying models for non-theses on the basis of unsuccessful derivations. What is more, on this basis it is possible to define mechanical procedures of proof search not only for classical logic but also for modal logics thus obtaining working decision procedures in decidable cases. In fact – as we have mentioned above – already [159] contains some algorithm of proof search for their ND system. However, it is a procedure based on the Kalmar's version of completeness proof and, from the point of view of efficiency, it is not very interesting. Below we present the solutions conceptually close to the proof search procedures defined for KE.

The fact that not all rules of ND satisfy subformula property does not exclude the possibility that the system is analytic in the sense explained in Chapter 3. There is a rich tradition of investigation on analytic versions of ND initiated by Prawitz [220] (independent result is in Raggio [227])

[2] Cf. e.g. remarks on performance of OSCAR in [216] and more substantial comparison of OSCAR and OTTER performance in [217].

and developed by many logicians under the heading of normalization of ND proofs, or the existence of ND proofs in normal form.[3]

To give a reader some idea of normal proof in possibly simple way, let us assume for the moment that we deal with some T- and F-system for propositional logic with only $\rightarrow$ and $\wedge$, then:

Definition 4.1 A proof $\mathcal{D}$ is in normal form iff no formula in it is both the consequence of an introduction rule and a major premise of an elimination rule for the same constant.

The combinations of applications of the rules excluded in normal proofs, lead to the introduction of the so-called maximum formulae in the proof which is a form of redundancy. Hence, we may simply say that a proof is normal if it has no maximum formulae. Proofs in normal form are analytic in the sense that they satisfy subformula property of the following kind:

Let $\mathcal{D}$ be a normal proof of $\Gamma \vdash \varphi$, then every formula $\psi \in \mathcal{D}$ is a subformula or negation of a subformula of $\Gamma \cup \{\varphi\}$

Notice that although all normal proofs are analytic, the converse does not hold. Analyticity of normal proofs follows from the fact that normal proofs have no detours and lead directly from premises to conclusion by, first, breaking compound formulae into pieces, and then, building up an expected input. In this simplified account we have omitted subtleties connected with the rules for disjunction[4] and negation (or $\bot$ which is often treated as primitive in normalizable ND systems). Several normalization proofs for various versions of ND differ in this respect; in particular, different sets of rules for $\bot$ were provided to obtain a result.

An interesting solution is offered by Negri and von Plato [193], who present a version of normalizable ND satisfying a stronger property to the effect that all major premises of elimination rules are assumptions in normal proofs. This is obtained at the expense of introducing generalized elimination rules since the standard elimination ones (except Gentzen's $[\vee E]$ and $[\exists E]$) also complicate matters. These new elimination rules are all proof construction rules of the form:

[3]Von Plato discovered that Gentzen himself has proved a normalization theorem for intuitionistic ND; he published Gentzen's version in [213].

[4]For example, the original proof of Prawitz is for the system with rules for $\bot, \wedge, \rightarrow, \forall$; the solution for full first-order language was provided much later by Seldin [247] and Stalmarck [263].

$[\wedge E]$	If $\Gamma, \varphi, \psi \vdash \chi$, then $\Gamma, \varphi \wedge \psi \vdash \chi$
$[\rightarrow E]$	If $\Gamma \vdash \varphi$ and $\Delta, \psi \vdash \chi$, then $\Gamma, \Delta, \varphi \rightarrow \psi \vdash \chi$
$[\forall E]$	If $\Gamma, \varphi[x/\tau], \vdash \chi$, then $\Gamma, \forall x \varphi \vdash \chi$
$[\neg E]$	If $\Gamma, \varphi \vdash \psi$ and $\Delta, \neg\varphi \vdash \psi$, then $\Gamma, \Delta \vdash \psi$

Incidentally, one should notice that still another combination of inference and proof construction rules for quantifiers is provided in this system. Both introduction rules are inference rules, whereas both elimination rules are proof construction rules (cf. Remark 2.1).

There are a lot of constructive proofs of normalization theorems for several ND systems showing for many logics that every proof in the respective system may be transformed into a normal proof. Theoretical value of these results is enormous, but from the practical point of view they are insufficient.[5] The situation is similar to that in sequent calculi; the constructive proofs of cut elimination show that we can always construct proofs using only analytic rules but they do not give any clue how to do it. What we need from the practical point of view is a proof search procedure for doing analytic proofs just from scratch.

Since we focus on the practical side of deduction, we dismiss the study of matters connected with the normalization of ND systems. A good exposition of these problems may be found, for example, in locus classicus Prawitz [220], in Troelstra and Schwichtenberg [276] or Negri and von Plato [193]. Very readable, although not yet completed, account is provided by Restall [232] on *http://consequently.org/writing/ptp*.

Instead, we describe some systems admitting analytic proofs only (and not only proofs). First, we introduce two analytic versions of standard ND: AND1 and AND2. The former is more restrictive but simpler; it is based on KE system. The latter is more flexible since the repertoire of primitive rules is richer. Completeness of both systems is easy to prove indirectly, by p-simulation of all KE-proofs (in fact, we show stronger result – step-wise simulation of every KE-proof tree), but for AND1 we also prove completeness directly because it will be needed for later modal extensions.

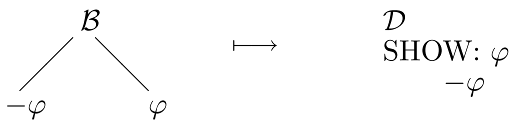
4.1.2 KE and ND

It is not difficult to notice a close relationship between KE and ND systems. It is apparent that ND α-elimination rules are ordinary tableau rules of the sort, whereas β-elimination rules are non-branching counterparts of ordinary

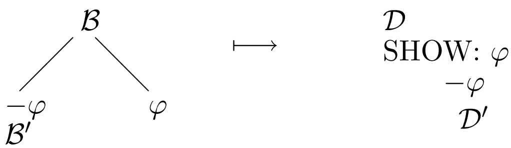
[5] The exception for classical logic is the work of Sieg mentioned above.

tableau branching rules. It is also obvious that suitable KE β-rules are elimination rules of ND system, thus all linear KE rules are ND inference rules. It makes KE, in a sense, closer to ND than to tableau systems. The only difference between KE and ND is branching introduced by (PB). But we can go a little step further and get rid of branching at all. Clearly, we can change trees into linear sequences in theory, but it is also easy in practice. Moment's reflection shows that everything that is derivable in KE by (PB) (or regressive cut in general) may be derived in ND by $[RED]$ with indirect assumption, as is shown in the following schemata:

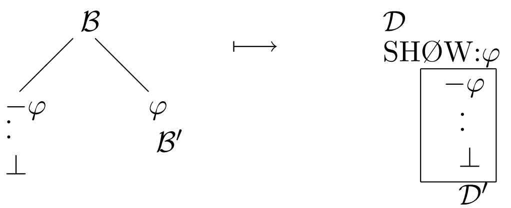
A)



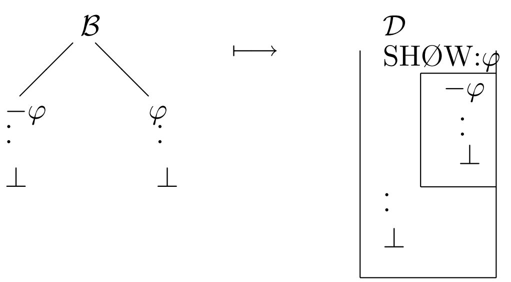
B)



C)



D)



Thus by induction on the number of application of (PB) one can step-wise simulate in ND every (open or closed) KE-tableau.

Although the idea is quite simple let us make it more precise and define a simple algorithm realizing the task. Recall that a part of a branch between two forks is called a *segment* of that branch (cf. Definition 1.7 in Section 1.1.5).

In case of an open tree we always choose the leftmost (or the shortest if we look for optimal derivation) open branch. If it has only one segment (no branchings), then we simply insert a suitable S-formula at the beginning thus receiving an open derivation of depth 1. In case of a branch consisting of more segments, we put a complement of the first formula in every segment as S-formula at the beginning of a subderivation corresponding to this segment. Every segment is thus an open subderivation. It is evident that for every open branch of the length n and consisting of k segments, we obtain an open derivation of the length $n + k$ and depth k, without boxes and cancelled S-formulae. So we obtain the optimal, in the sense of complexity, open derivation for tested formula.

In case of a closed tree, we will apply a procedure that is a particular case of transformation of trees into lists by algorithm PREORDER (cf. e.g. Aho and Ullman [3] or early Smullyan's [260] procedure of transformation of trees into the so-called nest structures). It means that we walk down from the root in a depth first manner from left to right. Because on every fork we first go to the left, then return, go to the right and return, we eventually pass through every fork three times. The order of visits dictates the operations we must perform.

the first step: Rewrite the first segment adding the first opening S-formula.

if it is the whole tree, **then** box the whole derivation, cancel prefix "SHOW:" and stop
else goto the next step.

the n-th step splits into three **cases** of visit in a fork.

1. **the first visit:** write the first element of the right segment as S-formula and the whole left segment as a subderivation, then do for this segment the following moves:
1.1. **if** it is the last segment of the branch, **then** close this subderivation in a box, cancel prefix "SHOW:" of the last S-formula and goto the last fork
else repeat 1. on the next fork.

2. **the second visit:** add the right segment without the first formula as a

continuation of the current derivation, next do step 1.1. for this segment.

3. **the third visit:**
   if it is the last fork in the tree (the rightmost one), **then** stop
   else goto the preceding fork.

After performing this procedure on a closed KE-tree we obtain ND proof of the length $n+1$ (where n is the number of nodes of simulated KE-proof) and the depth $\leq k$, where k is the maximal number of segments of the branches of this tree. We should note that in KE proof-tree we may have segments containing only one node. If it is the right segment, then after its subtraction (step 1. in the algorithm) we do not have any formulae in this segment. In such cases step 2. is performed trivially, i.e. we immediately go to the step 1.1.

For further considerations it is important to note that we can do reverse transformations as well. Every derivation being a linear nested structure may be turned into binary tree by treating, e.g. every subderivation as the right segment, and the part of its parent derivation (starting with the cancelled S-formula) as the left segment of the respective fork.

4.2 System AND1

Although we have shown that the full system of ND may step-wise simulate every KE proof, it is obvious that only some of the rules are necessary for that. Let us carefully examine what happen if we limit our ND resources to those indispensable for simulation. In such a system we have:

1. elimination rules as the only inference rules

2. $[RED]$ as the only proof construction rule

3. indirect assumptions as the only available assumptions

But this is not the whole story, as D'Agostino has proved that the completeness of KE for **CPL** requires only analytic applications of (PB).[6] On the basis of that we can admit only analytic applications of $[RED]$ as the only admissible proof construction rule.

The system thus obtained will be called AND1 (Analytical ND1). Formally, in our definition of derivation from Chapter 2 we must delete clauses

[6] In fact, only special sort of such analytic applications is needed. For details see Section 3.3.

4, 6, 8, 9, and 12. Moreover, we add proviso that the only admissible S-formulae are subformulae (or their complements) of the first S-formula. It is sufficient to add this requirement to clause A; in the revised form it reads:

A'. "$\mathcal{D}\oplus$SHOW:ψ" is a derivation of φ, for any ψ which belongs to $\overline{\{\neg\varphi\}}$

In case we define a derivation of φ from Γ, we must, of course, replace $\overline{\{\neg\varphi\}}$ by $\overline{\Gamma \cup \{\neg\varphi\}}$.

AND1 is adequate with respect to **CPL**. Soundness holds for any restricted version of the full system from Chapter 2, whereas completeness follows indirectly, from the completeness of KE and the simulation of all KE proofs by AND1. At first sight the completeness is allegedly doubtful as all the introduction rules are missing. But one should remember that our generic ND system is redundant – no introduction rules are necessary in AND1 as they are replaced by the elimination rules for the negated formulae.

Since KE allows of p-simulation of every tableau (most directly in the Smullyan's version, cf. [2]), so does AND1. If we want a more direct representation of tableau trees in ND we may add a new proof construction rule which embodies the application of β-rules from TS. It has the following shape:

[β] If Γ, β, $\beta_i \vdash \bot$, then Γ, $\beta \vdash \beta_j$, where $i \neq j \in \{1, 2\}$

In fact, Smullyan [259] proposed analytic ND based on [β] as the only proof construction rule. Admissibility of the above rule is easy to prove; below we display the schema of elimination in any ND derivation.

	Γ	
k	β	
	$\vdots$	
l	SHØW: β_j	$[n, \beta]$
$l+1$	β_i	$ass.$
	$\vdots$	
n	$\bot$	$(\bot I)$

$\longmapsto$

	Γ	
k	β	
	$\vdots$	
l	SHØW: β_j	$[n+1, RED]$
$l+1$	$-\beta_j$	$ass.$
$l+2$	β_i	$(k, l+1, \beta E)$
	$\vdots$	
$n+1$	$\bot$	$(\bot I)$

On the left we have the schema of the application of [β] in KM system, on the right we have the same result but obtained via the application of

$[RED]$; β_i is obtained here by the application of (βE) to β and the indirect assumption. So the elimination of every application of $[\beta]$ adds only one more line to a derivation and does not change its depth.

Let us note that in this way we may simulate the application of any branching tableau rule in ND system. For every rule of the schema:

$$\frac{\Gamma, \varphi}{\Gamma, \psi \quad | \quad \Gamma, \chi}$$

we may define ND proof construction rule of the shape:

If Γ, φ, $\psi \vdash \bot$, then Γ, $\varphi \vdash \chi$

This possibility will be thorougly exploited in the next chapters.

Clearly, in this way one may simulate not only TS for **CPL** but also for the first-order logic. KM is particularly useful for that aim since its original rule $(\exists E)$ is just like the respective rule of TS with side condition calling for a "new" variable.[7] It is a routine to prove DeMorgan laws in KM and on their basis to prove derivability of elimination rules for negated quantifiers – we leave it to the reader. Similarly, we may provide suitable rules for free logic in KM' (or KMP'): by means of DeMorgan laws (which are theses of free logic, as well) we may establish derivability of:

$(F\neg\forall E)$ $\neg\forall x\varphi,\ /\ Ey \wedge \neg\varphi[x/y]$, where y is new
$(F\neg\exists E)$ $\neg\exists x\varphi,\ Ey\ /\ \neg\varphi[x/y]$

With $(F\forall E)$ and $(F\exists E)$ they yield a complete set of rules for step-wise simulation of TS for free logic.

But what with universality? We have already mentioned that an open derivation for φ, as defined in Chapter 2, is not necessarily a disproof. In general, in ND-system it is not. The same applies to restricted version AND1 but in the present variant we can construct a derivation for φ in such a way that an open derivation is in fact a disproof of this formula. It should be obvious, since we have proved that AND1 can simulate every KE-tree, and KE is an universal system. But one should remember that a simulation is not the same as providing direct procedure of proof search for the respective system. The difference is like between proving cut elimination theorem for

[7] For those who prefer TS with rules introducing parameters instead of free variables the system KMP may be more convenient.

some SC by Gentzen method and direct proving of completeness of this SC without cut applying Hintikka method. The former result shows that it is possible "in principle" to build only analytic proofs, whereas the latter shows how to do it. So, having established that our ND system can exactly simulate KE-tableaux we are, in fact, more interested not in rewriting but in the direct use of the system for generating proofs and disproofs. For that we need a procedure that must eventually produce either a proof or an open but completed derivation in AND1. As a by-product we get a direct proof of both completeness and decidability for AND1-**CPL**. There are, at least, two reasons for paying some attention to this question:

First, it will be particularly useful for further extensions of ND system to modal logics. Many modal logics have analytic formalizations which are in fact quite easy to simulate by ND system, so in this case we can prove completeness indirectly. But not all logics considered later satisfy this condition. For example, although bimodal logics of linear time have some analytic SC or tableau formalizations, their simulation in ND is problematic. So the solutions we propose in Chapter 9, require direct completeness proof for these logics.

Second, the question of automated deduction in ND is interesting in itself, since nested structures, like derivations, lead to some restrictions on popular techniques applied in tableaux or resolution. Clearly, automated deduction is not the subject of this book, and we are not going to suggest that procedures of proof search defined in this Chapter and later, may be rivals to the popular programs of automated deduction. Such claims would require the implementation and testing of our procedures. That is why the matters of efficiency are not requisite for us, instead we are trying to offer algorithms as simple as possible. Some remarks concerning possible optimizations will be offered, however, by the end of this Section.

4.2.1 Hintikka Sets

Let us briefly recall the notion of a downward saturated set and Hintikka set needed for the completeness proof of AND1-**CPL**. For the sake of better control over saturation process in ND, it is also convenient to separate a weaker notion of linear saturation.

Definition 4.2 (Downward Saturated Set)

A. Γ is *l-saturated* (linearly saturated) for **CPL**, if the following conditions are satisfied:

1. if $\neg\neg\varphi \in \Gamma$, then $\varphi \in \Gamma$,
2. if $\alpha \in \Gamma$, then $\alpha_1 \in \Gamma$ and $\alpha_2 \in \Gamma$,
3. if $\beta \in \Gamma$ and $-\beta_i \in \Gamma$, then $\beta_j \in \Gamma$; for $i \neq j \in \{1, 2\}$;

B. Γ is *downward saturated* for **CPL**, if the following conditions are satisfied:

1. if $\neg\neg\varphi \in \Gamma$, then $\varphi \in \Gamma$,
2. if $\alpha \in \Gamma$, then $\alpha_1 \in \Gamma$ and $\alpha_2 \in \Gamma$,
3. if $\beta \in \Gamma$, then $\beta_1 \in \Gamma$ or $\beta_2 \in \Gamma$.

It is straightforward that every downward saturated set is l-saturated, but the opposite does not hold, since condition B.3 implies condition A.3. In case of l-saturated sets that are not downward saturated, every β-formula not satisfying condition B.3. will be called unused.

Definition 4.3 (Hintikka Set) Γ is *Hintikka set* for **CPL** if it is downward saturated and consistent: if a formula belongs to Γ, then its complement does not belong to Γ.

Lemma 4.1 (Satisfiability of Hintikka sets) *If Γ is Hintikka set for* **CPL***, then there is a valuation V such that:*
a) if $\psi \in \Gamma$, then $V(\psi) = 1$
b) if $\neg\psi \in \Gamma$, then $V(\psi) = 0$

Proof can be found in many places, e.g. in Smullyan [261].

The constructive completeness proof requires some algorithm of proof search which, in case of an open derivation, provides $\mathrm{U}(\mathcal{D})$ being a Hintikka set. There are a lot of ready procedures defined for tableau calculi that may be of some interest but the change in the format, from tableau to derivation, forces us to pay attention to some details. In fact, some types of strategies popular in tableaux are not at all applicable in ND setting due to the specific features of derivations.

Tableaux may be built in a breadth-first manner or in a depth-first manner, but in ND the former strategy fails. In fact, breadth-first strategy is quite natural and easy for tableaux; if we apply a rule to some formula above branchings we usually put the result of rule-application in all open branches at the same time. One can find many examples of tableau proof search procedures working in this way, for example, for **CPL** in Fitting [95], and for modal logics in Fitting [93], Goré [117]. In ND, tableau branchings correspond to a pair: subderivation and its outer derivation, and they are worked out one after another, so we are not able to use some formula immediately in all possible places. Typically, the result of a rule-application is put in some subderivation which is then boxed, so we must be able to repeat the rule to the same formula again in the subderivation to come. It is like in a tableau with branches – we would not put the result of a rule-application to some formula above branchings in all open branches going through this formula, but follow only one of them till the very end, and if it is closed we would return to the nearest fork. So in ND only depth-first strategies really work.

Is it a loss for ND? It depends. In case of an open tableau the depth-first construction may lead to quick answer – if we find an open branch before saturation of remaining thousand branches. But such a strategy has some drawbacks, as well. For example, the descriptions of algorithms often apply several forms of marking formulae as used (cf. Goré [117]), which makes the control over the performance of the procedure easier. Marking formulae as used is simple in case of algorithms working in a breadth-first manner, but in case of a depth-first manner strategies it is rather complicated. We must remember that a formula marked as used is not necessarily "used" in all branches going through the node decorated with this formula. Every time some branch is being closed and we must return to the nearest fork, all formulae used below – but in this closed branch – must be again ready for use in the next branch. In ND such a situation appears when the conclusion of the application of some rule is in the subderivation which is boxed. All premises of this rule must be again ready for use, so if they were marked as used, this sign must be deleted.

In case of derivations we have additional difficulty. Since the standard tableau rules have one premise we can base a procedure on simple instruction of the type "case of". At every stage just take the first (e.g. the highest and leftmost) unused formula and apply suitable rule marking the formula as used and go to the next one. Clearly, such strategy is not very efficient, but it is very simple to implement, even if we add some priority strategies that may shorten proof-search (e.g. first α-rules, then β-rules).

But this strategy is already not well suited for KE; because we have rules with two premises and with no premises at all, the time and conditions for their use must be specified in some other way referring rather to the whole (branch of a) tableau. The same applies for ND. In case of two-premise rules we must search the whole $\mathrm{U}(\mathcal{D})$ for matching suitable pairs. Similarly for deciding when to start a subderivation and which pair of formulae take as Show-formula and the corresponding indirect assumption. Because of that, every proof-search procedure for ND must be more involved than the simplest solutions applied for tableaux. Consequently, it is more difficult to prove its fairness and termination, and in case of automation, it is more difficult to implement and more time- and memory-consuming.

On the other hand, in a procedure based on checking the whole $\mathrm{U}(\mathcal{D})$, each time we start a new stage, we can easily involve some priority strategies that may shorten the proof-search. For example, it is possible to make consistency-check before a subderivation is started, whereas in standard tableau procedures it is performed at the end of an algorithm because it is expensive (cf. Fitting's remarks in [95]). We will say more about it in the remarks concerning the optimization.

4.2.2 Proof Search Procedure for AND1

We have already mentioned that an open derivation for φ, as defined in Section 2.5.3, is not necessarily a disproof, in general in ND-system it is not. The same applies to the restricted AND1 but in this variant we can construct a derivation for φ in such a way that an open derivation provides a disproof of this formula. D'Agostino sketches an algorithm of the sort. He calls "E-analyzed" any KE-tableau, where we applied rules to all α-formulae and to these β-formulae for which we had minor premises. It is evident that in contrast to ordinary tableau such E-analyzed tableau, although linear, need not be completed in the sense that all formulae were decomposed. We can have many β-formulae that were not used because we do not have any minor premises. In such a case we should apply (PB) for some missing premise (and its negation) and proceed further in accordance with our rules. Such a procedure must eventually produce either a proof (closed tree) or an open but completed tableau. The inspection of any open branch makes highlights the fact that the set of its formulae is downward saturated, hence we got both, completeness and decidability. Moreover, the requirements concerning admissible applications of (PB) are even sharpened in the sense that not all analytic applications are needed.

In our ND-system we can follow D'Agostino procedure quite closely. Of course, instead of a cut we start with an S-formula for a missing premise (or its negation, it does not matter which we choose – the subderivation corresponds to one branch and in case it closes, the continued outer derivation corresponds to the second). Below, we shortly present an algorithm which is a deterministic version of KE-canonic procedure from D'Agostino [2]. The principal idea is that the introduction of new subderivations (the counterpart of branching) should be postponed as much as possible, so we have some priority strategy: first l-saturation, then downward saturation. Let us note that it is not sufficient for putting the process of subderivation generation to the very end. Consistency test is also prerequisite before a new subderivation begins, which is possible but has some drawbacks (cf. a discussion on optimization). So in procedures to follow we put consistency test at the end of an algorithm, after the full downward saturation of U($\mathcal{D}$).

The application of the basic construction in later chapters (for proving completeness of analytic ND systems for modal logics), requires the division of the procedure into two modules. We define separately an algorithm for saturation and for proof-search; in the latter we repeatedly call the former procedure and consistency test of the whole derivation.

SAT(U($\mathcal{D}$)) PROCEDURE

Input: the set U($\mathcal{D}$) of any derivation.

Output: downward saturated U($\mathcal{D}$).

1. **Until** U($\mathcal{D}$) is not l-saturated **do**
   apply the rules of elimination to any U-formula.
2. **If** U($\mathcal{D}$) is not downward saturated, **then**
   choose the first unused β-formula,
   start a new subderivation (add "SHOW:$\beta_i \oplus -\beta_i$"),
   and **goto** step 1.
   else stop.

ALGORITHM 1 (proof search of φ in **CPL**):

0. **Start:** Write down "SHOW:$\varphi \oplus -\varphi$" as the beginning of a derivation.
1. **Call** procedure SAT(U($\mathcal{D}$)).
2. **If** U($\mathcal{D}$) is inconsistent, **then**
   apply ($\perp I$) and close the current subderivation by [RED]:
   2.1. **If** the degree of closed derivation = 1, **then**
      stop: $\vdash \varphi$
      else
      goto step 1.
   else
   stop: φ has no proof.

We establish *termination* and *fairness* of these procedures. To that aim we must show that every compound formula is used as many times as it is necessary in a derivation, and that both, the length of every subderivation and their number, is finite. First we show:

Lemma 4.2 *SAT(U($\mathcal{D}$)) is terminating.*

PROOF Let $\mathcal{D}$ be any derivation of the depth k. The procedure first performs l-saturation by the application of elimination rules to every α-formula and every β-formula for which minor premise is present. Every compound formula which was used this way and every literal is marked with **U** as used. By the application of a rule we mean that we write down a conclusion if it is not yet present in U($\mathcal{D}$), thus we can mark as used also such β for which we do not have minor premise but at least one of the possible conclusions is already present. Note also that we are looking for minor premises for β-rules among all members of U($\mathcal{D}$) at the current stage; all nonboxed formulae, unmarked and marked are at our disposal. Every unused β is signed by **W** as waiting. Because U($\mathcal{D}$) is finite and every rule satisfies subformula property so after finishing Step 1. we obtain a derivation $\mathcal{D}' \supseteq \mathcal{D}$ of the depth k, where U($\mathcal{D}'$) = $\mathbf{U} \cup \mathbf{W}$. If the set **W** is empty, then U($\mathcal{D}'$) is downward saturated, and the procedure stops. Otherwise, we have a finite set of n unused β-formulae. Step 2 introduces a new subderivation which provides a lacking minor premise for one of the element of **W**, and we return to Step 1. In this way **W** decreases since at least one β from it is used ("at least" since the extension of $\mathcal{D}'$ made in Step 2 may, through repetition of Step 1, yield minor premises or conclusions for other β-formulae from **W**, as well). As a result, after the repeated sequence of application of Steps 1 and 2 we must obtain $\mathcal{D}' \supseteq \mathcal{D}$ such that U($\mathcal{D}'$)=**U**, and the depth of $\mathcal{D}'$ is $k+i$, where $i \leq n$. ■

Before showing the termination and fairness of Algorithm 1, we must remark that any derivation is a kind of structure that can be redefined as a tree. The most immediate way would be to use again the scheme from Section 4.1.2, the one used to show that any KE-tree may be turned into a derivation. Now we use different and more interesting mapping which is not so obvious but is more direct with respect to the derivation construction. In this approach let us take the whole subderivation connected with some Show-line as the node of the tree. We will refer to so defined tree as $\mathcal{T}(\mathcal{D})$.

Definition 4.4 (Tree $\mathcal{T}(\mathcal{D})$) The root of $\mathcal{T}(\mathcal{D})$ is the only derivation of degree 1 and every subderivation is a node of this tree. For any subderivation of degree k take as the children of this node all subderivations of degree $k+1$ that are nested in it.

The definition of a derivation and the definition of related tree imply some simple facts that we record below:

- any branch of such a tree is a sequence of the length $i+j>0$, where:
  - the first i nodes are sets of U-formulae belonging to first i open subderivations (open nodes);
  - the remaining j nodes are nested boxes containing closed subderivations (closed nodes).
- For any branch either i or j may be 0, but every branch must have the length at least 1.
- Closed nodes never come before open nodes on any branch.
- All branches of the tree corresponding to a proof have only closed nodes ($i=0$).
- Exactly one branch of the tree corresponding to an open derivation consists of only open nodes ($j=0$).

Lemma 4.3 *Every branch in $\mathcal{T}(\mathcal{D})$ must terminate.*

Proof First consider any branch that does not contain boxed nodes (each subsequent derivation is open). In this case we have termination as a consequence of the preceding lemma since we have shown that realization of SAT(U($\mathcal{D}$)) gives only finite branches. Every new subderivation (=node) is introduced by appealing to some waiting β. Since due to subformula property the number of all different β-s that may appear in $\mathcal{D}$ is finite we must only show that no repetition is possible in the sequence of nested and open subderivations. But β once used to create a subderivation is marked as used and this mark could have been deleted only when this subderivation is boxed, which is impossible by definition of our branch. So no β can be the source of more than one subderivation in this branch.

The second Step of the algorithm makes some subderivation (node) boxed. Obviously, any branch that contains some boxed nodes must be finite, since if one (the current) degree is boxed the branch terminates; we can only put in boxes the preceding nodes but we cannot add more. Clearly, further performance of the procedure may stop boxing at some degree k and we may be forced to start a new subderivation of degree $k+1$ but this is the latest node of the new branch not the continuation of the old one. ■

In general, every β may generate new subderivations many times but only on different branches having the same parent. We must show that it may be done only finitely many times (i.e. $\mathcal{T}(\mathcal{D})$ is finitely branching).

Lemma 4.4 *Every node in $\mathcal{T}(\mathcal{D})$ has only a finite number of children.*

PROOF Take any subderivation of degree $k \geq 1$ and assume that we are precisely at this stage of procedure realization (call it stage n), where for the first time the point 2 of SAT(U($\mathcal{D}$)) is executed (so we are in a node with no children, so far). In order to obtain any nested subderivation of degree $k+1$ all compound formulae in U($\mathcal{D}$) must be marked as used except at least one β. Obviously, if we start a subderivation of degree $k+1$, and it remains open for the rest of the construction, it will be the only child of our node and we are done. So assume that at some later stage this node will be boxed. Then it is possible that after l-saturation of U($\mathcal{D}$) we can start another subderivation of degree $k+1$. We show that it may be done only finite number of times, by induction on the union of lengthes of all β-s in U($\mathcal{D}$) unused (marked as **W**) at the stage n. By assumption there must be at least one such β.

Basis: considered β has length 1. So both S-formula and assumption of the next subderivation are literals. Even if this subderivation is boxed at later stage, there is no new β-s in the outer derivation of degree k that can start another subderivation, since the only new formula in U($\mathcal{D}$) of this derivation is a literal (previous S-formula).

We show that our result holds if the sum of lengthes of unused β-s is n, provided it holds for any $k < n$. Again consider the situation that we started the first subderivation and it is boxed. This time the number of β-s marked with **W** may even increase (e.g. canceled S-formula could have been α built from two β-s), but since our starting β is now used (marked as **U**) the union of the lengthes of remaining ones must be at most $n-1$ and induction hypothesis applies.[8] ■.

[8] The same result may be demonstrated by showing that any β may be used to creation of new subderivations at most 2^{n+1} times, where n is the number of β-formulae preceding this β in the derivation.

By König lemma, and lemmata 4.3 and 4.4, $\mathcal{T}(\mathcal{D})$ must have finite number of nodes. Since each node is a finite set of formulae we have proven a termination of the algorithm. So it follows:

Lemma 4.5 *Every derivation performed by algorithm 1 is finite.*

We let the reader establish the next

Lemma 4.6 *If $\mathcal{D}$ is completed, then U($\mathcal{D}$) is a downward saturated set.*

Both lemmata (4.5 and 4.1) give us completeness (as well as decidability) of AND1-**CPL**.

Theorem 4.1 (Adequacy) *ADN1 is an adequate formalization of* **CPL**

PROOF Soundness was proven earlier for full KM, so only completeness remains. Assume that $\nvdash \varphi$, so, in particular, a derivation carried via algorithm 1 is finite and open. U($\mathcal{D}$) of this derivation is Hintikka set, so – according to lemma 4.1 – it is satisfiable. $\neg\varphi \in$U($\mathcal{D}$), so $\nvDash \varphi$. ■

4.2.3 Optimization

Let us focus on some questions that may be important for making proof search process in AND1 more efficient.

Algorithm 1 is certainly not very efficient. Consistency test is applied only after downward saturation of U($\mathcal{D}$). There are some advantages of such a solution; [95] pointed out that consistency test is expensive (necessity of searching of all U($\mathcal{D}$)), so it is better to apply it at the end. Moreover, in this case we may limit the search of U($\mathcal{D}$) only to literals (atomic consistency test). But this is an advantage in case of tableau systems, where rules simply break down single compound formulae into pieces. In case of ND it is of no use because β-rules lead necessarily to checking the whole U($\mathcal{D}$) much earlier, when we are looking for minor premises or direct subformulae of respective β-s. On the other hand, a separation of the saturation process from consistency test leads potentially to numerous repetitions of the same sequence of inferences. For example, even the first performance of SAT-procedure may build a derivation of the big depth, where a pair of complementary formulae occurs e.g. already in the subderivation of degree 1. In such a case, algorithm 1 will be repeatedly calling subprocedure SAT(U($\mathcal{D}$)) and making consistency test again and again, before the proper level of a derivation will be boxed. How to avoid such cumbersome situation?

One of the possible optimization would be to replace the requirement of closing the current subderivation by more radical requirement of closing all subderivations up to this degree, where inconsistency actually occurs. For instance: let the current subderivation have degree 20; the application of Step 2. finds φ in outer subderivation of degree 3, and $-\varphi$ in subderivation of degree 5. In such a situation we should close not only the subderivation of degree 20 but all outer subderivations up to the (and including) subderivation of degree 5, and after that repeat step 1.

More radical solution would be to apply a different algorithm with different order of steps: (a) l-saturation $\longrightarrow$ (b) consistency test $\longrightarrow$ (c) opening a new subderivation. In such a procedure, a deepening of the derivation is made only after linear extension and checking consistency, and the depth is always increased by one degree only (then again linear extension and consistency test). The strategy executed by such an algorithm may lead to the end much faster, particularly in case of theses, but it has two disadvantages:

1. Proving termination and fairness of such an algorithm is more complicated than for algorithm 1.[9]

2. Algorithm of this kind does not work for many modal logics (we cannot prove completeness) – it will be demonstrated in Chapter 10.

Hence, we keep Algorithm 1 as the basis of our future generalizations.

4.3 System AND2

In AND1 we severely limited our deductive resources but we got analytic decision procedure of the same level of complexity as KE. This is indeed in contrast with the usual practice and, of course, such system may be seen as not a genuine ND-system by many purists. In any case, we can keep the whole ND-system from Chapter 2 and think about the restricted one as the basis for an algorithm devised on a full system. Such a solution admits producing shorter and smarter proofs if possible, moreover, in contrast to ordinary ND, we can construct not only proofs but also disproofs.

We can ask if it is possible to obtain less restrictive but still analytic and universal ND system. Because AND1 follows closely KE it is instructive to analyze some properties of this system.

[9]Such a solution was applied by the author in [147].

In KE one can find that the proper choice of a cut-formula may have significant impact on the complexity of a proof. Bad choice often leads to many further branchings and repetitions of sequences of inferences in many branches, whereas a "good" choice may considerably shorten the proof. Obviously, the same applies to AND1, where by cut-formula we mean a formula and its complement used to initiate a subderivation. Let us compare two proofs of the same thesis.

1	SHØW: $(p \to q \wedge r) \to (p \to q \vee s) \wedge (p \to r \vee t)$	$[31, RED]$
2	$\neg((p \to q \wedge r) \to (p \to q \vee s) \wedge (p \to r \vee t))$	$ass.$
3	$p \to q \wedge r$	$(2, \alpha E)$
4	$\neg((p \to q \vee s) \wedge (p \to r \vee t))$	$(2, \alpha E)$
5	SHØW: p	$[15, RED]$
6	$\neg p$	$ass.$
7	SHØW: $p \to q \vee s$	$[11, RED]$
8	$\neg(p \to q \vee s)$	$ass.$
9	p	$(8, \alpha E)$
10	$\neg(q \vee s)$	$(8, \alpha E)$
11	$\bot$	$(6, 9, \bot I)$
12	$\neg(p \to r \vee t)$	$(4, 7, \beta E)$
13	p	$(12, \alpha E)$
14	$\neg(r \vee t)$	$(12, \alpha E)$
15	$\bot$	$(6, 13, \bot I)$
16	$q \wedge r$	$(3, 5, \beta E)$
17	q	$(16, \alpha E)$
18	r	$(16, \alpha E)$
19	SHØW: $p \to q \vee s$	$[25, RED]$
20	$\neg(p \to q \vee s)$	$ass.$
21	p	$(20, \alpha E)$
22	$\neg(q \vee s)$	$(20, \alpha E)$
23	$\neg q$	$(22, \alpha E)$
24	$\neg s$	$(22, \alpha E)$
25	$\bot$	$(17, 23, \bot I)$
26	$\neg(p \to r \vee t)$	$(4, 19, \beta E)$
27	p	$(26, \alpha E)$
28	$\neg(r \vee t)$	$(26, \alpha E)$
29	$\neg r$	$(28, \alpha E)$
30	$\neg t$	$(28, \alpha E)$
31	$\bot$	$(18, 29, \bot I)$

1	SHØW: $(p \to q \wedge r) \to (p \to q \vee s) \wedge (p \to r \vee t)$	$[23, RED]$
2	$\neg((p \to q \wedge r) \to (p \to q \vee s) \wedge (p \to r \vee t))$	*ass.*
3	$p \to q \wedge r$	$(2, \alpha E)$
4	$\neg((p \to q \vee s) \wedge (p \to r \vee t))$	$(2, \alpha E)$
5	SHØW: $p \to q \vee s$	$[14, RED]$
6	$\neg(p \to q \vee s)$	*ass.*
7	p	$(6, \alpha E)$
8	$\neg(q \vee s)$	$(6, \alpha E)$
9	$q \wedge r$	$(3, 7, \beta E)$
10	q	$(9, \alpha E)$
11	r	$(9, \alpha E)$
12	$\neg q$	$(8, \alpha E)$
13	$\neg s$	$(8, \alpha E)$
14	$\bot$	$(10, 12, \bot I)$
15	$\neg(p \to r \vee t)$	$(4, 5, \beta E)$
16	p	$(15, \alpha E)$
17	$\neg(r \vee t)$	$(15, \alpha E)$
18	$q \wedge r$	$(3, 16, \beta E)$
19	q	$(18, \alpha E)$
20	r	$(18, \alpha E)$
21	$\neg r$	$(17, \alpha E)$
22	$\neg t$	$(17, \alpha E)$
23	$\bot$	$(20, 21, \bot I)$

In the latter proof we did not use the first β-formula in a proof to start a subderivation but we have chosen the second one because its direct components are longer, so the additional assumption gives us more information. Unfortunately, there is no recipe for the optimal choice of a cut-formula; the strategy of taking the longest formula often leads to good results but it is certainly not universal. If the chosen formula does not enable many new inferences, in particular, if it does not give us minor premises for application of β-rules to unused formulae, we should rather follow different strategies.[10]

The analysis of many semantical methods of checking validity, like short (indirect) truth-table test, or the method of Quine'a [226], shows that in case of necessity of applying additional assumptions very good results are often obtained if we consider a variable having relatively many occurrences

[10] As we will see in Chapter 10 the application of cut in modal logics may lead to other problems, as well.

in analyzed formula (or a set of formulae)[11] Formally, it is a counterpart of a cut applied on literal. In fact, in the first proof we have used this strategy choosing p as a Show-formula and $\neg p$ as an indirect assumption, since this variable has the greatest number of occurrences in the proven thesis. But we did not obtain satisfying results since elimination rules do not allow for immediate application of indirect assumption. This time the lack of introduction rules in AND1 is the source of a problem. The next example shows that the application of introduction rule leads to a proof even shorter than the second one in AND1.

1	SHØW: $(p \to q \wedge r) \to (p \to q \vee s) \wedge (p \to r \vee t)$	$[20, RED]$
2	$\neg((p \to q \wedge r) \to (p \to q \vee s) \wedge (p \to r \vee t))$	$ass.$
3	$p \to q \wedge r$	$(2, \alpha E)$
4	$\neg((p \to q \vee s) \wedge (p \to r \vee t))$	$(2, \alpha E)$
5	SHØW: p	$[11, RED]$
6	$\neg p$	$ass.$
7	$p \to q \vee s$	$(6, \beta I)$
8	$\neg(p \to r \vee t)$	$(4, 7, \beta E)$
9	p	$(8, \alpha E)$
10	$\neg(r \vee t)$	$(8, \alpha E)$
11	$\bot$	$(6, 9, \bot I)$
12	$q \wedge r$	$(3, 5, \beta E)$
13	q	$(12, \alpha E)$
14	r	$(12, \alpha E)$
15	$q \vee s$	$(13, \beta I)$
16	$p \to q \vee s$	$(15, \beta I)$
17	$\neg(p \to r \vee t)$	$(4, 16, \beta E)$
18	$r \vee t$	$(14, \beta I)$
19	$p \to r \vee t$	$(18, \beta I)$
20	$\bot$	$(17, 19, \bot I)$

Combining the strategy of the longest formula with the full assortment of ND rules can even shorten the proof as the next two examples show:

[11] This is somewhat related to the strategies from resolution provers, like ordering or selection function, which are applied to reduce the number of useless inferences. But it is not possible to transfer these strategies directly to ordinary ND since it does not work on clauses – one more good reason to introduce RND in the next section.

1	SHØW: $(p \to q \wedge r) \to (p \to q \vee s) \wedge (p \to r \vee t)$	$[19, RED]$
2	$\neg((p \to q \wedge r) \to (p \to q \vee s) \wedge (p \to r \vee t))$	*ass.*
3	$p \to q \wedge r$	$(2, \alpha E)$
4	$\neg((p \to q \vee s) \wedge (p \to r \vee t))$	$(2, \alpha E)$
5	DØW: $p \to q \vee s$	$[10, COND]$
6	p	*ass.*
7	$q \wedge r$	$(3, 6, \beta E)$
8	q	$(7, \alpha E)$
9	r	$(7, \alpha E)$
10	$q \vee s$	$(8, \beta I)$
11	$\neg(p \to r \vee t)$	$(4, 5, \beta E)$
12	p	$(11, \alpha E)$
13	$\neg(r \vee t)$	$(11, \alpha E)$
14	$q \wedge r$	$(3, 12, \beta E)$
15	q	$(14, \alpha E)$
16	r	$(14, \alpha E)$
17	$\neg r$	$(13, \alpha E)$
18	$\neg t$	$(13, \alpha E)$
19	$\bot$	$(16, 17, \bot I)$

1	SHØW: $(p \to q \wedge r) \to (p \to q \vee s) \wedge (p \to r \vee t)$	$[15, COND]$
2	$p \to q \wedge r$	*ass.*
3	DØW: $p \to q \vee s$	$[8, COND]$
4	p	*ass.*
5	$q \wedge r$	$(2, 4, \beta E)$
6	q	$(5, \alpha E)$
7	r	$(5, \alpha E)$
8	$q \vee s$	$(6, \beta I)$
9	DØW: $p \to r \vee t$	$[14, COND]$
10	p	*ass.*
11	$q \wedge r$	$(2, 10, \beta E)$
12	q	$(11, \alpha E)$
13	r	$(11, \alpha E)$
14	$r \vee t$	$(13, \beta I)$
15	$(p \to q \vee s) \wedge (p \to r \vee t)$	$(3, 9, \alpha I)$

The last proof is the shortest but not the simplest in the important sense its structure being more complex due to the greater number of nested subderivations.

In fact, the application of introduction rules often renders avoiding subderivations possible, whereas in AND1 it is impossible. The next two examples illustrate the point.

1	SHØW: $((p \to q) \to (r \to s)) \to (r \to (q \to s))$	$[16, RED]$
2	$\neg(((p \to q) \to (r \to s)) \to (r \to (q \to s)))$	$ass.$
3	$(p \to q) \to (r \to s)$	$(2, \alpha E)$
4	$\neg(r \to (q \to s))$	$(2, \alpha E)$
5	r	$(4, \alpha E)$
6	$\neg(q \to s)$	$(4, \alpha E)$
7	q	$(6, \alpha E)$
8	$\neg s$	$(6, \alpha E)$
9	SHØW: $p \to q$	$[13, RED]$
10	$\neg(p \to q)$	$ass.$
11	p	$(10, \alpha E)$
12	$\neg q$	$(10, \alpha E)$
13	$\bot$	$(7, 12, \bot I)$
14	$r \to s$	$(3, 9, \beta E)$
15	s	$(5, 14, \beta E)$
16	$\bot$	$(8, 15, \bot I)$

1	SHØW: $((p \to q) \to (r \to s)) \to (r \to (q \to s))$	$[12, RED]$
2	$\neg(((p \to q) \to (r \to s)) \to (r \to (q \to s)))$	$ass.$
3	$(p \to q) \to (r \to s)$	$(2, \alpha E)$
4	$\neg(r \to (q \to s))$	$(2, \alpha E)$
5	r	$(4, \alpha E)$
6	$\neg(q \to s)$	$(4, \alpha E)$
7	q	$(6, \alpha E)$
8	$\neg s$	$(6, \alpha E)$
9	$p \to q$	$(7, \beta I)$
10	$r \to s$	$(3, 9, \beta E)$
11	s	$(5, 10, \beta E)$
12	$\bot$	$(8, 11, \bot I)$

The first proof is performed in AND1, the second is not since we applied (βI) in line 9. It is evident that due to the application of introduction rules we may not only shorten proofs but also reduce their depth. The examples considered were simple and the difference between them not drastic; but it is possible to find examples that in AND1 lead to the significant growth of the complexity of proof with respect to proofs obtainable in full ND. On the other hand, it seems that if we admit all the rules again we will loose analyticity and universality. However, if we analyze the examples of proofs where introduction rules were applied we should note that they are still analytic. All applications of $[COND]$, as well as, all applications of introduction rules in these proofs are analytic.

The problem is how to proceed with proof construction to save analyticity while using introduction rules and $[COND]$. We will try to show that the problem of saving analyticity for full ND is not as hard as it seems. It is instructive to compare proof search in ND with indirect truth-table test in **CPL**. In common opinion, tableaux are better than the full truth-table method, and they are rather compared to indirect truth-table test. We have already mentioned (Section 3.3.) D'Agostino's result showing that from the standpoint of relative proof length complexity standard TS is worse than KE since the latter may p-simulate tableau while the opposite does not hold. In fact, D'Agostino has shown that TS, in this respect, is even worse; it cannot p-simulate full truth-table test.[12] If tableau method, at least in case of classical logic, is indeed the formalized counterpart of informal indirect truth-table test, as D'Agostino seems to suggest, it would show that this method is also equally inefficient in the worst cases. But it is a mistaken view.

Closer analysis shows that indirect truth-table test tends to work faster than tableaux, and even faster than KE. The source of this phenomenon lies in the fact that in an indirect truth-table test we are also using semantic counterpart of cut in analytic form; we have mentioned this above during the discussion on the strategy of choosing as cut-formulae literals with the great number of occurrences. In this way we may work faster than in standard TS but not faster than in KE. Note, however, that while checking formulae indirectly we not only proceed from the value of a formula (and perhaps some of its subformulae) to values of its parts, but very often in the reverse direction; e.g. having established some β_i as true we simply treat the whole β as true. For example, when testing the thesis from the last two examples it seems obvious that after the first stage of establishing values of variables which leads to claiming that both r and q are true and s is false, we will use this information to establish the value of implication in the antecedent.

[12]But remember that these results should not be treated as indicating the weakness of tableaux for automated deduction in general, since the possibility of obtaining shorter proofs does not mean that the space of proof search is also smaller; sometimes it is just the opposite. We have mentioned about that already in Chapter 3.

By means of such shortcuts, we do not need to consider different variants of falsifying valuations, whereas in KE we have to apply (PB).

But such informal steps that considerably accelerate checking, in an obvious way correspond to ND-rules that we gave up in the preceding section when looking for decision procedure. This is because analyticity is commonly (and mistakenly) considered as equivalent to having only decomposition rules such as our elimination rules. Mainly for this reason the authors of several methods tend to be confined to this type of rules only. In Chapter 3 we have called such rules *strictly analytic*, because subformula property is involved in their schemata. On the other hand, building-up rules like our introduction rules do not have subformula property per se, but we can make them analytic in the more general sense, putting restrictions on their use, exactly as with cut (cf. a discussion in Section 3.3). So, once again, we can come back to our first ND system with all inference and proof-construction rules but with the following general analytic restrictions on the clauses 1., 4., 6., and 8. in the definition of a derivation:

1'. "$\mathcal{D}\oplus$SHOW:ψ" is a derivation of φ for any $\psi \in \overline{\{\neg\varphi\}}$

4'. "$\mathcal{D} \oplus \neg\neg\psi$" is a derivation of φ, provided $\psi \in$U$(\mathcal{D})$ and $\neg\neg\psi \in \overline{\{\neg\varphi\}}$

6'. "$\mathcal{D} \oplus \alpha$" is a derivation of φ, provided $\{\alpha_1, \alpha_2\} \subseteq$U$(\mathcal{D})$ and $\alpha \in \overline{\{\neg\varphi\}}$

8'. "$\mathcal{D} \oplus \beta$" is a derivation of φ, provided $\beta_i \in$U$(\mathcal{D})$ and $\beta \in \overline{\{\neg\varphi\}}$

We will call this version AND2. The system is adequate for **CPL**; soundness holds because it is a restricted version of sound ND system, completeness holds because it is an extension of AND1 which is complete. It may be extended to first-order system in the way presented in Section 4.2 or by addition of analytic restriction on any of the set of rules introduced in Section 2.7. In fact, we must restrict only the applications of $(\exists I)$ requiring that the introduced $\exists x\varphi$ must belong to $\overline{\{\neg\varphi\}}$; several forms of $[UNIV]$ are made analytic already by condition 1'. above concerning all S-lines that may be introduced into a derivation.

In the end we make some sketchy remarks concerning proof search in AND2. We have merely shown that shorter proofs may be obtained in AND2 than in KE. But it does not mean that such short proofs may be found in shorter time. The problem is similar as with KE compared to ordinary TS with respect to efficiency. A creation of shorter KE proofs may need longer proof search since the application of β-rules or (PB) requires checking the whole branch (cf. a discussion on proof search strategies in Section 4.2.1). Serious investigation on deductive capabilities of AND2 would require a definition of proof search algorithm that applies all the rules in an analytic way and its implementation to test how it works on hard problems. But even without that we may point out to some advantages and disadvantages.

Certainly, the admission of additional rules leads to a greater flexibility of proof construction. We may obtain shorter proofs and it is possible to apply richer set of strategies in their search. But this obviously implies costs – any proof search procedure for AND2 must be much more complicated due to the number and variety of rules. A tree of possible choices in proof search must have greater branching factor than such a tree in AND1. Also an amount of memory in case of automatization, must be much greater than for AND1.

Having all these complications in mind, we are not going to exploit the problem of proof search procedure for AND2 in such detail as we did for AND1. For our needs (in exploring analytic ND for modal logics in Chapter 10) the results obtained for AND1 are sufficient. But it is worth mentioning that there are some interesting approaches to this theme that may be simulated in AND2. All of them are connected with the existing provers or other computer programs for building ND proofs.

The main problem is how to avoid indeterminacy implied by the uncontrolled use of introduction rules. Many programs are based on some proof search procedures that use troublesome rules in an analytic way, so they may be adjusted to work in AND2. There is no space here for their thorough analysis and comparison. Terminology and notation concerning proof-search strategies is not fixed but varies from author to author, which makes such an attempt very difficult after all. So we limit our considerations to some sketchy remarks concerning the main ideas.

Many authors underline that introduction rules must be applied only in special circumstances. Generally, proof search must be goal-oriented, or interest-driven (cf. e.g. Pollock [216]), hence the call for e.g. $(\vee I)$ must be justified by its need as a major premise for application of some two-premise rule. Note that this complies strictly with our informal remarks given above. In some proposed procedures (e.g. in [46]) their application is treated as a last resort.

Finally, we should say that the analysis of different algorithms involved in ND-based provers exhibits one more reason for treating KM as the most suitable deductive tool. In many approaches, to control better the interest-driven application of troublesome rules, the set of processed data is divided into two lists: 1. premises and active assumptions with already-deduced (and active) conclusions (antelines in [204], conclusions in [217], proof-list in [46]), and 2. formulae we need to proceed further (goalstack in [204], interests in [217], goal-list in [46]). Notice that this division is already present in KM due to the division of formulae in a derivation into U- and S-lines. So using KM format for AND2 allows for a direct realization of strategies of proof search involved in several provers. This alternate process of introducing new U-lines and S-lines in KM corresponds well also to the combination of forward and backward reasoning present in many descriptions of automated proof search in ND (cf. Dyckhoff [81] or Pollock [216]). Roughly speaking (and simplifying a bit), by means of inference rules we realize forward reasoning, whereas by introducing new S-lines we proceed with backward reasoning.

4.4 Resolution and ND Combined

The results obtained thus so far show that ND systems, after some modifications, may be also treated as universal and general systems, i.e. good for performing different deductive tasks, and able to simulate strategies taken from other systems. In particular, one may also use them for automated deduction. It is important for both AND1 and AND2 not to lose natural and intuitive character of their deductive tools, and to extend the pedagogical value of ND by increasing its universality. Still, neither AND1 nor AND2 is as general form of ND, as it is possible.

We have compared AND2 and AND1 with respect to the length of derivations. The related question is the complexity of derivations measured by their depth. In ND the number of nested subderivations is the counterpart of the number of branches in TS or KE. In all these systems additional subderivations/branches may significantly increase the length of a derivation because some sequences of inferences may be repeated in several places. Let us recall that in KE we may obtain significant improvement over standard TS in this respect and AND1 behaves like KE. AND2, due to inclusion of introduction rules, obtains often shorter derivations also because we are not forced to introduce as many additional assumptions as in AND1. So shorter AND2-derivations are not only the result of performing more direct inferences but also of the reduction of repeated steps in different subderivations. But it is not possible to avoid additional subderivations in general, similarly as it is not possible to eliminate branching in TS or KE;

a variant of AND restricted in this way would be incomplete. It is important to consider if it is possible to obtain ND-like system wherein we may construct linear derivations in the strict sense, i.e. of the depth=1.

Notice that so far we have considered ND in comparison with systems like TS and KE, or indirect truth-value tests. We have not taken up the problem of how to make use, in ND, of resolution or DP. We have already underlined that these systems are the most powerful techniques developed for automated theorem proving. The source of the success of resolution lies mainly in its formal simplicity leading to straightforward implementations. Current versions of resolution are, in fact, not so simple but this is a price for increasing efficiency. We have also noted that the simplicity of resolution is of the kind that makes it artificial for humans. Natural deduction is the other extreme – a natural and flexible tool of deduction for humans but the rich machinery of rules and more complicated structure of proofs makes the implementation harder.

Is it possible to mix both approaches in order to get a system that enjoys advantages of either? We have shown that the richness of ND apparatus makes possible simulation of systems like tableau, even on the ground of quite standard basis. In order to simulate resolution and DP we only need one simple generalization; the admission of generalized clauses, instead of single formulae, as basic items in derivations. The resulting system, called RND (resolution based ND), is one of the possible solutions to the posed question. Moreover, it is also a positive answer to our problem of providing a system with really "flat" derivations.

4.4.1 Clauses Introduced

Clearly, we are more interested in such a form of resolution which does not require preprocessing step of transformation into normal form, but works directly on every formula. For our aim the most convenient treatment of resolution is that of Fitting [95]; moreover he applied it also to modal logics [94].

At first, we consider some minimal modification of KM system for **CPL**. Instead of single formulae we admit generalized clauses (i.e. finite sets of formulae) as U-lines in a derivation. So Γ, Δ denote arbitrary clauses, including empty one interpreted as $\bot$, whereas X, Y denote possibly empty sets of clauses. Recall the convention introduced in Chapter 1 to avoid ambiguity: in the description of the rules we use "," for separation of formulae in clauses, and ";" for separation of clauses. Thus we write Γ, Δ for concatenation of two clauses in one line, and $\Gamma; \Delta$ for two clauses from different lines of a derivation.

Clauses may appear in U-lines only, so proof construction rules are kept intact. The inference rules obtain the following, generalized form:

(Res)	$\Gamma, \varphi \ ; \ \Delta, -\varphi \ / \ \Gamma, \Delta$
(CNN)	$\Gamma, \neg\neg\varphi \ // \ \Gamma, \varphi$
$(C\alpha E)$	$\Gamma, \alpha \ / \ \Gamma, \alpha_i$, where $i \in \{1, 2\}$
$(C\alpha I)$	$\Gamma, \alpha_1 \ ; \ \Delta, \alpha_2 \ / \ \Gamma, \Delta, \alpha$
$(C\beta E)$	$\Gamma, \beta \ ; \ \Delta, -\beta_i \ / \ \Gamma, \Delta, \beta_j$, where $i \neq j \in \{1, 2\}$
$(C\beta I)$	$\Gamma, \beta_i \ / \ \Gamma, \beta$, where $i \in \{1,2\}$

It is obvious that this generalization is trivial if we do not add some rule which may introduce into derivations the sets containing more than one formula. After all, proof construction rules still operate on single formulae (or rather unit clauses) as S-lines and assumptions. The additional rule reveals clausal character of sets:

(C) $\Gamma, \beta \ / \ \Gamma, \beta_1, \beta_2$

It is straightforward that in the context of clausal ND, (Res) is just a generalization of $(\bot I)$. One may also keep $(\bot I)$ as a primitive rule and introduce (Res) as admissible rule by simple inductive proof on the cardinality of $\Gamma \cup \Delta$. Two other derivable rules which simplify derivations are:

$(C\beta I')$	$\Gamma, \beta_1, \beta_2 \ / \ \Gamma, \beta$
(W)	$\Gamma \ / \ \Gamma, \varphi$

The justification of the former is a consequence of the fact that we are using sets of formulae, so contraction is implicit. $(C\beta I)$ applied twice to β_1 and to β_2 yields $\Gamma, \beta, \beta = \Gamma, \beta$. In what follows we will not distinguish the applications of both forms of β introduction. Note that to prove (W) (weakening), you may always take any element of Γ as β_1 and apply $(C\beta I)$ with φ as β_2, then by (C) obtain Γ, φ. In case of empty Γ recall that $\Gamma{=}\bot$, so we may deduce any formula from it.

What do we achieve by this simple enrichment of ND? Here is an example – a proof of one of the distribution laws:

1	SHØW: $p \vee (q \wedge r) \rightarrow (p \vee q) \wedge (p \vee r)$		$[8, COND]$
2		$p \vee (q \wedge r)$	*ass.*
3		$p, q \wedge r$	$(2, C)$
4		p, q	$(3, C\alpha E)$
5		p, r	$(3, C\alpha E)$
6		$p \vee q$	$(4, C\beta I')$
7		$p \vee r$	$(5, C\beta I')$
8		$(p \vee q) \wedge (p \vee r)$	$(6, 7, C\alpha I)$

One should compare this proof with ordinary ND-proof of the same thesis to see the difference in the length and complexity, measured by the number of subderivations. Moreover, we can just reverse this proof, change the numeration of lines, and we immediately obtain a proof of the converse implication. It is possible because (C) is the inverse rule of $(C\beta I')$ and $(C\alpha I)$ is the inverse of $(C\alpha E)$ (applied to both constituents), and no rule which is not invertible (like (Res), $(C\beta E)$ or (W)) was used in this proof.

It seems that the restriction of S-lines and assumptions to single formulae is artificial and unjustified in such a system. In fact, one may devise generalized forms of proof construction rules, as well:

$[CCOND]$ If $X; \neg(\vee\Gamma) \vdash \Delta$, then $X \vdash \Gamma, \Delta$
$[CRED]$ If $X; \neg(\vee\Gamma) \vdash \bot$, then $X \vdash \Gamma$

In both cases X is a set of all usable clauses in a derivation.

Neither rule has to be primitive since we can demonstrate their admissibility in the clausal ND system with standard $[COND]$ and $[RED]$. We do not do that because much simpler solution is at hand.

Remark 4.1 ND T-systems of a similar character may be found in Boričić [53] and Cellucci [65]. They are defined for the proof of normalization theorem so some diferences appear, e.g. the rules operate on sequences, not on sets, so explicit rules of contraction and permutation are necessary. Still, the inference rules are very similar to the stated above; the main difference concerns proof construction rules. The counterparts of $[COND]$ and $[RED]$ are:

$[MCOND]$ If $X; \varphi \vdash \Gamma, \psi$, then $X \vdash \Gamma, \varphi \rightarrow \psi$
$[MRED]$ If $X; \varphi \vdash \Gamma$, then $X \vdash \Gamma, \neg\varphi$ ♣

4.4.2 System RND

Closer analysis shows that the system introduced in the preceding subsection is unnecessarily complicated and redundant. Suitable calculus may be significantly simplified, in particular, by introducing just one proof construction rule which covers all cases of standard ND rules of this sort. This system, under the name RND, was presented first in [150]. In RND we need only one proof construction rule called *Subsumption*:

> $[SUB]$ If $X; -\varphi_1; \ldots; -\varphi_k \vdash \Delta$, then $X \vdash \Gamma$, where:
> Γ is nonempty, $\Delta \subseteq \Gamma$, $\{\varphi_1, ..., \varphi_k\} \subseteq \Gamma$, $k \geq 0$.

Informally, the above rule reads: if in a subderivation, with unit clauses $-\varphi_i, k \geq i \geq 0$ as additional assumptions, a clause Δ is deduced, then we can close this subderivation and deduce Γ (a superset of Δ) on the basis of a set of clauses X alone. Note that all assumptions (if any) are complements of elements of Γ and that Δ empty ($= \bot$) is admitted.

Schematically the application of $[SUB]$ in KM looks like this:

$$\begin{array}{lll} & X & \\ i & \text{SHØW: } \Gamma & \\ i+1 & & \begin{array}{|c|}\hline -\varphi_1 \\ \vdots \\ -\varphi_k \\ \vdots \\ \Delta \\ \hline \end{array} \\ \vdots & & \\ i+k & & \\ \vdots & & \\ n & & \end{array}$$

The presence of many proof construction rules in ordinary ND-systems rather ensures a lot of flexibility in constructing derivations but complicates the system-description, metalogical proofs of system features, definitions of proof-search procedures, and many other things. Hence, the fact that in RND only one rule of this kind is sufficient is very important. Clearly, we must demonstrate that $[SUB]$ is general enough to cover all other ND proof construction rules.

Lemma 4.7 *Proof construction rules of KM are admissible in RND*

Proof Recall that in the original KM we have 3 such rules: $[DIR], [RED]$ and $[COND]$. The first is just a special case of $[SUB]$ with unit clause $\Gamma{=}\Delta$ and $k = 0$ Similarly, $[RED]$ is a special form of $[SUB]$ with Γ being unit clause, $k = 1$ and $\Delta = \bot$. $[COND]$ is an admissible rule as the following schema shows:

	X	
k	SHØW: β	$[n+1, SUB]$
$k+1$	SHØW: β_i, β_j	$[n, SUB]$
$k+2$	$-\beta_i$	*ass.*
	$\vdots$	
n	β_j	$(k+2, \ldots)$
$n+1$	β	$(k+1, C\beta I')$

β_j in line n was deduced from assumption $-\beta_i$ and possibly some elements of X with the help of the same deduction which justifies given application of $[COND]$. So every such an application may be eliminated at the expense of two additional lines ($k+1$ and $n+1$) and increasing the depth of original proof by 1 (two subproofs closed by $[SUB]$ instead of one closed by $[COND]$). ■

As we can see, ND system operating on clauses may be defined in a more concise and elegant way, where clauses and resolution are not ad hoc additions but provide an essential basis of a new system RND. Standard version of ND is just a particular case of RND. The system for **CPL** is obtained by addition to $[SUB]$ the following inference rules:

(W)	$\Gamma \;/\; \Gamma, \varphi$
(Res')	$\Gamma, \varphi \;;\; \Gamma, -\varphi \;/\; \Gamma$
(CNN)	$\Gamma, \neg\neg\varphi \;//\; \Gamma, \varphi$
$(C\alpha)$	$\Gamma, \alpha \;//\; \Gamma, \alpha_1 \;;\; \Gamma, \alpha_2$
$(C\beta)$	$\Gamma, \beta \;//\; \Gamma, \beta_1, \beta_2$

We may define formally a derivation of a clause in RND similarly as we did for ND in Chapter 2; we leave it to the reader[13] and define only a proof.

Definition 4.5 Let $\mathcal{D}$ be a derivation of a clause Π; if $\mathrm{S}(\mathcal{D}) = \varnothing$, then $\mathcal{D}$ is closed (its $\mathrm{U}(\mathcal{D}) = \{\Pi\}$), otherwise $\mathcal{D}$ jest open (both $\mathrm{U}(\mathcal{D})$ and $\mathrm{S}(\mathcal{D})$ are nonempty). Closed derivation of Π is a proof of Π (RDN $\vdash \Pi$).

In RND all previously stated rules (Res), $(C\alpha I)$, $(C\beta I)$ and $(C\beta E)$ are easily derivable with the help of (W). Since then, we will be using them freely in order to shorten derivations.

[13] One may find it also in [150].

It is rather easy to check that RND is sound for **CPL**:

Theorem 4.2 Soundness*: RND* $\vdash -\Gamma, \varphi$ *implies* $\Gamma \models \varphi$

PROOF Soundness of the inference rules is easy to provide if we interpret clauses as n-ary disjunctions. It remains to show that $[SUB]$ is normality preserving. Assume that $\vee\Delta$ follows from X and, possibly empty, set of assumptions: $-\varphi_1$, ..., $-\varphi_k$, then, clearly, so does $\vee\Gamma$, since $\Delta \subseteq \Gamma$. Hence, if $k = 0$, we are done, otherwise X implies $-\varphi_1 \rightarrow (-\varphi_2 \rightarrow ...(-\varphi_k \rightarrow \vee\Gamma)...)$, which is equivalent to $\varphi_1 \vee (\varphi_2 \vee ...(\varphi_k \vee (\vee\Gamma)...)$ which is equivalent to $\vee\Gamma$ since for each $i \leq k, \varphi_i \in \Gamma$. Therefore, $\vee\Gamma$ follows from X. ■

The form of the soundness theorem shows how to construct derivations justifying validity of arguments. We just start a proof of a clause which consists of a conclusion and complements of all premises. Here is an example of such a proof in RND:

1	SHØW: $\neg(p \rightarrow (q \rightarrow r)), p \rightarrow r, p \wedge \neg q$	$[10, SUB]$
2	$p \rightarrow (q \rightarrow r)$	*ass.*
3	$\neg(p \rightarrow r)$	*ass.*
4	$\neg p, q \rightarrow r$	$(2, C\beta)$
5	$\neg p, \neg q, r$	$(4, C\beta)$
6	p	$(3, C\alpha)$
7	$\neg r$	$(3, C\alpha)$
8	$\neg q, r$	$(5, 6, Res)$
9	$\neg q$	$(7, 8, Res)$
10	$p \wedge \neg q$	$(6, 9, C\alpha)$

Careful reader may try to make a proof using all three assumptions and getting $\perp$, or using two other possible pairs of assumptions (one is not enough). The above proof demonstrates that $p \rightarrow (q \rightarrow r), \neg(p \rightarrow r) \vdash p \wedge \neg q$.

Completeness of RND may be stated on the basis of lemma 4.7. and simple observation that all inference rules of ordinary KM are particular cases of RND (primitive or derivable) rules with $\Gamma = \Delta = \varnothing$. So everything derivable in KM-**CPL** is derivable in RND and we can state:

Theorem 4.3 *RND-system is strongly complete with respect to* **CPL**

Although purists may doubt, RND is essentially ND-system, at least in the sense that all proof resources of standard ND-system are present in it.

Moreover, RND is ND-system of a very simple structure with only one proof construction rule that covers all standard rules as special instances. Even the modest practice shows that proofs in this system may be simpler than in any standard ND. In particular, we are not forced to procure as many subderivations as in ordinary ND-systems, due to β-rule. In fact, in case we make an indirect derivation (we write down all possible assumptions) for a thesis we do not need more show-lines than (the starting) one. This is a consequence of the fact that in RND we can directly simulate standard resolution system (see the next subsection). Hence, we can always build proofs of the depth=1, which was impossible in other versions of ND examined so far. What is more, very often, even if we do not enter all possible assumptions, we can avoid opening subderivations. Our first example was a good illustration of the point. It should be underlined that in RND we can often obtain direct proofs, and with no subderivations, of theses that are normally proved in ND only by indirect proofs. Try, for example, to prove Peirce law in RND.

4.4.3 Simulation of Resolution and DP in RND

It seems that we can use RND as a handy tool for proof search, and we can do that in different ways. It is a consequence of the fact that RND is general enough to simulate proof techniques from many known systems which easily yield working decision procedures for **CPL**. It is not evident by inspection of the system; some rules are certainly not analytic in any sense, so the upper bound on the number of possible choices in proof search is not limited in advance. In particular, (W) is highly indeterministic. But we can easily obtain several decision procedures by imposing some restrictions on full RND as we did in case of standard ND, when producing its analytic versions. Another question is a definition of specific algorithms of proof search for RND. For the time being we confine ourselves only to showing that the system is able to simulate in a step-wise manner not only proofs but all kinds of deductive tasks, including model-extraction, which are realizable in ordinary resolution and Davis/Putnam procedure.

First of all, we can simulate standard resolution derivations by stipulating that we always write down all possible assumptions (i.e. we try indirect proof), then we apply all elimination rules (only one direction of $(CNN), (C\alpha)$ and $(C\beta)$) until we get atomic clauses. This stage corresponds to obtaining CNF of the input. Then we apply (Res) until ordinary termination conditions for resolution are satisfied. Note that due to this we do not need to apply (W) and building-up rules at all, and our derivation obeys subformula property. Moreover, we do not need to introduce more S-lines then the first one, and every proof thus obtained is of the depth=1.

Obviously, we are not forced to realize the simplest, and not very efficient, strategy since we are not forced to make a preprocessing step that corresponds to producing normal forms. We may freely intermingle the application of (Res) with other rules if it may shorten a proof. Also, to optimize the proof search in RND we can apply several known strategies characteristic for resolution based provers (consult e.g. [68, 101, 179]).

Making a derivation for a nonderivable clause opens a lot of interesting options concerning termination and building falsifying models, not necessarily by using strategies taken from resolution. We can, for instance, easily produce open derivations, where the set of U-clauses will be downward saturated, by simulation of KE. Suppose that we have reached a stage where, having applied all possible elimination rules, we obtained a set of atomic clauses X containing n different literals. It is enough to introduce every literal from this set and its complement as S-line and an assumption. Then applying (Res) and [SUB] sufficiently often we must obtain n unit clauses for each literal from X and this open derivation provides downward saturated set.

The strategy of saturation may be significantly improved by simulation of Davis/Putnam procedure which is one of the most efficient for **CPL** (cf. Chapter 3 or [77], see especially [95] for clear presentation). Basically it is performed by marking as used all clauses that cannot help to derive $\bot$. Recall that they are clauses containing tautologies (a rule of tautology), containing pure literals (no occurrence of complement literal in other clauses – pure literal rule), and being supersets of other clauses (subsumption rule). The splitting rule is simulated by introducing unit clause with suitable literal φ as a show-line and its complement as indirect assumption. First, we perform (Res') using our assumption as one of the premises and clauses contained above the last Show-line as the second premises. All the time we are marking all superset-clauses as used. If this subderivation is completed, we put it in a box, cancel SHOW in front of our chosen literal and repeat this procedure, now using φ as one of the premises for applications of (Res'). Otherwise, we will start the next Show-line with the next literal and its complement as an assumption and repeat the procedure. Unit literal rule is simply a special form of (Res') with one premise being unit atomic clause.

The above remarks show informally that RND may be treated as a simple frame suitable for direct simulation of several systems and strategies –

despite their apparent differences. We can ask which properties of RND are responsible for its flexibility. Why can RND simulate and combine proof search procedures from so different systems like resolution and tableau based systems like KE? It seems that the important reason for this flexibility is the fact that RND applies cut in its full generality, whereas other systems often use only one form. Let us explain what we mean by that. In Chapter 1 we have distinguished two forms of cut: progressive and regressive. First, note that resolution is a special case of progressive application of cut.

$$(Res) \quad \frac{\Gamma, \varphi \qquad - \varphi, \Delta}{\Gamma, \Delta}$$

Such a form of cut enables direct elimination of complementary formulae. On the other hand, in KE and tableau systems we deal with regressive application of cut. In Chapter 3 we have defined its form suitable for Hintikka-style systems; we recall it for the reader's convenience:

$$(R\text{-}Cut) \qquad \frac{\Gamma}{\Gamma, \varphi \mid \Gamma, -\varphi}$$

Although (R-Cut) in general is rather destructive for proof search since it introduces the indeterminacy, its analytic applications are useful. It adds complementary formulae in order to get some reductions in branches. So these systems exploit two different sides of the very same mechanism. The special feature of RND is the application of both sides in one system. Cut is used progressively by means of (Res) and regressively by [SUB] applied for indirect proof. It allows exploiting the full strength of cut in a proof and explains why such different proof systems like resolution and KE are simulated with the same ease. In fact, Davis/Putnam procedure is close to RND in this respect but the use of both forms of cut is unnecessarily restricted there – unit literal rule is a restricted form of resolution and splitting rule, a restricted form of regressive cut. In RND we have simple frame with unrestricted application of both forms of cut.

Let us underline that simulation of neither system uses (W) or introduction rules. Moreover, [SUB] is used only once for simulation of resolution, just in the beginning. In case of simulation of DP derivations [SUB] may be used many times (exactly because of restricted forms of cut in DP) but every application is analytic since S-line contains only subformulae of clauses already present. It shows that we can provide restricted but complete versions of RND which are analytic. The version sufficient for simulation of DP is a clausal counterpart of AND1, so we may call it ARND1. The version sufficient for simulation of resolution is even weaker since it requires only one application of $[SUB]$; we may call it ARND0. Finally, a comparison with AND2, shows that full elimination of (W) and introduction rules is not necessary for analyticity. In this way, by restricting application of these rules, we can obtain one more version – ARND2 which is a counterpart of AND2. More strictly:

1. The most restrictive ARND0 calls for only one S-line (the first one) and only elimination rules and (Res) are used but all assumptions (i.e. complements of elements of a proved clause) must be introduced. The completeness of ARND0 follows from the completeness of standard resolution system for **CPL**.

2. More liberal ARDN1 admits the introduction of other S-lines but with the analytic restriction. We may enter SHOW:Γ provided $\Gamma \subseteq \overline{\overline{\Pi}}$, where Π is a proved clause and $\overline{\overline{\Pi}}$ is the set of all subformulae of Π closed under addition of single negation.

3. ARDN2, in addition to ARND1 admits applications of weakening and introduction rules but with the same analytic restriction, i.e. any φ added by (W), or $\neg\neg\varphi, \alpha, \beta$ added by the respective introduction rules must be in $\overline{\overline{\Pi}}$.

Let us note that: every ANDi, $i \in \{1, 2\}$ derivation is a particular case of ARNDi derivation, exactly as in case of relationship between ND and RND. But ARND0 is not a generalization of any analytical version of ND since a version with no subproofs and without clauses and $(C\beta)$ would be incomplete.

Finally, note that RND, despite its strong dependence on cut, is by no means restricted to the simulation of only such systems that make use of this rule. In Section 7.2 we have shown that AND1 may simulate tableau trees, so it is obvious that ARND1 can make it too. But to facilitate direct step-wise simulation we may introduce admissible proof construction rule $[SEP]$ (from separation):

$[SEP]$ If $X; \Gamma; \Delta_1 \vdash \bot$, then $X; \Gamma \vdash \Delta_2$, where $\Gamma = \Delta_1 \cup \Delta_2$

It is a straightforward clausal generalization of the rule $[\beta]$ which was considered in Section 4.2 in the context of simulation of TS by ADN1. The proof of its admissibility is simple and left to the reader.

4.4.4 RND for First-Order Logic

RND may be easily extended to cover first-order logics in classical and free version. In case of classical logic we can extend RND in the same way as is usually done for resolution, supplying some general form of resolution rule involving unification and factoring. Below we will just generalize the original KM rules. There are three new rules of inference:

$(C\forall E)$ $\Gamma, \forall x\varphi \ / \ \Gamma, \varphi[x/\tau]$
$(C\exists I)$ $\Gamma, \varphi[x/\tau] \ / \ \Gamma, \exists x\varphi$
$(C\exists E)$ $\Gamma, \exists x\varphi \ / \ \Gamma, \varphi[x/y]$, where y is new

In order to obtain a rule for introduction of $\forall$ we only need to generalize slightly $[SUB]$. Now we require in side conditions that $\Delta \subseteq_\forall \Gamma$, where by $\subseteq_\forall$ we mean that for any $\varphi \in \Delta$, either $\varphi \in \Gamma$ or $\forall x\varphi \in \Gamma$ and x is not free in X and Γ. This form will be called $[SUB']$.

It is straightforward to check that everything provable in KM for **CQL** is provable with the help of these rules; so RND-**CQL** is complete. In order to prove soundness we must proceed indirectly as in Chapter 2, by providing RND version of KMG, showing its soundness and simulation of KM by KMG. We leave details to the reader as well as defining RND rules being counterparts of other ND formalizations of **CQL**.

Remark 4.2 Essentially the same rules for $\forall$ elimination and $\exists$ introduction are in Boričić [53] and Cellucci [65] ND systems. For $\forall$ introduction they apply inference rule with usual restrictions. For $\exists$ elimination Boričić applies proof construction rule in Gentzen style, whereas Cellucci has inference rule but introducing Hilbert's epsilon term. ♣

Similarly, we may enrich RND with clausal versions of quantifier rules for free logic. The new inference rules go as follows:

$(CF\forall E)$ $\Gamma, \forall x\varphi \ ; \Gamma, E\tau \ / \ \Gamma, \varphi[x/\tau]$
$(CF\exists I)$ $\Gamma, \varphi[x/\tau] \ ; \Gamma, E\tau \ / \ \Gamma, \exists x\varphi$
$(CF\exists E)$ $\Gamma, \exists x\varphi \ / \ \Gamma, \varphi[x/y] \wedge Ey$, where y is new

Once again we must slightly change $[SUB]$:

$[SUB'']$ If $X \ ; \psi_1 \ ; \ ..... \ ; \psi_k \ ; -\varphi_1 \ ; \ ..... \ ; -\varphi_i \ \vdash \ \Delta$, then $X \ \vdash \ \Gamma$,
where: Γ is nonempty, $\Delta \subseteq_\forall \Gamma$, $\{\varphi_1, ..., \varphi_i\} \subseteq \Gamma$, $i \geq 0$, $k \geq 0$ and for any $j \leq k$, $\psi_j = Ex$ and $\forall x\varphi \in \Gamma$.

Informally, $[SUB'']$ admits an existential assumption (Ex) for any generally quantified formula in Γ keeping all other requirements as in the version for classical first-order logic. Again, completeness of RND-**FQL** follows from completeness of KM', whereas soundness proof calls for the introduction of RND version of KMG'. One should try to do it according to the lines of proof described in Chapter 2.

Chapter 5

Survey of Modal Logics

This Chapter provides a necessary background for studying applications of ND in modal logics. It is a collection of basic facts needed for understanding of the remaining chapters. Section 5.1. introduces propositional languages of multimodal logics and establishes notational conventions. After a presentation of general taxonomy of modal logics in Section 5.2. we characterize them axiomatically in the next section. Section 5.4. introduces relational semantics for different families of modal logics. Except standard Krikpe's semantics it contains the basics of neighborhood semantics for weak modal logics. Some attention is paid to correspondence theory and some general schemata investigated later. After short section on completeness and decidability matters we finally present various kinds of first-order modal logic in Section 5.6.

5.1 Basic Modal and Tense Language

Before we start with the presentation of ND-systems for modal logics we recall the most basic and the most important (for our interests) facts concerning standard modal languages and logics. Most of the information from this chapter is introduced just to fix a notation and to keep the book self-sufficient. The reader who needs deeper knowledge of the subject should consult some textbooks, e.g. [35, 67, 112].

Let $\mathbf{L_M}$ denote standard (mono)modal propositional language i.e. an abstract algebra of formulae

$$\langle FOR, \neg, \wedge, \vee, \rightarrow, \Box, \Diamond \rangle \tag{5.1}$$

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*, Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_5,

with denumerable set of propositional symbols.

$$PROP = \{\ p, q, r, \ldots, p_1, q_1, \ldots\} \subseteq FOR \tag{5.2}$$

Except standard boolean functors there is an additional pair of modals: $\Box, \Diamond$. Usually these symbols are used to denote unary modal operators of *necessity* and *possibility*. It is an alethic interpretation, but of course many other interpretations of temporal, epistemic or deontic character, may be provided. Despite the possible interpretation, syntactically both functors behave like negation, i.e.

- if $\varphi \in FOR$, then $\odot\varphi \in FOR$, where $\odot \in \{\Box, \Diamond\}$

Sometimes it is convenient to use a language with only one modal functor. Languages of this kind will be denoted as $\mathbf{L}_\Box, \mathbf{L}_\Diamond$. In such cases the other functor will be understood as a definitional shortcut:

Definition 5.1 **($\Box, \Diamond$)** $\Box\varphi := \neg\Diamond\neg\varphi$; $\Diamond\varphi := \neg\Box\neg\varphi$

Languages (and logics) with one pair of modalities are commonly called monomodal. But for many purposes they are not sufficient and the generalization to multimodal languages with many modalities is quite natural. For example, we may need to have alethic and deontic modalities together, or to represent a knowledge of many agents in one system. In such cases we must apply indices to distinguish different modalities, e.g. $\Box_a, \Diamond_b$ or $[a], \langle b\rangle$. The latter notation is particularly useful when indices of modalities have complex form, i.e., if we admit combined modalities. In case we want just to point out that a logic has n distinct modalities we will usually use a notation $\Box_i, \Diamond_i,\ (n \geq i \geq 1)$ in contrast to $\Box^n, \Diamond^n$, which means that suitable modal constant is put n-times before some formula. Indices of modal operators are variables running on different sets, e.g. in case of epistemic logics they denote different agents.[1] Multimodal languages with n (pairs of) modalities will be denoted respectively as: $\mathbf{L_{M_n}}, \mathbf{L_{\Box_n}}, \mathbf{L_{\Diamond_n}}$.

In particular, we will be interested in temporal logic built in the language $\mathbf{L_T}$, the bimodal variant of $\mathbf{L_M}$ with interactive Priorean operators $\Box_F, \Diamond_F, \Box_P, \Diamond_P$ designed for dealing with temporal interpretation of modalities. We would rather use traditional symbols G, F, H, P for these operators which are in common use. They are interpreted respectively as:

[1] In such case we read a formula $\Box_a\varphi$ – a knows that φ holds.

G for "always in the future" (instead of $\Box_F$ or $[F]$)
F for "sometimes in the future" (instead of $\Diamond_F$ or $\langle F \rangle$)
H for "always in the past" (instead of $\Box_P$ or $[P]$)
P for "sometimes in the past" (instead of $\Diamond_P$ or $\langle P \rangle$)

From the technical point of view $\mathbf{L_T}$ is very important modal language since except ordinary (forward-looking) modalities (G corresponding to $\Box$ and F to $\Diamond$) it has a pair of backward-looking operators H and P. It yields an extra expressive power. In Chapter 11 we will consider even more expressive functors $\mathcal{A}$ and $\mathcal{E}$, called global modalities, and a difference modality $\mathcal{D}$.

Generally, formulae with modal operator or its negation as the main functor will be called *modal formulae*, or shortly, *m-formulae*. For convenience we will use in $\mathbf{L_{M_n}}$ the generalization of compact notation from [93], where π^i, or ν^i denote m-formula with functor of i-modality ($1 \leq i \leq n$), and π or ν denote arguments of this functor, in accordance with the following table:

π^i	ν^i	π and ν
$\Diamond_i\varphi$	$\Box_i\varphi$	φ
$\neg\Box_i\varphi$	$\neg\Diamond_i\varphi$	$\neg\varphi$

In particular, for bimodal temporal language $\mathbf{L_T}$ we apply the following notation:

π^F	π^P	ν^F	ν^P	π and ν
$F\varphi$	$P\varphi$	$G\varphi$	$H\varphi$	φ
$\neg G\varphi$	$\neg H\varphi$	$\neg F\varphi$	$\neg P\varphi$	$\neg\varphi$

In case of monomodal language π^i means simply $\Diamond\varphi$ or $\neg\Box\varphi$. Following this convention we will often divide m-formulae on π- or ν-formulae (or $\Box$- and $\Diamond$-formulae, if we want to exclude cases with negated modality in front of a formula). Note that the complement of every π-formula is always ν-formula and conversely.

$\Box\Gamma, \Diamond\Gamma$ will denote sets of formulae obtained from the set Γ by adding $\Box$ or $\Diamond$ to every formula. Sets of formulae (clauses) consisting of only π- (ν-) formulae will be called π- (ν-)sets (clauses) and denoted by Π^i (Υ^i), possibly with subscripts. Every occurence of a symbol Π_k (or Υ_k) will denote the set of formulae obtained from π-set Π^i_k (Υ^i_k) by deleting all modal functors of suitable sort; e.g. let $\Upsilon^P_3 = \{Hp, \neg Pq, H(p \to r), \neg P\neg r\}$, then $\Upsilon_3 = \{p, \neg q, p \to r, \neg\neg r\}$.

5.2 Modal Logics in General

We recall rather standard characterization of the most important modal logics, based on [69].[2] Every modal logic is understood here as a set of formulae containing all tautologies of **CPL** and closed under some operations (rules). Formally:

Definition 5.2 (Modal Logic) Modal logic **L** is any set of formulae in some modal language $\mathbf{L_{M_n}}$ which satisfies the following conditions:

- TAUT $\subseteq$ **L**, where TAUT denotes the set of tautologies of **CPL**
- if $\varphi \in \mathbf{L}$, then $\mathrm{e}(\varphi) \in \mathbf{L}$, where e is any endomorphism (substitution) from $PROP$ into FOR
- if $\varphi \in \mathbf{L}$ and $\varphi \to \psi \in \mathbf{L}$, then $\psi \in \mathbf{L}$

It is a very general characteristics of modal logic and hence of little interest since no particular modal law or rule is specified. We have only substitutions of classical tautologies in the richer language. In order to obtain more interesting cases we must consider some specific modal formulae or rules. For simplicity we formulate the basic classification for monomodal logics in the language $\mathbf{L}_\Box$. Usually the main classes of modal logics are characterized with the help of the following conditions/formulae.

(RE)	if $\varphi \leftrightarrow \psi \in \mathbf{L}$, then $\Box\varphi \leftrightarrow \Box\psi \in \mathbf{L}$
(RM)	if $\varphi \to \psi \in \mathbf{L}$, then $\Box\varphi \to \Box\psi \in \mathbf{L}$
(RC)	if $\varphi \wedge \psi \to \chi \in \mathbf{L}$, then $\Box\varphi \wedge \Box\psi \to \Box\chi \in \mathbf{L}$
(RR)	if $\wedge\Gamma \to \psi \in \mathbf{L}$, then $\wedge\Box\Gamma \to \Box\psi \in \mathbf{L}$, where $\Gamma \neq \varnothing$
(RG)	if $\varphi \in \mathbf{L}$, then $\Box\varphi \in \mathbf{L}$
M	$\Box(\varphi \wedge \psi) \to \Box\varphi \wedge \Box\psi$
C	$\Box\varphi \wedge \Box\psi \to \Box(\varphi \wedge \psi)$
K	$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$
N	$\Box\top$

With the help of these specific modal principles one can distinguish the following classes of modal logics being extensions of **CPL**.

- Every **L** closed under (RE) is congruent (also called classical or equivalential)

[2] Cf. also [59] and [208].

- Every **L** closed under (RM) (or every congruent **L** containing M) is monotonic
- Every **L** closed under (RR) (or every monotonic **L** containing C) is regular
- Every regular **L** closed under (RG) (or containing N) is normal.[3]

Obviously, every normal logic is regular, every regular is monotonic, and monotonic is congruent, so we have a hierarchy of these classes. Since every class is closed under taking products, we have in every family the weakest logic . Let **E** denote the weakest congruent logic, **M** – the weakest monotonic, **R** – the weakest regular, and **K** – the weakest normal logic[4] Of course this is not the whole story since we can find some modal logics not falling into any of the above category, like famous Lewis' systems **S1, S2, S3**. Generally, every logic containing **E** but not closed under (RE) is called quasi-congruent, containing **M** but not closed under (RM) – quasi-monotonic e.t.c.

The above systematization of modal logics may be generalized to multimodal logics but usually the papers and books dealing with multimodal logics, like e.g. [166, 100], are concerned rather with the investigation on normal logics.[5] No doubts, normal modal logics is the most popular and investigated class of modal logics and many fundamental results are stated only for them. This is the reason that in many textbooks they are the only modal logics taken under consideration. It may be justified theoretically by the fact that other classes of logics may be simulated by bimodal normal logics.[6] But this line of thought may be applied equally well to multimodal normal logics since they are simulated by monomodal ones (Thomason's results from 1970s); and to modal logics in general, since they may be reduced to fragments of first- and second-order logic – cf. Section 5.4.3. In

[3]Usually elements of this class are defined in a bit different (but equivalent) way as modal logics containing K and closed under (RG).

[4]We follow Chellas [69] in naming conventions (although he used the name classical instead of congruent logics). Often different names are applied, particularly for regular logics, e.g. instead of **R**, **C** is used by Segerberg [246] and Fitting [93], and **C2** by Lemmon [173].

[5]The exception is Gasquet [106], where multimodal regular and monotonic logics are dealt with.

[6]This result was proved by Gasquet and Herzig [107] and improved by Kracht and Wolter [167] and by Hansen [124].

practice it is better to work directly with weak logics in these area where they apply well[7] and use simulation rather for transfer of results.

Multimodal logics may be *homogenous* or *heterogenous*. The former have modalities of the same sort (characterized by the same properties), the latter combine different kinds of modalities. In both cases a logic may be just a simple fusion of a few systems with no interaction of distinct modalities. If different modalities are independent of themselves (no interaction), the results which hold for monomodal logics are straightforward to extend to multimodal case. The situation is more interesting (and more difficult) for multimodal logics with interactive modalities since their expressive power is usually considerably stronger. The external sign of such a situation is the presence of interactive principles like this simple instance of *inclusion axiom*:

$$\Box_a\varphi \to \Box_b\varphi \tag{5.3}$$

It means, under epistemic interpretation, that the knowledge of agent b contains all the knowledge of agent a. The best, up-today exposition of problems connected with combining logics may be found in [100].

In the group of interactive multimodal logics one of the most interesting case is represented by bimodal temporal logics. In this book we limit our interests to Priorean logics of instant time in the language $\mathbf{L_T}$, often called tense logics in contrast to temporal logics applied in computer science, like **PTL**.[8] In the latter case rather different constants are often used as primitive like e.g.: $\bigcirc$ (next moment), $\mathcal{U}$ (until), $\mathcal{S}$ (since), moreover, their semantics is slightly different than standard relational semantics applied for Priorean logics.

Priorean temporal logics are normal i.e. both (pairs) of modalities satisfy (RR) and (RG) (as well as C, K and N). Moreover, they must contain interactive principle $(B\text{-}Te)$ expressing the symmetry of past and future:

$$\varphi \to \Box_i\Diamond_j\varphi, \ where\ i \neq j \in \{F, P\} \tag{5.4}$$

The weakest temporal logic satisfying these conditions is called **Kt**. Note however that from the practical point of view some stronger logics are treated as basic since we want to express at least such properties like transitivity of time flow.

---

[7] Hansen [124] gives a couple of examples concerning interesting applications of monotonic logics.

[8] Cf. e.g. [177, 116].

We will be interested not only in normal logics but also in some congruent, monotonic and regular ones. The first two classes will be called weak modal logics, in contrast to strong modal logics including normal and regular ones.

Weak modal logics are not very popular, many good books on modal logic do not even mention them, as we have already noted. The investigations on proof methods and decision procedures for such logics are also rather modest. Although these remarks partly apply to regular logics as well, there are good reasons to keep them together rather with normal logics. The important rationale for such a division lies in the applicability of systems. It is well known that epistemic or doxastic interpretation of modal constants leads to unintuitive results in the context of normal or regular logics. This is the main reason that weaker logics are often considered as better candidates in this area.

There is also a good semantical criterion for treating congruent and monotonic logics as a separate category. Most normal and regular modal logics are characterizable by Kripke frames (cf. Section 5.4), but for congruent and monotonic systems this approach fails. Because of that semantic uniformity, Fitting's fundamental work [93] on proof methods for modal logics covers also many regular (and quasi-regular) ones, but no weaker logic, except **M** (called **U** in [93]). Fortunately, congruent and monotonic logics are also characterizable in terms of some possible world-based semantics, namely by neighbourhood (called also minimal – cf. [69]) frames. This is more general kind of semantics and perhaps not so handy in use although intuitively as natural as Kripke semantics.

5.3 Axiomatic Approach to Modal Logics

The earliest and still the most popular syntactic style of defining modal logics was axiomatic (or Hilbert). We recall here well known axiomatizations of the four weakest monomodal logics for easy reference.

(a) Hilbert formalization of the weakest congruent modal logic **E** denoted by H-**E** consists of:

1. Axioms of **CPL** (cf. Section 1.1, but any complete set is suitable)
2. Axioms of **E**:

Pos $\Diamond p \leftrightarrow \neg\Box\neg p$

3. Rules:

(MP) $\vdash_E \varphi \to \psi, \vdash_E \varphi \ / \ \vdash_E \psi$
(RE) $\vdash_E \varphi \leftrightarrow \psi \ / \ \vdash_E \Box\varphi \leftrightarrow \Box\psi$
(SUB) $\vdash_E \varphi \ / \ \vdash_E e(\varphi)$,
where $e : PROP \longrightarrow FOR$

(b) Hilbert formalization of the weakest monotonic modal logic **M** denoted by H-**M** consists of:

1. Axioms of **CPL**

2. Axioms of **M** (only *Pos* like in H-**E**)

3. Rules – (MP), (SUB) and:

(RM) $\vdash_M \varphi \to \psi \ / \ \vdash_M \Box\varphi \to \Box\psi$

(c) Hilbert formalization of the weakest regular modal logic **R** denoted by H-**R** consists of:

1. Axioms of **CPL**

2. Axioms of **R** – *Pos* and:

K $\Box(p \to q) \to (\Box p \to \Box q)$

3. Rules – like in H-**M**

(d) Hilbert formalization of the weakest normal modal logic **K** denoted by H-**K** consists of:

1. Axioms of **CPL**

2. Axioms of **K** – like in H-**R**

3. Rules – (MP), (SUB) and:

(RG) $\vdash_K \varphi \ / \ \vdash_K \Box\varphi$

$\vdash_L \varphi$ means of course that φ is a thesis of respective system H-**L** i.e. has a proof in H-**L** being a sequence of formulae deduced from axioms by means of primitive rules. $\vdash_L \Gamma$ means that all formulae in Γ are theses of respective system. The set of all theses of **L** will be denoted by Th(**L**). $\nvdash_L \varphi$ means that φ is not a thesis of **L**.

It is evident that axiomatic characterization is closely related to abstract definition of normal modal logics as sets satisfying some closure conditions. So in what follows for simplicity we will identify modal logics with their stated axiomatizations and use notation $\vdash \varphi$, whenever **L** is established or unimportant.

Other important modal logics may be obtained by the addition of some extra axioms to our basic Hilbert formalizations. The following table displays schemata of the most popular axioms.[9]

Name	Axiom
D	$\Box\varphi \to \Diamond\varphi$
$\Box D$	$\Box(\Box\varphi \to \Diamond\varphi)$
DC	$\Diamond\varphi \to \Box\varphi$
T	$\Box\varphi \to \varphi$
$\Box T$	$\Box(\Box\varphi \to \varphi)$
4	$\Box\varphi \to \Box\Box\varphi$
$\Box 4$	$\Box(\Box\varphi \to \Box\Box\varphi)$
$4C$	$\Box\Box\varphi \to \Box\varphi$
B	$\varphi \to \Box\Diamond\varphi$
$\Box B$	$\Box(\varphi \to \Box\Diamond\varphi)$
5	$\Diamond\varphi \to \Box\Diamond\varphi$
2	$\Diamond\Box\varphi \to \Box\Diamond\varphi$
M	$\Box\Diamond\varphi \to \Diamond\Box\varphi$
3	$\Box(\Box\varphi \to \psi) \vee \Box(\Box\psi \to \varphi)$
L	$\Box(\Box\varphi \wedge \varphi \to \psi) \vee \Box(\Box\psi \wedge \psi \to \varphi)$
F	$\Box(\Box\varphi \to \psi) \vee (\Diamond\Box\psi \to \varphi)$
R	$\Diamond\Box\varphi \to (\varphi \to \Box\varphi)$
G	$\Box(\Box\varphi \to \varphi) \to \Box\varphi$
Grz	$\Box(\Box(\varphi \to \Box\varphi) \to \varphi) \to \varphi$
Go	$\Box(\Box(\varphi \to \Box\varphi) \to \varphi) \to \Box\varphi$

We will follow the convention of Lemmon in naming logics; if a logic is axiomatized by addition of axioms X, Y, Z, to say **K**, then we call it **KXYZ** (possibly with application of dots as punctuation marks if the natural number is the name of the axiom). The exception to this principle applies to some well known normal logics commonly called:

[9] The names of axioms – with little exceptions – come from [117].

D = **KD**
T = **KT**
B = **KTB**
S4 = **KT4**
S5 = **KT5**

Particularly important class of modal logics contains all systems built as combinations of axioms: D, T, B, 4 and 5 over **E, M, R, K**. We will call them *basic modal logics*. How many different logics of this kind do we find? Although in every class there is 32 ($= 2^5$) possible combinations there is not so many different logics because some sets of axioms yield the same logic. It is the result of the following deductive interrelations:

Lemma 5.1

$\mathbf{CPL} + T \vdash D$	$\mathbf{E} + T + 5 \vdash 5$	$\mathbf{M} + B + 4 \vdash 5$
$\mathbf{CPL} + T + 5 \vdash B$	$\mathbf{E} + B + 4 + D \vdash 4$	$\mathbf{M} + B + 5 \vdash 4$
$\mathbf{CPL} + D + 4 + B \vdash T$	$\mathbf{E} + B + T \vdash N$	$\mathbf{M} + B \vdash N$

In the light of this result e.g. **S5** = **KT45** = **KTB4** = **KT5**. In the first three cases we have used the form **CPL**+X because we want to stress that there is no use of (RE) or (RM) in the proof. As a result there is 18 distinct congruent logics, 15 monotonic logics, 12 regular logics and 15 normal logics axiomatized with $D, T, 4, B, 5$. To this number we can add also 16 congruent logics with (RG) and 10 monotonic logics with the same rule, since – due to the lack of K in these logics – they are not normal. In particular, the lattice of monotonic logics is isomorphic to the well known lattice of normal logics obtained by the combination of these axioms over **K**. Relatively smaller number of regular logics follows from the fact that any regular logic containing B is normal.

It is important to note that all axioms (and many others) characterizing particular basic logics are special instances of *Geach Axiom*:

$$\Diamond^m \Box^n \varphi \to \Box^s \Diamond^t \varphi \tag{5.5}$$

For example: 5 is the case with $m = s = t = 1$ and $n = 0$, whereas T has $n = 1$ and $m = s = t = 0$; axiom 2 is the special case of Geach axiom with all modalities having just one occurrence.

But not all axioms from the table fall under general schema of Geach Axiom. Among logics that are axiomatized by formulae of different shape we will pay special attention to normal logics axiomatized with the help of axiom 3 and L. They are called here linear logics because they serve to formalize several kinds of linear order on model domains (c.f. the next section) The weakest logic of this kind is called **K4.3** = **K4L**, and its minimal extensions are **K4D.3** = **K4DL** and **S4.3**. Both axioms are often replaced by some equivalents of the form:

$3'$ $\quad \Diamond\varphi \wedge \Diamond\psi \rightarrow \Diamond(\Diamond\varphi \wedge \psi) \vee \Diamond(\varphi \wedge \Diamond\psi)$
L' $\quad \Diamond\varphi \wedge \Diamond\psi \rightarrow \Diamond(\Diamond\varphi \wedge \psi) \vee \Diamond(\varphi \wedge \Diamond\psi) \vee \Diamond(\varphi \wedge \psi)$

For axiomatic characterization of **Kt** we must double K, Pos and (RG) in H-**K** by putting G (or H) instead of $\Box$ and F (or P) instead of $\Diamond$. We need also a pair of interactive axioms concerning interrelation between future and past (cf. Formula 5.4.):

GP $\quad p \rightarrow GPp$
HF $\quad p \rightarrow HFp$

All other temporal logics are obtained by the addition of some further axioms to H-**Kt**. Usually **Kt4** is treated as the basic logic in this family, because the addition of 4 formally express the transitivity of time flow. One should note that in case of temporal logics we have two counterparts of every axiom from the table, e.g. for 4 we have:

$4F$ $\quad G\varphi \rightarrow GG\varphi$ and
$4P$ $\quad H\varphi \rightarrow HH\varphi$

To obtain a formalization of **Kt4** it is enough to add only one of them to **Kt** since they are interderivable. But such interderivability is not a rule but rather an exception. When constructing strengthenings of **Kt** with the help of temporal counterparts of axioms from the table, one should remember that in many cases we can obtain independent variants for both modalities (heterogenous logics). For example, $DF - G\varphi \rightarrow F\varphi$ and $DP - H\varphi \rightarrow P\varphi$ are independent, so we can obtain three different extensions of **Kt**: homogenous **KtD** = **KtDF.DP** and two different heterogenous logics – **KtDF** and **KtDP**. In particular, this remark applies to linear temporal logics — if we want, e.g. **Kt4.3**, we must add to **Kt4** both temporal variants of L; an addition of only one of them yields the logics of tree-like structures. One should note that the richer language of temporal logics may express linearity with the help of more compact formulae:

LF $\quad H\varphi \wedge \varphi \wedge G\varphi \rightarrow GH\varphi$
LP $\quad H\varphi \wedge \varphi \wedge G\varphi \rightarrow HG\varphi$

Axioms $3F$ and $3P$ may be obtained from the above formulae just by deletion of φ from conjunction in antecedents of LF and LP.

In the field of multimodal logics we may also distinguish a class of logics axiomatized by instances of multimodal version of Geach axiom, called in Catach [64] $G^{a,b,c,d}$ or *a, b, c, d-incestuality axiom*:

$$\Diamond_a \Box_b \varphi \rightarrow \Box_c \Diamond_d \varphi \tag{5.6}$$

where a, b, c, d are indices of, not necessarily different, modalities. At first sight it may seem that this formula is not as general as Geach axiom, because only one occurrence of each modality is present. But one should note that any of a, b, c, d may represent a complex modality obtained by composition or union of simpler modalities. Thus $[a; b]$ denotes a composition of a-necessity and b-necessity, whereas $[a \cup b]$ denotes their union. These operations are defined as follows: $[a; b]\varphi := [a][b]\varphi$ and $[a \cup b]\varphi := [a]\varphi \wedge [b]\varphi$.[10] If we take under consideration also empty modality $[\varepsilon]$, then it is obvious that schema $G^{a,b,c,d}$ covers all possible instances of Geach axiom and many more. In particular, most of the interactive principles considered in literature are instances of this schema with some of a, b, c, d different. For example: both temporal axioms GP and HF are cases with $a = b = \varepsilon$ and $c = F, d = P$ or $c = P, d = F$, whereas simple inclusion axiom has $a = d = \varepsilon$.

All normal (i.e. satisfying K and (RG) for each modality) multimodal logics axiomatized by instances of $G^{a,b,c,d}$ only, will be called, after Catach, *incestual modal logics*. Clearly, if complex (and empty) modalities are admitted we must add also for each a, b, axioms of the form:

$[\varepsilon]\varphi \leftrightarrow \varphi$
$[a; b]\varphi \leftrightarrow [a][b]\varphi$
$[a \cup b]\varphi \leftrightarrow [a]\varphi \wedge [b]\varphi$

This wide class may be divided further on serial, symmetric, euclidean and other logics if we restrict admissible axioms to schemata with some of a, b, c, d being ε, e.g. axioms with $a = c = \varepsilon$ give us a class of serial logics, whereas with $a = d = \varepsilon$ give a class of grammar or inclusion logics. For more on these subdivisions and properties of distinct classes c.f. Baldoni [18, 19]. To appreciate the generality offered by the schema of a, b, c, d-incestuality axiom, note that it covers not only all instances of

[10] It may be of interest to consider also other operations on modalities. Resolution and labelled tableau systems for some logics of this kind are provided by De Nivelle, Schmidt and Hustadt [196, 242].

Geach axiom with different modalities, but also, due to the operation of union of modalities, implications with conjunctions as antecedents or consequents. For instance, one may easily observe that axioms LF and LP may be rewritten as $[P \cup \varepsilon \cup F]\varphi \rightarrow [F;P]\varphi$ and $[P \cup \varepsilon \cup F]\varphi \rightarrow [P;F]\varphi$ respectively. In result, linear temporal logics like **Kt4.3** also belong to the class of incestual logics. This is in contrast to monomodal case, where axioms like L or 3 do not fall under Geach axiom schema.

5.3.1 Deducibility

The relation of deducibility (provability) may be defined in two nonequivalent ways:

Definition 5.3 (Local ($\vdash$) and global ($\Vdash$) deducibility)

- $\Gamma \vdash_L \varphi$ iff $\vdash_L \psi_1 \wedge .... \wedge \psi_n \rightarrow \varphi$, where $\{\psi_1, ..., \psi_n\} \subseteq \Gamma$
- $\Gamma \Vdash_L \varphi$ iff there is a proof of φ in **L**, where formulae from Γ are also used as premises for the application of rules

On the ground of at least congruent modal logics $\Vdash$ is stronger than $\vdash$, because the closure under any of $(RE), (RM), (RR), (RG)$ means that for $\Vdash$ the *deduction theorem* in simple form does not hold.[11] For example in **K** we have $p \Vdash \Box p$ (due to the closure under (RG)), but $p \nvdash \Box p$ (since $\nvdash_K p \rightarrow \Box p$). We have only a dependance in one direction:

$$\text{if } \Gamma \ \vdash_L \varphi, \text{ then } \Gamma \ \Vdash_L \varphi$$

For $\vdash$, the deduction theorem is satisfied by definition; in particular one should note that:

$\Gamma \vdash_L \varphi$ iff φ is deduced from $\Gamma \cup Th(\mathbf{L})$ by only one rule (MP)

Different authors have different preferences concerning the importance of both relations. Only few of them treat both as equally important; in particular, Fitting [93] applies clever complex notation:

$$\Gamma \Vdash_L \Delta \vdash_L \varphi$$

[11] It does not mean that deduction theorem in weaker form does not hold for these logics. Many theorems of this kind were established by Perzanowski in [206, 207].

which means that φ has a proof in which formulae from Γ are global assumptions (axioms), and elements from Δ are local assumptions. A distinction between these two types of assumptions may be well understood e.g. in terms of interactive proof engines. In such a context elements of a knowledge base are global assumptions, whereas data provided by the user are local assumptions.

Because the rules of ND-systems and other calculi naturally generate relations of the type $\vdash$, in what follows we will be rather interested in the first (weaker) notion of deducibility, incidentally noting how to formalize $\Vdash$. In particular, we define Γ as **L**-inconsistent iff $\Gamma \vdash_L \bot$; otherwise Γ is **L**-consistent.

Primitive rules of axiomatic systems are theoretically sufficient, but in practice one can use many others in order to obtain shorter proofs. We divide secondary (or additional) rules on two groups[12]:

- $\Gamma \;/\; \varphi$ is **L**-derivable iff $\Gamma \vdash_L \varphi$

- $\Gamma \;/\; \varphi$ is **L**-admissible iff $\vdash_L \Gamma$ implies $\vdash_L \varphi$

Clearly, every **L**-derivable rule is also **L**-admissible, but the opposite usually does not hold for the logics under consideration. The syntactic proofs of admissibility of rules are sometimes hard to obtain. Note also that the set of derivable rules is preserved with respect to stronger logics, but it does not hold for admissible rules in general. These two classes of rules will be of interest for us not in the context of axiomatic systems but rather in other kinds of deductive systems. For considerations on their interrelations it will be crucial to show that some rules primitive in one system may be shown to be secondary in the other.

5.4 Relational Semantics

Now we focus on semantical characterization of modal logics. The most popular semantic approach to normal modal logic is based on the use of relational frames (models) often called *possible worlds semantics* or Kripke frames, although independently of Saul Kripke [169] semantics of this kind were introduced by many other logicians e.g. [160, 132].[13] Although this approach has serious limitations – it is not suitable not only for weak modal

[12]This is a particular exemplification of distinctions introduced in Chapter 1.
[13]Detailed history of these early investigations may be found in [71].

logics (like congruent or monotonic) but also for many normal and regular ones (cf. e.g. [112]) – it is still the most popular and simple way of interpreting normal modal logics. The popularity of this approach is connected with the fact that it offers a very natural and philosophically motivated interpretation of modal operators. It is also a natural tool for interpretation of many other nonclassical logics like intuitionistic logic and superintuitionistic logics being sublogics of **CPL**, conditional logics or relevant logics. In the family of regular logics this kind of semantics needs adjustments, we describe them briefly in Section 5.4.4. For weak logics we need a different kind of semantics due to Scott [245].[14]

Definition 5.4 (Frame) A modal frame is a structure $\mathfrak{F} = \langle \mathcal{W}, \mathcal{R} \rangle$, where $\mathcal{W} \neq \varnothing$ is the set of states (possible worlds), and $\mathcal{R}$ is a binary relation on $\mathcal{W}$, called *accessibility relation.*

In alethic modal logics $\mathcal{R}ww'$ means that w' is accessible from w (possible relative to w); $\mathcal{R}(w) = \{w' : \mathcal{R}ww'\}$ is the set of all alternatives for w. Frames for monomodal logics have only one such relation. In multimodal case instead of $\mathcal{R}$ we have a family of accessibility relations, each for one (pair of) modalities. For example, in bimodal temporal logics we may use frames $\langle \mathcal{W}, \mathcal{R}_F, \mathcal{R}_P \rangle$, where $\mathcal{W}$ is the set of time instants, $\mathcal{R}_F$ is the relation of time succedence, and $\mathcal{R}_P$ is the relation of time precedence. Complex modalities are interpreted set-theoretically i.e. $\mathcal{R}_\varepsilon$ is an identity relation, $\mathcal{R}_{a;b} := \mathcal{R}_a \circ \mathcal{R}_b$, and $\mathcal{R}_{a \cup b} := \mathcal{R}_a \cup \mathcal{R}_b$. In practice, for temporal logics we can still use frames with only one relation since the intended meaning of the second one is the converse of the first. So it is simpler to define:

Definition 5.5 (Temporal frame) A temporal frame is a structure $\mathfrak{T} = \langle \mathcal{T}, < \rangle$, where $\mathcal{T} \neq \varnothing$ is the set of time-instants and $<$ is a binary relation on $\mathcal{T}$ – *the flow of time relation.*

Definition 5.6 (Model) A model on the frame $\mathfrak{F}$ (or $\mathfrak{T}$) is any structure $\mathfrak{M} = \langle \mathfrak{F}, V \rangle$, where V is a valuation function on atoms ($V : AT \longrightarrow \mathcal{P}(\mathcal{W})$). The set of all points of a model $\mathfrak{M}$ will be referred to as $\mathcal{W}_\mathfrak{M}$; the set of all models on a frame $\mathfrak{F}$ will be referred to as $MOD(\mathfrak{F})$.

For temporal logic models are defined analogously on $\mathfrak{T}$. In what follows, we will usually state facts in general for modal logic and only in cases where the use of temporal language leads to different results we will point out the differences.

[14] One should note however that **M** is adequately characterized in terms of multimodal frames with countably many accessibility relations – cf. [93].

5.4.1 Interpretation

Due to the more complicated character of a semantics, the notion of an interpretation of a formula (and related semantical concepts) may be defined on different levels. The most basic is the notion of satisfaction of a formula in a state of a model, which is defined as follows:

$$
\begin{array}{lll}
\mathfrak{M}, w \vDash \varphi & \text{iff} & w \in V(\varphi) \text{ for any } \varphi \in PROP \\
\mathfrak{M}, w \vDash \neg\varphi & \text{iff} & \mathfrak{M}, w \nvDash \varphi \\
\mathfrak{M}, w \vDash \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \vDash \varphi \text{ and } \mathfrak{M}, w \vDash \psi \\
\mathfrak{M}, w \vDash \varphi \vee \psi & \text{iff} & \mathfrak{M}, w \vDash \varphi \text{ or } \mathfrak{M}, w \vDash \psi \\
\mathfrak{M}, w \vDash \varphi \rightarrow \psi & \text{iff} & \mathfrak{M}, w \nvDash \varphi \text{ or } \mathfrak{M}, w \vDash \psi \\
\mathfrak{M}, w \vDash \Box\varphi & \text{iff} & \mathfrak{M}, w' \vDash \varphi \text{ for any } w' \text{ such that } \mathcal{R}ww' \\
\mathfrak{M}, w \vDash \Diamond\varphi & \text{iff} & \mathfrak{M}, w' \vDash \varphi \text{ for some } w' \text{ such that } \mathcal{R}ww'
\end{array}
$$

and for temporal operators:

$$
\begin{array}{lll}
\mathfrak{M}, t \vDash G\varphi & \text{iff} & \mathfrak{M}, t' \vDash \varphi \text{ for any } t' \text{ such that } t < t' \\
\mathfrak{M}, t \vDash F\varphi & \text{iff} & \mathfrak{M}, t' \vDash \varphi \text{ for some } t' \text{ such that } t < t' \\
\mathfrak{M}, t \vDash H\varphi & \text{iff} & \mathfrak{M}, t' \vDash \varphi \text{ for any } t' \text{ such that } t' < t \\
\mathfrak{M}, t \vDash P\varphi & \text{iff} & \mathfrak{M}, t' \vDash \varphi \text{ for some } t' \text{ such that } t' < t
\end{array}
$$

In case when model is established we will simply write $w \vDash \varphi$. The set of all states where φ is satisfied in a model will be denoted as $\|\varphi\|_{\mathfrak{M}}$. Formally: $\|\varphi\|_{\mathfrak{M}} = \{w \in \mathcal{W}_{\mathfrak{M}} : w \vDash \varphi\}$. Usually we will use simply $\|\varphi\|$ when $\mathfrak{M}$ is established or unimportant. $\|\varphi\|$ is sometimes called a *proposition* expressed by φ. Although there are serious obstacles for considering this as a representation of a logical proposition we also follow this convenient habit.

The preceding definition states conditions for local (at a state in a model) satisfiability. Naturally the next step is the concept of global satisfiability in a model (and some related notion), defined as follows:

- $\mathfrak{M} \vDash \varphi$ iff $\forall_{w \in \mathcal{W}_{\mathfrak{M}}}, \mathfrak{M}, w \vDash \varphi$ (or $\|\varphi\|_{\mathfrak{M}} = \mathcal{W}_{\mathfrak{M}}$)

- *the content of a model* $\mathfrak{M}$ is $E(\mathfrak{M}) = \{\varphi : \mathfrak{M} \vDash \varphi\}$

Formulae globally satisfiable are often called universally true in a model. Both notions of satisfiability as well as other concepts may be extended to sets of formulae in the following way:

- $\mathfrak{M}, w \vDash \Gamma$ iff $\forall_{\varphi \in \Gamma}, \mathfrak{M}, w \vDash \varphi$

- $\mathfrak{M} \models \Gamma$ iff $\forall_{\varphi \in \Gamma}, \mathfrak{M} \models \varphi$
- $\|\Gamma\|_{\mathfrak{M}} = \bigcap \|\psi\|_{\mathfrak{M}}$ for $\forall_{\psi \in \Gamma}$

We say that Γ (or φ) is simply *satisfiable* iff there is some model and a state which locally satisfies Γ, otherwise Γ is *unsatisfiable*. In a dual manner we may define falsifiability – formally:

Definition 5.7 (Satisfiability, falsifiability) φ (Γ) is *satisfiable in a model* $\mathfrak{M}$ iff $\|\varphi\| \neq \varnothing$ $(\|\Gamma\| \neq \varnothing)$;
φ (Γ) is *satisfiable* iff, there is a model, where it is satisfiable;
φ (Γ) is *falsifiable in a model* $\mathfrak{M}$ iff $\|\varphi\| \neq \mathcal{W}_{\mathfrak{M}}$ $(\|\Gamma\| \neq \mathcal{W}_{\mathfrak{M}})$
(or: $\mathfrak{M}$ falsifies φ (Γ));
φ (Γ) is *falsified* iff, there is a model which falsifies it

And the last handy notion: the set of all models, where φ (or Γ) is globally satisfied is denoted as

$$Mod(\varphi) = \{\mathfrak{M} : \mathfrak{M} \models \varphi\} \; (Mod(\Gamma) = \{\mathfrak{M} : \mathfrak{M} \models \Gamma\})$$

5.4.2 Normal Logics

Semantical characterization of modal logics is connected not with particular models but with frames and their sets, otherwise we do not secure the closure under substitution.[15] This leads to further generalization of the notion of interpretation, namely validity at a state on a frame and validity on a frame. Both relations (and some related concept) are defined as follows:

- $\mathfrak{F}, w \models \varphi$ iff $\forall_{\mathfrak{M} \in MOD(\mathfrak{F})}, \mathfrak{M}, w \models \varphi$
- $\mathfrak{F} \models \varphi$ iff $\forall_{\mathfrak{M} \in MOD(\mathfrak{F})}, \mathfrak{M} \models \varphi$
- *the content of a frame* $\mathfrak{F}$ is $E(\mathfrak{F}) = \{\varphi : \mathfrak{F} \models \varphi\}$.

These relations may be generalized in a natural way to classes of frames (denoted by $\mathcal{F}$) which is of great importance for defining modal logics. Why? At this stage it is easy to observe that any $E(\mathfrak{F})$ is a normal logic. But the domain of any frame (the number or a character of objects) is of no importance for defining logics, only properties of accessibility relations play

[15] Of course, if we drop this condition from the definition of modal logic, we may characterize logics in terms of models – the content of every model is a modal logic in this sense (indeed normal logic). Such more general approach is presented e.g. in [112].

essential role in this respect. So dealing with uniform (modulo accessibility relations) classes of frames (or models) gives us the proper level of abstraction. In what follows we will be talking about classes of frames (models) with the same sort of relation (in monomodal case). For example, we will say that $\mathcal{F}$ (the class of frames) or $\mathcal{M}$ (the class of models) is reflexive, if every frame $\mathfrak{F} \in \mathcal{F}$ is reflexive (every model $\mathfrak{M} \in \mathcal{M}$ is reflexive), i.e. the relation of accessibility in every frame belonging to $\mathcal{F}$ is reflexive. We will use also generalizations of aforementioned concepts, in particular:

$MOD(\mathcal{F})$ is the class of all models built on any frame from $\mathcal{F}$ (or in other words $MOD(\mathcal{F}) = \bigcup\{MOD(\mathfrak{F}) : \mathfrak{F} \in \mathcal{F}\}$);

$E(\mathcal{F}) = \bigcap\{E(\mathfrak{F}) : \mathfrak{F} \in \mathcal{F}\}$ denotes the content of the class of frames $\mathcal{F}$ (analogously $E(\mathcal{M})$ denotes the content of the class of models $\mathcal{M}$);

$Mod_{\mathcal{F}}(\varphi) = Mod(\varphi) \cap MOD(\mathcal{F})$.

If we take the class of all frames we obtain the concept of (sheer) validity of a formula:

$$\models \varphi \text{ iff } \forall_{\mathfrak{F}}, \mathfrak{F} \vDash \varphi$$

It is well known fact that the set of all valid formulae in $\mathbf{L_M}$ coincides with $\mathbf{K}$. That is $\mathbf{K} = E(\mathcal{K})$, where $\mathcal{K}$ denotes the set of all frames. Similarly the set of all valid formulae in $\mathbf{L_T}$ coincides with $\mathbf{Kt}$.

Stronger logics over $\mathbf{K}$ or $\mathbf{Kt}$ are modeled by restricting the class of frames to those that satisfy some conditions on accessibility relation. This leads to the concept of restricted validity on the suitable class of structures:

$$\models_{\mathcal{F}} \varphi \text{ iff } \forall_{\mathfrak{F} \in \mathcal{F}}, \mathfrak{F} \vDash \varphi \text{ iff } \varphi \in E(\mathcal{F})$$

If some normal logic $\mathbf{L}$ is characterized in this way by some $\mathcal{F}$ we will say that $\mathcal{F}$ determines $\mathbf{L}$.

We say that Γ is $\mathcal{F}$-satisfiable ($\mathcal{F}$-unsatisfiable) if we restrict ourselves only to models belonging to $MOD(\mathcal{F})$.

In fact, the set of validities (or the content) of any $\mathcal{F}$ is a normal modal logic, although not every normal modal logic is characterized by some class of frames. Since, in what follows we will be dealing only with logics that posses such a characterization, we will usually identify logics with suitable sets of validities, but distinguish their several syntactic formalizations.

5.4.3 Expressive Strength of Ordinary Modal Language

We have already mentioned that important normal modal logics are determined by classes of frames satisfying some properties on accessibility relations. A great success of Kripke (and other relational) semantics in characterization of many modal logics, has led to more systematic research on the expressive power of modal languages. Among others, serious investigations started on the applicability of modal languages as description languages for several relational structures used in AI. In 70-ties van Benthem laid down the foundations of so called correspondence theory. But why modal languages may be used for talking about relational structures, and how much can they express? It is possible because formulae of $\mathbf{L_M}$ correspond to some relational conditions; more precisely:

Definition 5.8 (Correspondence) φ defines the class of structures $\mathcal{F}$ iff $\forall_{\mathfrak{F}}(\mathfrak{F} \models \varphi$ iff $\mathfrak{F} \in \mathcal{F})$

For example, well known axioms: T: $\Box\varphi \rightarrow \varphi$ defines reflexivity, 4: $\Box\varphi \rightarrow \Box\Box\varphi$ defines transitivity. The following table displays well known correspondencies for many axioms displayed in the table from Section 5.3.

Name	Condition	Axiom
Seriality (successors)	$\forall x \exists y \mathcal{R}xy$	D
Almost-seriality	$\forall xy(\mathcal{R}xy \rightarrow \exists z \mathcal{R}yz)$	$\Box D$
Almost-functionality	$\forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \rightarrow y = z)$	DC
Reflexivity	$\forall x \mathcal{R}xx$	T
Almost-reflexivity	$\forall xy(\mathcal{R}xy \rightarrow \mathcal{R}yy)$	$\Box T$
Transitivity	$\forall xyz(\mathcal{R}xy \wedge \mathcal{R}yz \rightarrow \mathcal{R}xz)$	4
Almost-transitivity	$\forall xyzv(\mathcal{R}xy \rightarrow (\mathcal{R}yz \wedge \mathcal{R}zv \rightarrow \mathcal{R}yv))$	$\Box 4$
Density	$\forall xy(\mathcal{R}xy \rightarrow \exists z(\mathcal{R}xz \wedge \mathcal{R}zy))$	$4C$
Symmetry	$\forall xy(\mathcal{R}xy \rightarrow \mathcal{R}yx)$	B
Almost-symmetry	$\forall xyz(\mathcal{R}xy \wedge \mathcal{R}yz \rightarrow \mathcal{R}zy)$	$\Box B$
Euclideaness	$\forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \rightarrow \mathcal{R}yz)$	5
Church-Rosser property (or directedness)	$\forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \rightarrow \exists v(\mathcal{R}yv \wedge \mathcal{R}zv))$	2
Strong (right) connectedness	$\forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \rightarrow \mathcal{R}yz \vee \mathcal{R}zy)$	3
Weak (right) connectedness	$\forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \rightarrow \mathcal{R}yz \vee \mathcal{R}zy \vee y = z)$	L
F	$\forall xyz(\mathcal{R}xy \wedge \neg\mathcal{R}yx \rightarrow (\mathcal{R}xz \rightarrow \mathcal{R}zy))$	F
R	$\forall xyz(\mathcal{R}xy \wedge x \neq y \rightarrow (\mathcal{R}xz \rightarrow \mathcal{R}zy))$	R
Predecessors	$\forall x \exists y \mathcal{R}yx$	DP
Strong (left) connectedness	$\forall xyz(\mathcal{R}yx \wedge \mathcal{R}zx \rightarrow \mathcal{R}yz \vee \mathcal{R}zy)$	$3P$
Weak (left) connectedness	$\forall xyz(\mathcal{R}yx \wedge \mathcal{R}zx \rightarrow \mathcal{R}yz \vee \mathcal{R}zy \vee y = z)$	LP

On the basis of the table we may establish many determination results for numerous normal logics; e.g. **S4** is determined by the class of frames with quasi-ordering accessibility relation (i.e. reflexive and transitive), since $\mathbf{S4} = \mathbf{KT4}$, T defines reflexivity, and 4 – transitivity.

In what follows we will be using the notation $MOD(\mathbf{L})$ instead of $MOD(\mathcal{F})$ and $\models_{\mathbf{L}}$ instead of $\models_{\mathcal{F}}$, whenever $\mathbf{L} = E(\mathcal{F})$. Any $\mathfrak{F}$ from $\mathcal{F}$ will be called an **L**-frame, and any model on such a frame – an **L**-model. We will apply names: serial logics, reflexive logics, linear logics e.t.c. – for classes of logics determined by suitable classes of frames.

Moreover, a standard modal language is expressive enough to define not only elementary (i.e. first-order) conditions but also many important conditions which are expressible in second-order language, e.g. McKinsey axiom M: $\Box\Diamond\varphi \rightarrow \Diamond\Box\varphi$.[16] On the other hand, conditions below the line are not definable in $\mathbf{L_M}$; to express them we must use a bimodal language $\mathbf{L_T}$ (namely axioms $DP, 3P, LP$).

The main tool for investigations in correspondence theory is the standard translation function ST_x which translates modal formulae into first-order formulae with one free variable, in accordance with the definition of satisfaction relation. It may be defined as follows:

$$\begin{array}{lcl}
ST_x(p) & = & Px \\
ST_x(\bot) & = & x \neq x \\
ST_x(\neg\varphi) & = & \neg ST_x(\varphi) \\
ST_x(\varphi \odot \psi) & = & ST_x(\varphi) \odot ST_x(\psi) \\
ST_x(\Diamond\varphi) & = & \exists y(\mathcal{R}xy \wedge ST_{x/y}(\varphi)) \\
ST_x(\Box\varphi) & = & \forall y(\mathcal{R}xy \rightarrow ST_{x/y}(\varphi))
\end{array}$$

where $\odot \in \{\wedge, \vee, \rightarrow\}$ and y is a variable not occurring in $ST_x(\varphi)$.

Since relational models may be treated as models for first-order correspondence language, it may be shown that:

Lemma 5.2 *For all $\varphi, w, \mathfrak{M}, \mathfrak{F}$ the following holds:*

$$\begin{array}{rcl}
\mathfrak{M}, w \models \varphi & \text{iff} & \mathfrak{M}, a^x_w \models ST_x(\varphi) \\
\mathfrak{M} \models \varphi & \text{iff} & \mathfrak{M} \models \forall x ST_x(\varphi) \\
\mathfrak{F}, w \models \varphi & \text{iff} & \mathfrak{F}, a^x_w \models \forall P_1, ..., \forall P_n ST_x(\varphi) \\
\mathfrak{F} \models \varphi & \text{iff} & \mathfrak{F} \models \forall P_1, ..., \forall P_n \forall x ST_x(\varphi)
\end{array}$$

[16] In fact, this condition is first-order definable in reflexive and transitive frames.

where $P_1, ..., P_n$ are standard translations of all propositional symbols in φ, and $\mathfrak{M}, a^x_w \vDash ST_x(\varphi)$ means that $ST_x(\varphi)$ is satisfied in $\mathfrak{M}$ under an assignment a^x_w where w is a value of free variable x in $ST_x(\varphi)$

This lemma shows that on the level of models standard modal language corresponds to first-order language, whereas on the level of frames it corresponds to second-order language. But this is only a general result; in fact there is a lot of elementary (first-order) frame conditions equivalent to second-order standard translations of modal formulae. Conditions mentioned above, like reflexivity, symmetry or transitivity may serve as good examples. The most general result showing which modal formulae define first-order conditions is due to Sahlqvist.

Definition 5.9 (Sahlqvist formula) Let boxed formula be any formula of the form $\Box^n\varphi, n \geq 0$ (called boxed atom, if $\varphi \in AT$), negative formulae be any formulae where each occurrence of an atom is in the scope of odd number of negations (otherwise it is positive). $\varphi \rightarrow \psi$ is *Sahlqvist implication* iff φ is built up from $\top, \bot$, boxed atoms and negative formulae with the help of $\vee, \wedge$ and $\Diamond$, and ψ is a positive formula. Finally, *Sahlqvist formula* is any boxed Sahlqvist implication, boxed conjunction of them, and a disjunction of Sahlqvist formulae that have no atoms in common.

The definition is quite complicated but it covers a large class of modal formulae and it will be an important point of reference for discussion on hybrid language expressivity in Chapter 11. Two important results are based on this concept:

Theorem 5.1 (Sahlqvist Correspondence) *Every Sahlqvist formula φ is equivalent on frames to some first-order condition effectively computable from φ by so called Sahlqvist-van Benthem algorithm.*

Theorem 5.2 (Sahlqvist Completeness) *Let Γ be any set of Sahlqvist formulae, then* H-$\mathbf{K} + \Gamma$ *is strongly complete for the class of frames defined by Γ.*

H-$\mathbf{K} + \Gamma$ is an axiom system obtained from H-$\mathbf{K}$ by the addition of Γ as the set of additional axioms. The last theorem is very important since we obtain automatically the completeness result for any logic which is axiomatizable by Sahlqvist formulae only.

One may observe that Geach axiom $\Diamond^m\Box^n\varphi \rightarrow \Box^s\Diamond^t\varphi$ is (a quite simple) instance of Sahlqvist formula. We are not going to formulate a first-order condition corresponding to Sahlqvist formula since it is even more complicated task, but the condition corresponding to Geach axiom is easy to state. It is just a generalization of Church-Rosser property corresponding to axiom 2

$$\forall xyz(\mathcal{R}^m xy \wedge \mathcal{R}^s xz \rightarrow \exists v(\mathcal{R}^n yv \wedge \mathcal{R}^t zv)) \tag{5.7}$$

where m, n, s, t denote the lengths of $\mathcal{R}$-paths in each case.

a, b, c, d-incestual axiom defines the following frame property:

$$\forall xyz(\mathcal{R}_a xy \wedge \mathcal{R}_c xz \rightarrow \exists v(\mathcal{R}_b yv \wedge \mathcal{R}_d zv)) \tag{5.8}$$

where a, b, c, d are indices of suitable, possibly complex modalities.

Although expressive abilities of $\mathbf{L_M}$ exceed first-order language (McKinsey axiom is an example) they have also serious limitations. Indeed, there are a lot of first-order conditions, often very simple, that are not modally definable even in $\mathbf{L_T}$. Below we list some of the more important:

Name	Condition
Irreflexivity	$\forall x \neg \mathcal{R}xx$
Asymmetry	$\forall xy(\mathcal{R}xy \rightarrow \neg \mathcal{R}yx)$
Antisymmetry	$\forall xy(\mathcal{R}xy \wedge x \neq y \rightarrow \neg \mathcal{R}yx)$
Intransitivity	$\forall xyz(\mathcal{R}xy \wedge \mathcal{R}yz \rightarrow \neg \mathcal{R}xz)$
Right directedness	$\forall xy \exists z(\mathcal{R}xz \wedge \mathcal{R}yz)$
Dichotomy	$\forall xy(\mathcal{R}xy \vee \mathcal{R}yx)$
Trichotomy	$\forall xy(\mathcal{R}xy \vee \mathcal{R}yx \vee y = z)$
Right discreteness	$\forall xy(\mathcal{R}xy \rightarrow \exists z(\mathcal{R}xz \wedge \neg \exists v(\mathcal{R}xv \wedge \mathcal{R}vz)))$

A famous Goldblatt–Thomason theorem establishes model theoretic criteria for definability of first-order conditions (for details consult e.g. [35]):

Theorem 5.3 *Elementary class of frames is definable by the set of modal formulae iff it is closed under construction of generated frames, disjoint unions and bounded morphic images, and reflects ultrafilter extensions.*

Which means that a first-order property of such a class of frames is preserved under taking one of these three operations, whereas its negation is preserved under taking ultrafilter extensions. All the classes of frames that satisfy some conditions from the table, break at least one of the four requirements, e.g. irreflexivity and asymmetry are not preserved under bounded morphic images. There are interesting general schemata of first-order formulae that are undefinable by standard modal languages because they cover conditions like irreflexivity. Some of the most important general classes of this sort are Horn clauses and geometric theories discussed in Section 1.1.5. Note that if we count $\bot$ as atom, irreflexivity may be stated as Horn clause of the form $\forall x(\mathcal{R}xx \to \bot)$; it is easy to find similar Horn formulations for asymmetry, antisymmetry, intransitivity.

There are some ways to overcome limitations of standard modal languages, e.g. the use of nonstructural rules of Gabbay [98]. But for other conditions it is not so simple and we need richer languages like hybrid ones that will be introduced in Chapter 11. As we will see, such richer languages allow of significant extensions of correspondence theory to important conditions undefinable by standard modal languages.

5.4.4 Regular Logics

Regular logics are not characterizable by ordinary Kripke frames but need some adjustments, namely the addition of a special category of worlds called nonnormal or queer.

Definition 5.10 (Augmented frame) An augmented modal frame is a structure $\mathfrak{F} = \langle \mathcal{W}, \mathcal{Q}, \mathcal{R} \rangle$, where $\mathcal{W} \neq \varnothing$ is the set of states (possible worlds), $\mathcal{Q} \subseteq \mathcal{W}$ is possibly empty set of queer worlds, and $\mathcal{R}$ is a binary relation on $\mathcal{W}$, called an *accessibility relation*.

Models on augmented frames are defined exactly as in Section 5.4.1 but the definition of satisfiability relation must be changed a bit. For all nonmodal formulae it is kept intact, for modal formulae we have a division of ways. For every $w \in \mathcal{W} - \mathcal{Q}$ (we may call them *normal worlds*) we have the same definition as before, but for queer worlds we just stipulate for every φ:

$$\mathfrak{M}, w \vDash \Diamond\varphi \quad \text{and} \quad \mathfrak{M}, w \nvDash \Box\varphi$$

R is characterized by the class of all augmented frames. As for stronger regular logics, we focus on the family of basic regular logics. But even in this family we must get rid of symmetric and euclidean logics if we want to consider those basic logics that are characterized by some classes of augmented frames. Reasons for that are simple: B simply forces every world to be normal, whereas 5 makes every queer world automatically normal to the same effect. Of course one can consider weaker forms of suitable axioms specific for this class of logics. For example, in Fitting [93] the class of regular logics of symmetric frames is described, axiomatized by the formula

B': $\varphi \to (\Box\top \to \Box\Diamond\varphi)$. D, T and 4 may be combined over **R** giving logics characterizable by classes of augmented frames, but suitable conditions on frames must be slightly modified as well. Reflexivity is restricted to normal worlds, whereas seriality is partly restricted:

seriality' $\forall_{w \in \mathcal{W}-\mathcal{Q}} \exists_{w' \in \mathcal{W}} \mathcal{R}ww'$

In case of logics containing 4 we must have an additional global condition of closenesses on frames: every world accessible to normal world is also normal. Transitive regular logics characterized by frames without this condition need a weaker axiom $4'$: $\Box\varphi \to \Box(\Box\top \to \Box\varphi)$ which corresponds to the following condition of partly restricted transitivity:

transitivity' $\forall_{w_1, w_2 \in \mathcal{W}-\mathcal{Q}} \forall_{w_3 \in \mathcal{W}} (\mathcal{R}w_1w_2 \wedge \mathcal{R}w_2w_3 \to \mathcal{R}w_1w_3)$

So for further considerations we reserve the following regular logics having adequate characterization in terms of augmented frames satisfying respective conditions: **R**, **RD** (Fitting's **CD**, Lemmon's **D2**), **RT** (Fitting's **CT**, Lemmon's **E2**), **R4** (Fitting's **CN4**, Lemmon's **D4**), **RD4** (Fitting's **CND4**, Lemmon's **D4**), **RT4** (Fitting's **CNS4**, Lemmon's **E4**).

5.4.5 Weak Logics

Neither congruent nor monotonic logics are, in general, determined by Kripke frames. Fortunately, they are determined by neighborhood frames, a kind of more general relational semantics.

Definition 5.11 (Neighborhood Frame) Let $\mathfrak{F} = \langle \mathcal{W}, \mathcal{N} \rangle$ be a neighborhood frame where $\mathcal{W} \neq \varnothing$ is the set of states (possible worlds), and $\mathcal{N}$ is a function $\mathcal{N} : \mathcal{W} \longrightarrow \mathcal{P}(\mathcal{P}(\mathcal{W}))$. A model on the frame $\mathfrak{F}$ is any structure $\mathfrak{M} = \langle \mathfrak{F}, V \rangle$, where V is a valuation function on atoms ($V : AT \longrightarrow \mathcal{P}(\mathcal{W})$).

Satisfaction of a formula in a state of a model is defined as in Kripke models with the exception of modals that are evaluated as follows:

$$\mathfrak{M}, w \vDash \Box\varphi \quad \text{iff} \quad \|\varphi\| \in \mathcal{N}(w)$$
$$\mathfrak{M}, w \vDash \Diamond\varphi \quad \text{iff} \quad -\|\varphi\| \notin \mathcal{N}(w)$$

All semantic notions from preceding sections apply without changes to neighborhood frames so we pay attention only to questions characteristic for the most important congruent and monotonic logics. Let $\mathcal{M}$ denote the set of neighborhood frames (or shortly $\mathcal{M}$-frames) satisfying the following monotonicity condition:

(m) if $X \in \mathcal{N}(w)$ and $X \subseteq Y$, then $Y \in \mathcal{N}(w)$, for any $X, Y \subseteq \mathcal{W}$

The following determination result holds:

Theorem 5.4 (Adequacy)

- $\varphi \in \mathbf{E}$ iff it is valid on all neighborhood frames
- $\varphi \in \mathbf{M}$ iff it is $\mathcal{M}$-valid

We will consider also extensions of **E** and **M** obtained by the addition of axioms $D, T, 4, B$ and 5 – the class of basic congruent and monotonic logics. Any extension of **E** or **M** by some axioms $X, Y, ..., Z$ will be denoted as **EXY...Z** (**MXY...Z** respectively). Moreover, we will consider all strengthenings of these logics additionally closed under (RG) (or equivalently with N added as an axiom). It is reasonable, because without K, the addition of (RG) does not change them into normal logics which is the case in the class of regular logics. These extensions will be called EN-logics (MN-logics respectively). Any logic containing D will be called D-logic, containing T – T-logic and so on.

The following conditions on neighbourhood frames correspond to respective axioms and (RG):

(d) if $X \in \mathcal{N}(w)$, then $-X \notin \mathcal{N}(w)$
(t) if $X \in \mathcal{N}(w)$, then $w \in X$
(4) if $X \in \mathcal{N}(w)$, then $\{w' : X \in \mathcal{N}(w')\} \in \mathcal{N}(w)$
(b) if $w \in X$, then $\{w' : -X \notin \mathcal{N}(w')\} \in \mathcal{N}(w)$
(5) if $X \notin \mathcal{N}(w)$, then $\{w' : X \notin \mathcal{N}(w')\} \in \mathcal{N}(w)$
(n) $\mathcal{W} \in \mathcal{N}(w)$

The following determination result holds for all logics under consideration (cf. [69]):

Theorem 5.5 (Adequacy)

- $\varphi \in$ **EXY..Z** iff it is valid on all neighborhood frames satisfying conditions $(x), (y), ..., (z)$
- $\varphi \in$ **MXY..Z** iff it is valid on all $\mathcal{M}$-frames satisfying conditions $(x), (y), ..., (z)$

5.4.6 Entailment

The concept of an entailment (consequence relation) may be defined in at least two[17] nonequivalent ways:

1. φ *follows locally* in $\mathcal{F}$ from Γ:

- $\Gamma \models_{\mathcal{F}} \varphi$ iff $\forall_{\mathfrak{M} \in MOD(\mathcal{F})} (\|\Gamma\|_{\mathfrak{M}} \subseteq \|\varphi\|_{\mathfrak{M}})$
  (or $\forall_{\mathfrak{M} \in MOD(\mathcal{F})}, \forall_{w \in \mathcal{W}_{\mathfrak{M}}}$ (if $\mathfrak{M}, w \vDash \Gamma$, then $\mathfrak{M}, w \vDash \varphi$))

2. φ *follows globally* in $\mathcal{F}$ from Γ:

- $\Gamma \mid\models_{\mathcal{F}} \varphi$ iff $Mod_{\mathcal{F}}(\Gamma) \subseteq Mod_{\mathcal{F}}(\varphi)$
  (or $\forall_{\mathfrak{M} \in MOD(\mathcal{F})}$ (if $\mathfrak{M} \vDash \Gamma$, then $\mathfrak{M} \vDash \varphi$))

Note the following:

Lemma 5.3

1. if $\Gamma \models_{\mathcal{F}} \varphi$, then $\Gamma \mid\models_{\mathcal{F}} \varphi$
2. $\Gamma \mid\models_{\mathcal{F}} \varphi$ iff $\Box^n\Gamma \models_{\mathcal{F}} \varphi$, where $\Box^n\Gamma = \{\Box^n\varphi : \varphi \in \Gamma\}$
3. $\Gamma \models_{\mathcal{F}} \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is $\mathcal{F}$-unsatisfiable.

In what follows we will be rather concerned with local consequence relation because of the reasons mentioned already in Section 5.3.1.

The rules of axiomatic systems (primitive or secondary) may be divided from the semantic point of view on two groups:

- $\Gamma \,/\, \varphi$ is $\mathcal{F}$-normal iff $\Gamma \models_{\mathcal{F}} \varphi$
- $\Gamma \,/\, \varphi$ is $\mathcal{F}$-valid iff $\models_{\mathcal{F}} \Gamma$ implies $\models_{\mathcal{F}} \varphi$

Note that in the set of primitve rules used in Section 5.3. for axiomatization of modal logics only (MP) is normal (and hence also valid) rule in every logic we consider; remaining ones are in general only valid rules. This semantic characterization of rules gives a useful semantic criterion for admissibility of rules, namely:

[17] We omit frame consequence relation which is not recursively characterizable cf. e.g. [93].

Lemma 5.4 $\Gamma \;/\; \varphi$ *is* **L**-*admissible, if* **L** *is determined by some* $\mathcal{F}$ *and* $\Gamma \;/\; \varphi$ *is* $\mathcal{F}$-*valid.*

This result applies also to admissibility of proof construction rules in ND systems with respect to $\mathcal{F}$-normality preserving rules (the notion of $\mathcal{F}$-validity may be seen as a special instance, sufficient for H-systems, of the more general notion of $\mathcal{F}$-normality preservation).

5.5 Completeness, Decidability and Complexity

We have already mentioned that stronger logics (in the semantic sense) are modeled by classes of frames where relation of accessibility satisfies some conditions. It was a great success of relational semantics that many well known (in axiomatic sense) modal logics like Feys' **T** or Lewis' **S4** and **S5** obtained simple semantic characterizations. The link between syntactic formalizations of **L** and classes of frames $\mathcal{F}$ is obtained via soundness and completeness theorems of the form:

- (Soundness) if $\Gamma \vdash_L \varphi$, then $\Gamma \models_{\mathcal{F}} \varphi$
- (Completeness) if $\Gamma \models_{\mathcal{F}} \varphi$, then $\Gamma \vdash_L \varphi$

The last one is often formulated equivalently:

- if Γ is **L**-consistent, then Γ is $\mathcal{F}$-satisfiable

If the first theorem holds, then **L** is sound with respect to $\mathcal{F}$, if the second holds, then **L** is (strongly) complete with respect to $\mathcal{F}$. If **L** is adequate (i.e. both sound and complete) with respect to $\mathcal{F}$, then $\mathcal{F}$ characterizes **L** or **L** is determined by $\mathcal{F}$.

Note that if Γ is empty (in the first formulation) or finite, we have a weak completeness, otherwise we have a strong form (i.e. admitting infinite Γ). There are modal logics which are weakly complete, but not strongly complete, with respect to some class of frames.

Standard proofs of completeness for modal logics apply well known construction of a canonical model which is based on Henkin/Lindenbaum result concerning maximalization of consistent sets. As a result we obtain a unique infinite model belonging to $MOD(\mathbf{L})$ that falsifies every formula unprovable in **L**. For questions of decidability and automated theorem proving it is more important that for many logics under consideration there are constructive methods of proving completeness. They show how to find for any unprovable formula some finite falsifying model. In particular, for many normal logics we obtain special falsifying models based on rooted frames in the sense specified below.

Definition 5.12 (Rooted Frames) *A frame* $\mathfrak{F} = \langle\ \mathcal{W}, \{\mathcal{R}_i\}\rangle$ is rooted if there is $w_0 \in \mathcal{W}$, such that $\mathcal{W} = \{w : \mathcal{R}^+ w_0 w\}$, where $\mathcal{R} = \bigcup\{\mathcal{R}_i\}$; a model based on such frame is called *rooted model.*

Let us recall that $\mathcal{R}^+$ denotes transitive closure of $\mathcal{R}$. Such a frame is generated by w_0 with the help of $\mathcal{R}^+$. We will be dealing almost always with falsifying models of this sort. Moreover, for many logics frames of such models are in fact trees or they turn into trees if we take as nodes not single points but their *clusters.*

Definition 5.13 (Cluster) Let $\mathfrak{F} = \langle \mathcal{W}, \mathcal{R}\rangle$ be any transitive frame, a cluster $\mathcal{C}$ is a maximal subset of $\mathcal{W}$, such that $\mathcal{R}$ is universal relation in it. A cluster is *degenerate* if it contains only one point w, such that $\neg\mathcal{R}ww$ (w is irreflexive).

A nondegenerate cluster is *simple* if it contains only one point w, such that $\mathcal{R}ww$ (w is a reflexive point), otherwise it is *proper* (contains at least two different points).

In the following definition we do not assume that $\mathcal{R}$ is transitive:

Definition 5.14 (Parachute) A pair consisting of a degenerate cluster $\{w_0\}$ and a nondegenerate cluster $\mathcal{C}$, such that some elements of $\mathcal{C}$ are $\mathcal{R}$-accessible from w_0 is called a parachute; if all elements of $\mathcal{C}$ are $\mathcal{R}$-accessible from w_0, then it is a *complete parachute.*

Note that if $\mathcal{R}$ is transitive, then a parachute must be complete.

The following table displays completeness results for the most important monomodal normal logics. The middle column contains results obtainable by canonical model method; so there are results that may be read off from the table in Section 5.4.3. The right column gives determination results in terms of finite frames. Detailed data concerning sources may be found in Goré [116]; a characterization of euclidean but not transitive logics comes from Kracht [166]. We do not display results for temporal logics, since they are, in principle, the same as for monomodal logics. The reader interested in details should consult e.g. Burgess [60] or Goldblatt [112].

L	**L**-frames	Finite **L**-frames
K	Any	Intransitive and irreflexive trees
D	Serial	Intransitive trees with reflexive leafs
T	Reflexive	Intransitive and reflexive trees
K4	Transitive	Trees of finite clusters
KB	Symmetric	Degenerated or simple clusters, or intransitive symmetric trees
K5	Euclidean	Single finite clusters or parachutes
KD5	Serial and euclidean	Single finite nondegenerate clusters or parachutes
KDB	Serial and symmetric	Simple clusters or intransitive symmetric trees
B	Reflexive and symmetric	Symmetric trees of reflexive points
K4B	Transitive and symmetric	Single finite clusters
K4D	Transitive and serial	Trees of finite clusters with nondegenerate clusters as leafs
K45	Transitive and euclidean	Single finite clusters or complete parachutes
KD45	Serial, transitive and euclidean	Single finite nondegenerate clusters or complete parachutes
S4	Reflexive and transitive	Trees of finite nondegenerate clusters
S5	Equivalential	Single, finite, nondegenerate clusters
K4.3	Transitive and weakly connected	Sequences of finite clusters
S4.3	Transitive, reflexive and connected	Sequences of finite nondegenerate clusters

All modal propositional logics considered in this book are decidable. It is implied by the fact that they satisfy finite model property and are axiomatizable in finite way (a profound result due to Segerberg [246]).

Clearly, being decidable does not mean being tractable or practically solvable. Even in **CPL** one may define quite simple formulae which would be tested by millennia. In pioneering era of investigation on decidability of mathematical theories it was sufficient to prove that the theory is decidable (or not). Since 70s the theory of complexity has grown up from computability theory, and many results were established for modal logics as well. We are not going to pursue in this book problems of computational complexity since this is not the subject of our study. But ocasionaly we will make some remarks concerning bahaviour of some (classes of) logics, e.g. in Chapter 11. So, for the reader's convenience we recall the most basic notions and facts. One may find a good introduction to the subject, as well as concrete results concerning modal logics, in [35] or in [67], where more advanced references are also displayed.

First, we recall the basic complexity hierarchy for easy reference:

$P \leq NP \leq PSPACE \leq EXPTIME$

where:

- P is the class of polynomial problems, trated as practically solvable since, for any input, known deterministic algoritms work in time bounded by some polynomial of the size of an input.

- NP is the class of nondeterministic polynomial problems, i.e. there are known polynomial solutions but with indeterministic choice involved.

- $PSPACE$ is the class of problems solvable by deterministic algorithms in polynomial space of the size of an input.

- $EXPTIME$ is the class of problems solvable by deterministic algorithms in exponential time of the size of an input.

All that we know for sure is that $P \neq EXPTIME$ but it is strongly believed that all these classes of problems are indeed different. Problems which belong to some class C and to which all other problems in this class are polynomially reducible (i.e. there are p-simulated; cf. Section 1.2) are C-complete. Moreover, in case of classes of problems solvable by nondeterministic algorithms it makes sense to consider also the class of complementary problems, since it is not known if they have the same level of complexity. Strictly speaking this applies only to time-nondeterminism since it was demonstrated by Savitch that $PSPACE = NPSPACE$, hence, we introduce only the class $coNP$ of problems that are complementary to those in NP. In case of any deterministic class C it holds that $C = coC$. In particular, we do not know whether validity problem and complementary sat-problem (cf. Chapter 1) represent the same level of complexity if we know that one of them belongs to NP (or to $coNP$).

For our considerations we need only these classes of problems since considered modal logics do not belong to other classes. In particular:

- decidability (i.e. validity problem) of **CPL**, **S5**, **KD45**, **K4.3**, **Kt4.3** and **S4.3** is $coNP$-complete (their sat-problem is NP-complete).

- logics between **K** and **S4** are $PSPACE$-complete

These results are due to Ladner, Ono, Nakamura, Hemaspaandra and others (cf. detailed notes in [35]). One should note that complexity of decision problem for local entailment in modal logics is the same as for validity but for global entailment it belongs to higher level class. For example, if validity is decidable for **L** in $PSPACE$, then decidability of global entailment is $EXPTIME$. Multimodal homogeneous logics like **K** and **S4** with no interactive axioms are as complex as suitable monomodal logics, but in case of at least bimodal **S5** we also obtain $PSPACE$-completeness (the result due to Halpern and Moses). In case of interactive logics the complexity is usually higher.

One should note that in the sequel (cf. Chapters 9 and 10) we will be dealing with practical ways of establishing decidability via terminating algorithms of proof search. They tend to behave computationally worse than it is admitted by known bounds recorded above. For example, tableau systems for euclidean or linear logics are based on systems for **K** and accordingly algorithms for proof search in these logics are not better than for **K**. But even in case of **K** and other $PSPACE$-complete logics algorithms we will be dealing with are more complex and, in consequence, decision procedures for these logics are practically harder (i.e. exponential in the length of an input) than they might be. It follows from the fact that obtained sets of formulae provide complete descriptions of falsifying models.

5.6 First-Order Modal Logics

Propositional modal logics may be lifted to first-order languages in many nonequivalent ways. Garson [103] offers a detailed exposition of several approaches with discussion of their virtues and limitations. We will follow his conventions with respect to names of different variants of logics. Before we introduce concrete axiomatic systems and their semantics it is convenient to recall briefly some features that stand beyond different formalizations. It may help to make a choice of which logic best fits to our purposes. For more detailed treatment one should consult e.g. [103, 96].

5.6.1 Introductory Remarks

Below we point out only the most important factors that should be considered before we decide which variant of first-order modal logic (shortly **QML**) is the best for our needs. One should consider e.g.:

1. whether in different worlds there are the same objects or no

2. if the same, then we should ask if they must be in every world (do we have in every world the same domain?)

3. whether every name should have the same designate in every world or not

4. should every name have a designate in every world?

5. should a designate of a name be an object?

Let us examine the most popular choices. As for the first question there are basically two approaches:

- David Lewis [176] is the most famous advocate of so called *modal realism*. In this approach every world has a separate domain of objects. In order to talk about possible courses of action of some agent we must introduce the theory of counterparts. Lewis' theory is not commonly accepted and caused many objections concerning its strong ontological commitments and the lack of intuitiveness. For our aim the most important is that there are serious technical problems with adequate formalization of counterpart relation.

- More popular solution is to admit that the same objects may occur in different worlds. It seems to be more intuitive and certainly technically simpler in construction of adequate semantics.

The second question has a quite natural reflection in the construction of models; we may choose between:

- Models with *constant domain* – the same objects in different worlds but with different properties. This solution is technically simpler, in particular we may use **CQL** as a basis for modal logic. The problem is that traditional interpretation of quantifiers (existential import) gives nonintuitive results. A possible solution lies in the application of possibilistic reading of quantifiers (cf. Chapter 1).

- Models with *varying domains* – domains of different worlds need not be the same; every world has its own domain of objects existing in them (actualism).[18] This solution is more intuitive and keeps traditional

[18]By the way it is the only acceptable solution if we want to provide semantics for Lewis' approach – in fact it must be strengthened to the effect that the intersection of two different domains is empty.

interpretation of quantifiers but it is difficult to base on **CQL**. Either we should modify semantics or put restrictions on the language; both approaches are rather artificial. So it is more natural, particularly if we have individual names in a language, to admit **FQL** as a basis.

As for the third question we have also two choices:

- *Rigid denotation* – every name has the same designate in every world. This solution is technically simpler but not very intuitive; in particular it leads to serious difficulties with proper treatment of identity and definite descriptions. Saul Kripke [171] is well known advocate of this solution, at least with respect to individual names and general names for natural kinds.

- *Nonrigid denotation* – a name may have different designates in different worlds. This solution seems to be more intuitive but technically more complicated since even rules of **FQL** fail in such logics. However, Garsons [105] shows that we may overcome some difficulties if we have two sorts of names in a language (some rigid and some nonrigid).

The fourth point may lead to proper treatment of empty names (e.g. fictional objects) and assumes distinction between existence of an object in a world and designate of a name in that world. As Fitting and Mendelsohn pointed out in [96] these questions are often mistakenly identified.

Finally the fifth question concerns the status of designates of terms:

- Simple forms of semantics like truth value semantics based on substitution interpretation[19] just eliminate this problem by dispensing with domain of objects and reducing evaluation to atomic sentences. This approach is really simple and may serve as an excellent point of start in didactics of modal logics, but has some important limitations.

- The most popular approach is based on *objectual interpretation.* It is natural and intuitive in many cases but sometimes may generate technical problems following from the difference between interpretation of nonrigid terms and the domain of quantification.

- *Intensional interpretation* changes the perspective. Instead of objects (extensions) we take as designates of terms their intensions i.e. functions from the set of worlds into domain of objects. This approach

[19] cf. [105].

was convincingly advocated by Garson in [104] as the basic form of semantics. It has some intuitive motivation for some interpretation of modalities. For example in temporal setting it may be appealing to take temporally changing objects as sequences of time-slices. Even if in some interpretations (or some kinds of terms) operating on intensions may seem unintuitive, we can easily restore objectual interpretation by taking only constant functions from $\mathcal{W}$ into D under consideration. For our aims it is important that intensional interpretation provides an adequate semantics for syntactically simple systems of **QML**.

5.6.2 Identity

The introduction of identity makes a room for further problems and possible choices. We may divide the stage for philosophical and technical problems, but in fact they are hardly separable. The leading philosophical problem is that of a *transworld identity.* It concerns the criteria of identity of objects in different worlds. The problem is not really new, only the language of discussion is different.[20]

Again, from the standpoint of Lewis' modal realism, it does not even make sense of talking about transworld identity. Objects in different worlds are really different and we have rather a problem of criteria od similarity. But for these researchers who are not as radical as Lewis the problem is important. How do we identify the same object in different worlds or time points? What is responsible for believing that we deal with the very same object in different contexts despite possible differences? Is that something like individual essence or form (e.g. Duns Scotus' haecceitas) which guarantees sameness? One of the reasonable answer is provided by Kripke account of individual names as rigid designators. It does not make sense to consider how do we recognize the same object in different worlds since the object is already identified by name. Her identity is not a theme for discussion but something which is assumed in advance.

The most important technical problem is connected with apparent inadequacy of rules and axioms of identity theory in intensional contexts. Famous examples of morning/evening star due to Frege, or the number of planets due to Quine were devised to show that. Let us recall the last one: *The number of planets is 9. Nine is necessarily greater than 7. So, the*

[20] Probably the first formulation is the famous Plutarchus' story of Theseus' boat where, through the centuries, every piece of wood and metal was replaced with a new one by grateful Athenians.

number of planets is necessarily greater than 7. Although this argument seems to be invalid it may be stated as an example of correct application of Leibniz's Law *LL*:

$$a = 9,\ \Box(9 > 7)\ \models\ \Box(a > 7) \tag{5.9}$$

It is possible to avoid the problem by modification of a translation or by modification of logic of identity. In the first approach different strategies are applied.

One may possibly treat complex names from the example as not genuine terms. The Russelian method of description elimination will work in this case. But one may provide similar examples with individual names. Clearly, also in this cases we may proceed as if they are instances of (hidden) descriptions. But we have already discussed (in Chapter 1) disadvantages of such an approach.

Perhaps modified translation of modal phrases may lead to more natural solution. We enter here a famous problem of distinction between the two interpretations of modal functors: de dicto and de re. Traditionally it was conceived as a question whether modal operator applies to a sentence or to a thing described in modal sentence. In modern terms (not necessarily equivalent to traditional) it is stated as a distinction between modal operators occurring outside the scope of quantifiers and inside (thus having formulae with free variables in the scope of modal constant). It may be illustrated by simple example taken from [96]; a sentence "Something necessarily exists" is ambiguous since we may understand it and express it in two different ways:

- de dicto — $\Box\exists x Ex$
- de re — $\exists x \Box Ex$

Now, the first is a tautology in modal logic of nonempty domains, as we will see soon, whereas the second is a rather controversial claim on the existence of necessary being. It is easy to observe that if the conclusion of Quine argument is interpreted as de re statement – $\exists x(x = a \land \Box(x > 7))$ – we have still an example of correct reasoning but the conclusion does not seem to be false. In general, to resolve problems based essentially on scoping difficulties, we need much more elaborate machinery (cf. an application of lambda operator in [96]) but in this case even such simple solution works.

Of course Quine would be unconvinced by such a response since he was well known opponent of using de re modalities. According to Quine, even using them is a manifestation of some form of essentialism. But it is mistaken view.[21] Leaving the question of essentialism as a viable theory aside we should say that using de re modalities is at most a form of talking about necessary properties of objects, not of claiming that there are any. Even philosophers with minimalistic and reductionistic program should prefer formal tools that do not just eliminate some unquestionable linguistic phenomena. So it is rather an advantage of modal first-order language that we may express in it several philosophical positions in a neutral way.

But distinction between de re/de dicto modalities is not a solution of all problems. Perhaps some modification of the logic of identity is needed. If we apply (unrestricted) Leibniz's Law in modal first-order logic we may easily prove for any terms τ_1, τ_2 the thesis:

$$LI \; \tau_1 = \tau_2 \rightarrow \Box(\tau_1 = \tau_2)$$

It leads to elimination of contingent identities which seems to be counter-intuitive. Is it really necessary that Lech Kaczyński is the present president of Poland? Different reactions are possible to that problem. One of the most popular is to accept this thesis but to understand it properly. This solution is the consequence of Kripke's theory of rigid designators mentioned above. If two names apply to the same object it must be necessary because the denotation is rigid. But even Kripke himself restricts his theory to individual names (and general names of natural kinds), so if we deal with other kinds of names and do not want to eliminate them, LI is too strong.

[96] makes a distinction between the Leibniz's Law, or its strengthened form called Principle of Indiscernibility of Identicals, and Principle of Substitution. The latter claims that if two terms τ_1, τ_2 have the same designate, then any pair of sentences φ, $\varphi[\tau_1//\tau_2]$, where the second is a result of replacement, have the same value. The source of problematic arguments is the use of Substitution principle, which is simply false. On the other hand, LL is restricted to variables only and as such is rather uncontroversial. $x = y$ means that the same object is a value of both variables (under some assignment). LI is still provable on this basis, but only with variables in places of τ_i ($i \in \{1, 2\}$).

We may weaken LL in different way by restricting φ to atoms. This is the solution of Garson [105] which we follow in this book either. One advantage of this approach is that we do not loose anything in classical logic; full version of LL is easily proved by induction on the length of φ.

[21] The literature on these questions is enormous; one may consult [96] for readable account.

On the other hand in modal first-order logic neither full version of LL, nor theses like LI are provable. If we admit some terms as rigid we may stipulate LI as additional axiom for that category of names. Note that this simple syntactic solution must have its counterpart in semantics – an interpretation allowing nonrigid terms.

5.6.3 Semantics

Relational semantics for first-order modal logic is generally obtained by combining first-order models with modal frames but, due to a number of possible choices mentioned above, it takes several forms. Moreover, it's not the case that every propositional modal logic combined with every quantificational approach yields complete formalization. For the sake of simplicity we assume that we deal here only with monomodal normal logics, although it is worth mentioning that [14] contains interesting results on **QML** characterized by neighborhood semantics. In the class of normal logics we generally assume that **L** is any logic characterized semantically by classes of frames defined with universal implications; general strategies of completeness proof applied by Garson [104] fail for logics like **S4.2** where existential quantifier is involved in the condition for $\mathcal{R}$. Note that it applies also to seriality, but in this case the difficulty may be overcome, so all basic normal logics are covered by the formalizations to follow. Interesting remarks on special cases of incomplete first-order modal logics may be found in [136]. Also for simplicity we state our formulation of variants of semantics directly in terms of models and we do not state exactly all first-order counterparts of several semantic notions of (global, local) satisfiability, entailment e.t.c. introduced previously in a detailed way for propositional logics. A careful reader may redefine all of them on the basis of simplified characterisation given below.

Now we briefly introduce a family of models for several versions of **QML**. Let us start with the simplest version – models with the possibilist (constant) domain and rigid denotation (shortly PR-models).

Definition 5.15 (PR-Model) A possibilist rigid model is a structure $\mathfrak{M} = \langle \mathcal{W}, \mathcal{R}, D, V \rangle$, where $\mathcal{W}, \mathcal{R}$ is a standard modal frame, D is a nonempty domain, and V is an interpretation of nonlogical terms, defined as follows:

- $V(c) \in D$, for every (rigid) individual constant
- $V(A^n) \subseteq D^n \times \mathcal{W}$, for every n-argument predicate

An assignment a is defined in a standard way as $a : VAR \longrightarrow D$, similarly for the notion of x-variant.

Interpretation I of a term τ in a model and under an assignment is characterized as in classical models:

$$I(\tau) := \begin{cases} a(\tau) & \text{if } \tau \in VAR \\ V(\tau) & \text{if } \tau \in CON \end{cases}$$

Satisfiability of a formula in a world of a model and under an assignment is defined as follows:

$$\begin{array}{lll}
\mathfrak{M}, a, w \vDash P^n(\tau_1, ..., \tau_n) & \text{iff} & \langle I(\tau_1), ..., I(\tau_n), w\rangle \in V(P^n) \\
\mathfrak{M}, a, w \vDash \neg\varphi & \text{iff} & \mathfrak{M}, a \nvDash \varphi \\
\mathfrak{M}, a, w \vDash \varphi \wedge \psi & \text{iff} & \mathfrak{M}, a \vDash \varphi \text{ and } \mathfrak{M}, a \vDash \psi \\
\mathfrak{M}, a, w \vDash \varphi \vee \psi & \text{iff} & \mathfrak{M}, a \vDash \varphi \text{ or } \mathfrak{M}, a \vDash \psi \\
\mathfrak{M}, a, w \vDash \varphi \rightarrow \psi & \text{iff} & \mathfrak{M}, a \nvDash \varphi \text{ or } \mathfrak{M}, a \vDash \psi \\
\mathfrak{M}, a, w \vDash \Box\varphi & \text{iff} & \mathfrak{M}, w' \vDash \varphi \text{ for any } w' \text{ such that } \mathcal{R}ww' \\
\mathfrak{M}, a, w \vDash \Diamond\varphi & \text{iff} & \mathfrak{M}, w' \vDash \varphi \text{ for some } w' \text{ such that } \mathcal{R}ww' \\
\mathfrak{M}, a, w \vDash \tau_1 = \tau_2 & \text{iff} & I(\tau_1) = I(\tau_2) \\
\mathfrak{M}, a, w \vDash \forall x\varphi & \text{iff} & \mathfrak{M}, a^x_o \vDash \varphi \text{ for all } o \in D \\
\mathfrak{M}, a, w \vDash \exists x\varphi & \text{iff} & \mathfrak{M}, a^x_o \vDash \varphi \text{ for some } o \in D
\end{array}$$

One may easily note that all clauses except the first are identical to classical and modal ones; only the point of reference is doubled (w and a). The only significant difference is in the first clause, where n-argument predicate is explained in terms of $n + 1$ tuples. But this is only a technical trick which allows V to be defined in a simple way in rigid models.

Note that in defining first-order counterparts of several semantic notions of (global, local) satisfiability, entailment we must take into account also the presence of an assignment in valuation clauses. This, in general, may lead to a great proliferation of semantic notions, some of them having no real importance, at least for our aims. So let us display only the most important:

- φ is true at w in $\mathfrak{M}$ ($\mathfrak{M}, w \vDash \varphi$) iff $\mathfrak{M}, a, w \vDash \varphi$ for all a
- Γ is *satisfiable in a model* $\mathfrak{M}$ iff $\mathfrak{M}, a, w \vDash \Gamma$ for some w and a;
- Γ is **L**-*satisfiable* iff, there is **L**-model, where it is satisfiable

The first notion is a first-order analogue of satisfiability at a world in a model with no reference to an assignment. In fact if we restrict considerations to sentences it really doesn't matter which assignment we choose and the two notions ($w \vDash \varphi$ and $a, w \vDash \varphi$) coincide.

The concept of global satisfiability (or validity in a model) is now defined as in Section 5.4.1:

- $\mathfrak{M} \vDash \varphi$ iff $\mathfrak{M}, w \vDash \varphi$ for all w

All other notions of validity (in a frame, class of models/frames) go without changes by reference to this concept. Local entailment may be simply defined in terms of **L**-satisfiability.

- $\Gamma \models_L \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is **L**-unsatisfiable

Models with actualist (varying) domains and rigid denotation (AR-models) are defined as follows:

Definition 5.16 (AR-Model) An actualist rigid model is a structure $\mathfrak{M} = \langle \mathcal{W}, \mathcal{R}, D, d, V \rangle$, where $d : W \longrightarrow \mathcal{P}(D)$ is a function which assigns to every world a set of objects.

All other components of a model, as well as the notion of I and a, are defined as for PR-models. But we need here also for some versions of logics a notion of w-assignment a, where a codomain is restricted to $d(w)$.

The concept of satisfiability is defined by the same clauses, except the cases of quantifiers which read:

$$\mathfrak{M}, a, w \vDash \forall x\varphi \quad \text{iff} \quad \mathfrak{M}, a^x_o \vDash \varphi \text{ for all } o \in d(w)$$
$$\mathfrak{M}, a, w \vDash \exists x\varphi \quad \text{iff} \quad \mathfrak{M}, a^x_o \vDash \varphi \text{ for some } o \in d(w)$$

Informally $d(w)$ may be interpreted as the set of objects existing in w, whereas $D - d(w)$ is the set of possible (nonexistent in w) objects. Although individual constants may name nonexistent objects, quantifiers have existential import. Because of that connotation, but also because of technical conveniency, we define in AR-models also a clause for existence predicate E.

$$\mathfrak{M}, a, w \vDash E\tau \quad \text{iff} \quad I(\tau) \in d(w)$$

Note that one may introduce also possibilistic quantifiers (using e.g. Tarskian symbols $\bigwedge, \bigvee$ for that purpose) defined by clauses from PR-models (i.e. by reference to all D).

Typically various conditions are considered on AR-models; we list here only three:

- (EX): $d(w)$ may be required nonempty for every w
- (MON): d is $\mathcal{R}$-monotonic (*expanding domains*): $\mathcal{R}ww'$ implies $d(w) \subseteq d(w')$
- (AMON): d is $\mathcal{R}$-antymonotonic (*contracting domains*): $\mathcal{R}ww'$ implies $d(w') \subseteq d(w)$

The notion of φ being true at w should be weakened a bit. Instead of all assignments we take under consideration only w-assignments (i.e. having value in $d(w)$ only). The definition reads:

$\mathfrak{M}, w \vDash \varphi$ iff $\mathfrak{M}, a, w \vDash \varphi$ for all w-assignments a

Note that in consequence it leads to informal understanding of valid formulae not as universally true but rather as never false. Such weakening is necessary if we want to keep **CQL** as a basis.

Resignation from rigid denotation of (some) terms leads to more fundamental modification. Let AN-model be any actualist nonrigid model defined as follows:

Definition 5.17 (AN-Model) An actualist nonrigid model is a structure $\mathfrak{M} = \langle \mathcal{W}, \mathcal{R}, D, d, V \rangle$, where V is an interpretation of nonlogical terms in worlds, defined as follows:

- $V_w(c) \in D$, for every individual constant and world
- $V_w(P^n) \subseteq D^n$, for every n-argument predicate and world

All other components and the notion of an assignment are kept intact but interpretation I of a term τ under an assignment is also relativized to worlds in a model and defined accordingly:

$$I_w(\tau) := \begin{cases} a(\tau) & \text{if } \tau \in VAR \\ V_w(\tau) & \text{if } \tau \in CON \end{cases}$$

In the definition of satisfaction we must make the following adjustments:

$$\begin{array}{lll} \mathfrak{M}, a, w \vDash P^n(\tau_1, ..., \tau_n) & \text{iff} & \langle I_w(\tau_1), ..., I_w(\tau_n) \rangle \in V_w(P^n) \\ \mathfrak{M}, a, w \vDash \tau_1 = \tau_2 & \text{iff} & I_w(\tau_1) = I_w(\tau_2) \\ \mathfrak{M}, a, w \vDash E\tau & \text{iff} & I_w(\tau) \in d(w) \end{array}$$

Note the main difference: in rigid models we deal with extensions of terms, whereas in nonrigid we deal with intensions as well. So V_w or I_w picks up an extension of an expression in a world, whereas V or I simpliciter is an intension i.e. a function from $\mathcal{W}$ into the set of extensions. In fact one may define all types of models in a uniform way, having richer structures just from the beginning and obtaining suitable classes with the help of additional conditions. Thus, PR-model may be redefined as AN-model, where $d(w) = D$ and $V_w(c) = V_{w'}(c)$ for every w, w' and c. We prefer to complicate models when we move from simpler to more complex logic.

Notions of satisfiability, validity, entailment e.t.c. go without changes from preceding semantics.

Finally one may resign from objectual interpretation in favor of intensions (individual concepts) i.e. the set In of all functions f from $\mathcal{W}$ into D. This way difficulties may be overcome which follow from the fact that (nonrigid) terms refer to intensions whereas quantifiers refer to extensions. In order to avoid some technical problems a subset S of all possible intensions In is chosen which is assumed to cover substances. The simplest semantics of this kind is considered in Garson [105].

Let IN-model be any intensional nonrigid model defined below:

Definition 5.18 (IN-Model) An intensional nonrigid model is a structure $\mathfrak{M} = \langle \mathcal{W}, \mathcal{R}, D, d, S, V \rangle$, where $S \subseteq In$ is a nonempty subset of substances defined as above. The notion of V and I are like for AN-models, but definition of an assignment is changed: $a : VAR \longrightarrow S$. So $a(x)$ is now understood as some $f : \mathcal{W} \longrightarrow D$ not as an object from D.

In the definition of satisfaction we must change only clauses for quantifiers:

$$\mathfrak{M}, a, w \vDash \forall x\varphi \quad \text{iff} \quad \mathfrak{M}, a^x_f \vDash \varphi \text{ for all } f \in S \text{ such that } f(w) \in d(w)$$
$$\mathfrak{M}, a, w \vDash \exists x\varphi \quad \text{iff} \quad \mathfrak{M}, a^x_f \vDash \varphi \text{ for some } f \in S \text{ such that } f(w) \in d(w)$$

IN-models may be seen as corresponding to possibilistic models in having one set of substances for the whole model. In Garson [103] a version of intensional semantics is introduced which is like actualist semantics in having suitable sets of substances for every world. Let AIN-model be any such actualist intensional nonrigid model defined as follows:

Definition 5.19 (AIN-Model) An actualist intensional nonrigid model is a structure $\mathfrak{M} = \langle \mathcal{W}, \mathcal{R}, D, d, s, V \rangle$, where for every w, $s(w) \subseteq In$ is a nonempty subset of substances from the world w. The notion of V, I and a are like for IN-models.

But in the definition of satisfaction we must change not only clauses for quantifiers; E is needed as an intensional predicate identifying substances from considered world:

$$\begin{array}{lll} \mathfrak{M}, a, w \models \forall x\varphi & \text{iff} & \mathfrak{M}, a^x_f \models \varphi \text{ for all } f \in s(w) \\ \mathfrak{M}, a, w \models \exists x\varphi & \text{iff} & \mathfrak{M}, a^x_f \models \varphi \text{ for some } f \in s(w) \\ \mathfrak{M}, a, w \models E\tau & \text{iff} & I(\tau) \in s(w) \end{array}$$

Note that in the last clause instead of I_w, i.e. an extension of τ in w, we take its intension, i.e. a function from $\mathcal{W}$ into D.

The most general variant of intensional semantics, which serves as a basis for unification of other approaches, is considered in Garson [104]. We will call it FIN-model for full intensional nonrigid model, and define as follows:

Definition 5.20 (FIN-Model) A full intensional nonrigid model is a structure $\mathfrak{M} = \langle \mathcal{W}, \mathcal{R}, D, d, S, s, V \rangle$, where $S \subseteq In$ is a general set of substances, whereas $s(w) \subseteq In$ is world-relative set of substances for every w (and in general it is not required that $s(w) \subseteq S$). The notion of V, I and a are like for AIN-models.

In the definition of satisfaction we must change only the clauses for quantifiers:

$$\begin{array}{lll} \mathfrak{M}, a, w \models \forall x\varphi & \text{iff} & \mathfrak{M}, a^x_f \models \varphi \text{ for all } f \in In \text{ such that } f \in s(w) \\ \mathfrak{M}, a, w \models \exists x\varphi & \text{iff} & \mathfrak{M}, a^x_f \models \varphi \text{ for some } f \in In \text{ such that } f \in s(w) \end{array}$$

5.6.4 Some Logics

A lot of interesting logics may be characterized with the help of classes of models described above. We revue below their axiomatic formalizations. Again we proceed from the simplest to more complex.

Syntactically the simplest solution is just to take H-**CQL** and add suitable modal axioms and rules. This way we obtain H-**QPL-L**, where **L** is a modal logic under consideration. Surprisingly enough this simplest logic is not characterized by the simplest semantics. It is adequate with respect to the class of AR-models satisfying monotonicity condition (MON). An addition of identity to **QPL** requires additional axioms except ID and restricted LL, namely:

LI $\quad \tau_1 = \tau_2 \rightarrow \Box(\tau_1 = \tau_2)$
LNI $\quad \tau_1 \neq \tau_2 \rightarrow \Box(\tau_1 \neq \tau_2)$

They express rigidity of terms; we have already noted this with respect to LI, but LNI is also valid. Note that even if we strengthen LL to admit any φ, not just atoms, it is incomplete since LNI is not provable. Instead of LNI one may use:

MI $\quad \Diamond(\tau_1 = \tau_2) \rightarrow \tau_1 = \tau_2$

Q1-L is a class of logics characterized by PR-models for **L**. H-**Q1-L** is obtained in general by addition of Barcan Formula BF to H-**QPL-L**:

BF $\quad \forall x \Box \varphi \rightarrow \Box \forall x \varphi$

Note that in cases, where **L** is characterized by symmetric frames, BF is provable with the help of B. It means that for symmetric L, **QPL-L**=**Q1-L**. It is a consequence of the fact that BF corresponds to (AMON), and this condition together with (MON) yields the same domain for all worlds related by transitive closure of $\mathcal{R}$. In rooted frames we have just models with constant domain, and they are sufficient for determination of **QPL**. By the way, a formula-schema which corresponds to (MON) is a converse of Barcan Formula:

CBF $\quad \Box \forall x \varphi \rightarrow \forall x \Box \varphi$

It is not necessary to add it to axiomatic base of **QPL** because it is provable already in H-**QPL-K**. Extension to identity is the same as in **QPL**.

If we change **CQL** on **FQL** as quantificational basis we obtain other classes of logics which we list in order of increasing syntactical complexity.

The simplest solution **QS-L** is obtained by a combination of H-**FQL** with H-**L**. Logics of this kind are determined by **L**-classes of AIN-models. Since terms are in general nonrigid a proper treatment of identity requires only ID and restricted LL. Note that formulae of the form $E\tau$ are not treated as atomic, so, for example:

LLE $\quad \tau_1 = \tau_2 \rightarrow (E\tau_1 \rightarrow E\tau_2)$

is not a tautology of AIN-models. But the following rule is validity preserving and may be added to axiomatic basis:

$(LLE)\ \vdash \tau_1 = \tau_2\ /\ \vdash E\tau_1 \to E\tau_2$

The logic **G-L** which is the weakest system considered in [104] is like **QS-L** but both $F\forall E$ and $F\exists I$ must be restricted: only variables may be values of τ. These logics are determined by **L**-classes of FIN-models.

The logic **F-L** is the weakest system of [105]. It is like **G-L** but E is counted as atomic. It means in particular that LLE is provable in this system as a simple instance of LL. This class of logics is adequate with respect to IN-models for suitable **L**.

Q1R-L is a class of logics determined by AR-models for **L** with no other restrictions. Without identity it is just like **QS-L**; an adequate formalization is obtained by addition of H-**L** to H-**FQL**. Extension to identity is as in **QPL-L** since all terms are rigid. Also note that in contrast to **QS-L** (but like in **F-L**) formulae of the form $E\tau$ must be counted as atomic.

Interestingly enough if we want to obtain a formalization for logics of ANR models (i.e. nonrigidity combined with objectual interpretation) things become more complicated, at least in axiomatic setting. Complications with suitable quantificational rules for instantiation of nonrigid terms may be fortunately avoided if we admit also some sort of rigid terms. Garson shows that in such case it is enough to have suitable rules defined for rigid terms only. Hence these logics, which we call, after Thomason, **Q3-L** (in [105] they are called oS but in ND formulation) are formalized as **G-L** above, but still to prove completeness one may use generalized rules because standard axioms and rules of **FQL** are insufficient. For simplicity we state them only for $\forall$:

$(G\forall E)\quad \vdash \varphi_1 \to \Box(\varphi_2 \to ...\Box(\varphi_n \to \forall x\psi)...)$
$\quad / \vdash \varphi_1 \to \Box(\varphi_2 \to ...\Box(\varphi_n \to (Ey \to \psi[x/y])...)$
$(G\forall I)\quad \vdash \varphi_1 \to \Box(\varphi_2 \to ...\Box(\varphi_n \to (Ex \to \psi(x))...)$
$\quad / \vdash \varphi_1 \to \Box(\varphi_2 \to ...\Box(\varphi_n \to \forall x\psi)...)$

where $x \notin VF(\{\varphi_1, ..., \varphi_n\})$

Note that $(G\forall E)$ is restricted to variables only which are rigid. Clearly if we have some sort of other rigid terms in a language we may generalize this rule by admitting them also as possible substitution terms.

For completeness we need also the following rule:

$$(G = E) \quad \vdash \varphi_1 \to \Box(\varphi_2 \to ...\Box(\varphi_n \to x \neq \tau)...)$$
$$/ \vdash \varphi_1 \to \Box(\varphi_2 \to ...\Box(\varphi_n \to \bot)...)$$

where $x \notin VF(\{\varphi_1, ..., \varphi_n, \tau\})$

Remark 5.1 In the presentation of various forms of **QML** we have omitted some systems known from older literature on the subject, like e.g. Gabbay's systems **GKc** and **GKs** or Kripke's logic with no terms. One may find their description in Garson [103] and their ND formalization in Indrzejczak [139]. ♣

Chapter 6

Standard Approach to Basic Modal Logics

In this Chapter we focus on the class of non-axiomatic systems that are called standard in the sense of keeping intact all the machinery of suitable systems for classical logic. Extensions are obtained by means of additional modal rules. This group covers modal extensions of standard Gentzen SC, Hintikka-style modal TS,[1] and some ND systems.

No doubt systems of this sort are the earliest and the most popular non-axiomatic formalizations of modal logics in general. In this Chapter we restrict a presentation to formalizations of basic logics because most of the proposals are limited only to some logics in this family. Moreover, there is a straightforward correspondence of results between SC, TS and ND in standard approaches as far as they are stated as formalizations of basic logics, so they make a really uniform group. As we shall see in the next Chapter, this correspondence is sometimes broken when we left a family of basic logics.

Close resemblance of SC's and TS's of this type does not require explanation; in Chapter 3 we already explained that the latter is just an inversion and simplification of the former in classical logic, and it is true also for modal extensions. In what way standard ND-systems are related to these systems will be explained in what follows. For the time being we point out only one thing; a characteristic property of all these systems is that rules for modal operators lead to the loss of some information and/or modification of the rest. In SC and TS it is connected with putting side-conditions

[1] We mean the family of TS's called implicit by Goré [117].

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*,
Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_6,

on parametric formulae. In the most popular modal ND it is connected with the introduction of a special category of subproofs, where reiteration of formulae is limited.[2]

It seems reasonable to start with short presentation of standard SC systems for basic modal logic first, in Section 6.1. They may be seen as an easy point of reference, and in fact, they were historically the first forms of non-axiomatic formalizations of modal logics. Standard TS's are not discussed separately but collected together with SC's.

In case of ND we have a division of ways. There are four basic approaches to modality via ND-formalization which may be called standard since they do not alter radically the basic format of ND system for classical logic. They are based on: modalization of assumptions, modalization of rules, modalization of reiteration rule, application of modal assumptions. In fact, the last two may be seen as language-based variants of basically the same methods: Fitch's technique of modal (or strict) subderivations, where the former deals rather with $\Box$, whereas the latter is based on $\Diamond$. In many versions, including Fitch original work, these two approaches are indeed mixed together. They are separated by some authors, e.g. Fitting [93], because of semantic reasons; we will show one more difference, of syntactic character. In Sections 6.2, 6.3, and 6.4. we present all these approaches to modal ND, paying attention to practical matters, especially the scope of applicability. As a result only Fitch's approach seems to be reasonably extensive. The last two sections show how Fitch's approach to ND may be extended to weak modal logics, and to first-order modal logics.

6.1 Standard Sequent Calculi and Tableau Systems

6.1.1 Historical Remarks

It seems that extensions of standard SC to some modal logics were even earlier than invention of full-fledged semantics, since they were prior to famous works of Kripke [169], Hintikka [132] and others. The first modal sequent systems appeared in Feys [85] (for **S4**), Curry [75, 76] (for **S4** and **S5**), Ridder [233] (also **T**) and Kanger [160] (the same logics). However the most fundamental work in this field was due to Ohnishi and Matsumoto [197, 198], where some weaker logics like **S2** and **S3** were also formalized. This

[2] In fact, also a system of destructive resolution of [94] may be included in this group – it will be introduced later in Chapter 7 (Section 7.4.2).

line of investigation was then extended by Zeman [288] and Fitting [93] to many normal and regular logics.[3] In variety of later works this approach was extended to many other regular and normal logics including such important ones like **G** and **S4Grz**. Some of these extensions of standard SC will be described later in Chapter 7 and 9. More detailed historical remarks may be found e.g. in [93, 116, 280, 281]. SC systems for logics weaker than regular were introduced much later by Lavendhomme and Lucas [172], and by Indrzejczak [152, 154].

Although the first tableau formalizations of some modal logics due to Kripke were invented in late 50s we will not treat them as standard, and a discussion of them will be postponed to Chapter 7 because of reasons which will be explained therein. Tableau systems which are called standard in this book were devised much later. These are TS's in Hintikka format, introduced by Rautenberg [229] and extended by others (cf. Goré [116, 117] for detailed documentation) to variety of other normal logics.

6.1.2 Standard SC for Basic Modal Logics

As we mentioned above, standard modal SC is obtained just by addition of extra rules for modals to Gentzen SC for **CPL**. For example, to obtain a formalization of **K** in a language with box only (i.e. $\mathbf{L}_\Box$), it is sufficient to add only one rule:

$$(\Rightarrow \Box)\quad \frac{\Gamma \Rightarrow \varphi}{\Box\Gamma \Rightarrow \Box\varphi}$$

In case of a language with $\Diamond$ we need additionally the following rule:

$$(\Diamond \Rightarrow)\quad \frac{\varphi \Rightarrow \Delta}{\Diamond\varphi \Rightarrow \Diamond\Delta}$$

It is worth remarking that although addition of any of these rules to SC for **CPL** provides an adequate formalization of **K** in a language with one modal constant, the addition of both rules is not sufficient for completeness of **K** in full $\mathbf{L_M}$. It was already noted by Kripke [170] that such rules are insufficient for proving interdefinability of $\Box$ and $\Diamond$. In order to remedy the problem one should rather use the following pair of rules:

$$(\Rightarrow \Box')\quad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Box\Gamma \Rightarrow \Diamond\Delta, \Box\varphi} \qquad\qquad (\Diamond \Rightarrow')\quad \frac{\varphi, \Gamma \Rightarrow \Delta}{\Diamond\varphi, \Box\Gamma \Rightarrow \Diamond\Delta}$$

[3] In fact Zeman's work contains formalizations of the whole Lewis' family **S1**–**S5**.

The modification of these rules and the addition of others leads to formalization of many normal logics. In particular, to obtain **T** it is enough to add to the formalization of **K** the following pair of rules:

$$(\Box\Rightarrow)\ \frac{\varphi,\Gamma\Rightarrow\Delta}{\Box\varphi,\Gamma\Rightarrow\Delta} \qquad (\Rightarrow\Diamond)\ \frac{\Gamma\Rightarrow\Delta,\varphi}{\Gamma\Rightarrow\Delta,\Diamond\varphi}$$

The formalization of other logics, e.g. transitive, requires usually at least a modification of rules $(\Rightarrow\Box')$ and $(\Diamond\Rightarrow')$, taking into account a type of operations admissible on parametric formulae. For example, in order to get SC for **S4**, we add to **CPL** $(\Rightarrow\Diamond)$, $(\Box\Rightarrow)$ and the following variants of $(\Diamond\Rightarrow')$ and $(\Rightarrow\Box')$:

$$(\Rightarrow\Box^4)\ \frac{\Box\Gamma\Rightarrow\Diamond\Delta,\varphi}{\Box\Gamma\Rightarrow\Diamond\Delta,\Box\varphi} \qquad (\Diamond\Rightarrow^4)\ \frac{\varphi,\Box\Gamma\Rightarrow\Diamond\Delta}{\Diamond\varphi,\Box\Gamma\Rightarrow\Diamond\Delta}$$

Since, at least for basic logics, such an approach requires only a suitable modification of rules $(\Diamond\Rightarrow')$ and $(\Rightarrow\Box')$, it may be nicely summarized in the general schemata due to Fitting:

$$(\Rightarrow\Box^F)\ \frac{\Gamma^\star\Rightarrow\Delta^\natural,\varphi}{\Gamma\Rightarrow\Delta,\Box\varphi} \qquad (\Diamond\Rightarrow^F)\ \frac{\varphi,\Gamma^\star\Rightarrow\Delta^\natural}{\Diamond\varphi,\Gamma\Rightarrow\Delta}$$

where $\Gamma^\star$ and $\Delta^\natural$ are defined accordingly. In the table below we put definitions for 15 basic normal logics. For simpler comparison with other formalizations based on similar principles we use general π, ν-notation.

Logic	$\Gamma^\star$	$\Delta^\natural$
K, **D**, **T**	$\{\nu : \nu^i \in \Gamma\}$	$\{\pi : \pi^i \in \Delta\}$
S4	$\{\nu^i : \nu^i \in \Gamma\}$	$\{\pi^i : \pi^i \in \Delta\}$
K4, **D4**	$\{\nu : \nu^i \in \Gamma\} \cup \{\nu^i : \nu^i \in \Gamma\}$	$\{\pi : \pi^i \in \Delta\} \cup \{\pi^i : \pi^i \in \Delta\}$
KB, **DB**, **B**	$\{\nu : \nu^i \in \Gamma\} \cup \{\pi^i : \pi \in \Gamma\}$	$\{\pi : \pi^i \in \Delta\} \cup \{\nu^i : \nu \in \Delta\}$
S5	$\{\nu^i : \nu^i \in \Gamma\} \cup \{\pi^i : \pi^i \in \Gamma\}$	$\{\pi^i : \pi^i \in \Delta\} \cup \{\nu^i : \nu^i \in \Delta\}$
KB4	$\{\nu : \nu^i \in \Gamma\} \cup \{\nu^i : \nu^i \in \Gamma\}$ $\cup\{\pi^i : \pi \in \Gamma\}$	$\{\pi : \pi^i \in \Delta\} \cup \{\pi^i : \pi^i \in \Delta\}$ $\cup\{\nu^i : \nu \in \Delta\}$
K5, **KD5**	$\{\nu : \nu^i \in \Gamma\} \cup \{\pi^i : \pi^i \in \Gamma\}$	$\{\pi : \pi^i \in \Delta\} \cup \{\nu^i : \nu^i \in \Delta\}$
K45, **KD45**	$\{\nu : \nu^i \in \Gamma\} \cup \{\nu^i : \nu^i \in \Gamma\}$ $\cup\{\pi : \pi^i \in \Gamma\} \cup \{\pi^i : \pi^i \in \Gamma\}$	$\{\pi : \pi^i \in \Delta\} \cup \{\pi^i : \pi^i \in \Delta\}$ $\cup\{\nu : \nu^i \in \Delta\} \cup \{\nu^i : \nu^i \in \Delta\}$

Definitions of $\Gamma^\star$ and $\Delta^\natural$ above the line are taken from Fitting [93]. These below the line, are modeled on the rules due to Takano (for **KB4** from [269], and for **K5** and **KD5** from [270]) and Schvarts [253] (for **K45** and **KD45**) with slight modifications however, because in both authors original rules are defined in $\mathbf{L}_\Box$. In some cases it is possible to simplify the definition, but we will illustrate this later when discussing ND system based on the use of the reiteration rule.

Of course, for every reflexive logic we must add rules $(\Rightarrow \Diamond)$ and $(\Box \Rightarrow)$. In case of serial logics we need one additional rule instead:

$$(D \Rightarrow) \quad \frac{\Gamma^\star \Rightarrow \Delta^\natural}{\Gamma \Rightarrow \Delta}$$

The general form of rules introduced by Fitting is more convenient for compact description of the whole class of basic modal logics. Moreover, it contains tacit applications of weakening rules, so we can get rid of these rules from SC introducing generalized form of (AX). Additionally we can obtain proof-search procedures for those cases that admit cut elimination, but to save completeness of reflexive logics, we must change a bit $(\Box \Rightarrow)$ and $(\Rightarrow \Diamond)$:

$$(\Box \Rightarrow') \quad \frac{\Box\varphi, \varphi, \Gamma \Rightarrow \Delta}{\Box\varphi, \Gamma \Rightarrow \Delta} \qquad\qquad (\Rightarrow \Diamond') \quad \frac{\Gamma \Rightarrow \Delta, \varphi, \Diamond\varphi}{\Gamma \Rightarrow \Delta, \Diamond\varphi}$$

As we noted in Chapter 3 such a version of SC easily leads to creation of Hintikka type TS, just by taking the rules in an upside-down manner. To our SC rule $(\Rightarrow \Box)$ for **K** there corresponds $(\Box E)$ or $(\pi^i E)$ in generalized form:

$$(\Box E) \quad \frac{\Box\Gamma, \neg\Box\varphi}{\Gamma, \neg\varphi} \qquad\qquad (\pi^i E) \quad \frac{\Gamma, \pi^i}{\Gamma^\star, \pi}$$

A pioneer of this approach to modal logics is Rautenberg [229]; a detailed exposition of this technique applied to many normal logics may be found in Goré [117]. It may seem strange that we do not mention here the classical works of Kripke (e.g. [169]) which is much earlier. But the approach of Kripke is not standard in our sense, because it comprises a standard TS in Beth format for **CPL** embedded in the graphical representation of relational semantics. So it has a slightly different character than the solution of Rautenberg; it is not purely syntactical but rather semantical or even hybrid approach (cf. Introduction). That is why we will discuss Kripke approach in the next Chapter; Also Goré [117] is treating Kripke systems in this way and put them together with labelled TS under the name explicit systems.

In order to get SC (or TS) for regular basic logics one may add a side-condition on the application of the above rules to the effect of nonempty $\Gamma^\star$. This works only for the small group of logics we have considered in Chapter 5; some other regular (and quasi-regular) logics need further modifications described in [93].

6.1.3 SC for Weak Basic Logics

The case of monotonic and congruent logics is more complicated because we need a single rule corresponding to every axiom. On the other hand, contrary to normal and regular logics, SC presented below is modular. Lavendhomme and Lucas [172] defined standard SC for **E** and **M**; Indrzejczak [152, 154] presented formalizations for all basic monotonic and congruent logics. For simplicity we use a language with $\Box$ only. We need the following rules:

$$(E)\quad \frac{\varphi \Rightarrow \psi \qquad \psi \Rightarrow \varphi}{\Box\varphi \Rightarrow \Box\psi} \qquad (M)\quad \frac{\varphi \Rightarrow \psi}{\Box\varphi \Rightarrow \Box\psi}$$

$$(D\text{-}2)\quad \frac{\Rightarrow \varphi, \psi \qquad \varphi, \psi \Rightarrow}{\Box\varphi, \Box\psi \Rightarrow} \qquad (D)\quad \frac{\varphi, \psi \Rightarrow}{\Box\varphi, \Box\psi \Rightarrow}$$

$$(4\text{-}2)\quad \frac{\Box\varphi \Rightarrow \psi \qquad \psi \Rightarrow \Box\varphi}{\Box\varphi \Rightarrow \Box\psi} \qquad (4)\quad \frac{\Box\varphi \Rightarrow \psi}{\Box\varphi \Rightarrow \Box\psi}$$

$$(5\text{-}2)\quad \frac{\Rightarrow \Box\varphi, \psi \qquad \Box\varphi, \psi \Rightarrow}{\Rightarrow \Box\varphi, \Box\psi} \qquad (5)\quad \frac{\Rightarrow \Box\varphi, \psi}{\Rightarrow \Box\varphi, \Box\psi}$$

$$(B\text{-}2)\quad \frac{\Rightarrow \Box\varphi, \psi \qquad \Box\varphi, \psi \Rightarrow}{\Rightarrow \varphi, \Box\psi} \qquad (B)\quad \frac{\Rightarrow \Box\varphi, \psi}{\Rightarrow \varphi, \Box\psi}$$

SC for **CPL** with cut and weakening yields an adequate formalization of **E** after addition of (E), whereas addition of (M) gives us **M**. The extensions are obtained in a modular way by addition of the rules with suitable names; each axiom A $(= D, 4, 5, B)$ corresponds to the rule (A) on the basis of SC-**M**, and to the rule $(A\text{-}2)$ on the basis of SC-**E**. It is worth noting that in contrast to axiomatic formalizations, where we use the same formulae as axioms, in SC we must use different rules in the context of congruent logics and in the context of monotonic ones. One can easily check that characteristic rules in the right column are not sound in neighbourhood semantics for respective congruent logics. Only T is characterized with the same rule as in the class of normal and regular logics.

One may obtain additional calculi for other weak logics with the help of the following rules:

$$(C\text{-}3)\quad \frac{\varphi, \psi \Rightarrow \chi \qquad \chi \Rightarrow \varphi \qquad \chi \Rightarrow \psi}{\Box\varphi, \Box\psi \Rightarrow \Box\chi} \qquad (N)\quad \frac{\Rightarrow \varphi}{\Rightarrow \Box\varphi}$$

(N) is simply a sequent formulation of (RG) which allows us to get formalizations of EN-logics and MN-logics mentioned in the last Chapter. $(C\text{-}3)$ corresponds to axiom C: $\Box\varphi \wedge \Box\psi \rightarrow \Box(\varphi \wedge \psi)$ on the ground of **E**; it enables a formalization of the class of EC-logics. The addition of this rule to calculi for monotonic logics does not make sense since they collapse into regular ones (K is derivable). In fact, to change SC-**M** into SC-**R** we need much simpler rule (C): $\varphi, \psi \Rightarrow \chi \ / \ \Box\varphi, \Box\psi \Rightarrow \Box\chi$ which cannot be used for congruent logics because it is not sound in **EC**.

6.2 Some Standard ND for Modal Basic Logics

In contrast to the situation in SC and TS, we may distinguish more than one ND formalization for modal logics which may be called standard (since the basic ND system for **CPL** is not modified). There are four such approaches to modality via ND-formalization, based on: modalization of assumptions, modalization of rules, modalization of reiteration rule, application of modal assumptions. In fact, the last two approaches may be seen as language-based variants of basically the one method: Fitch's technique of modal (or strict) subderivations.

The first two approaches rather fail to be extensive, hence they are treated briefly in one section in contrast to the last two that have quite a satisfying scope of application and will be discussed thoroughly.

6.2.1 Modal Assumptions

The first approach to the extension of ND-techniques to modal logics, due to Curry [75], was based on the concept of modal assumptions. The idea is that the application of some rule of necessity introduction to a formula φ is dependent on the shape of undischarged assumptions of φ. It should be limited to cases where the set of assumptions is empty or consists only of somewhat modalized assumptions. In the first case it is simply an application of (RG), in the second, we must check all the assumptions whether they satisfy suitable conditions defined for respective logic. Curry [75] defined his system only for propositional **S4** but this approach was soon, and rather independently, extended by others. Borkowski and Słupecki [54] provide a system for **S4** and **S5** but in the language with strict implication as primitive. Prawitz [220] formalized the same modal logics but also on minimal and intuitionistic basis and in first-order language. A similar system for **S5** was provided by Corcoran [74].

The advantage of this approach lies in its independence of the format of ND system. Both Curry and Prawitz have used Gentzen's T-F-format, Borkowski and Słupecki, as well as Corcoran, have used Jaśkowski's format[4] Despite this format-independence, any variant of Gentzen's format seems to be better prepared for this solution because all actual assumptions of each formula are displayed. In Jaśkowski's format, a formula may be put in the scope of assumption on which it is not in fact dependent (cf. in this respect Section 2.4.3), hence the control over applicability of $\Box$ introduction is harder. This inconvenience may also lead to construction of more complicated proofs. This is the reason why, in this section, we will use Gentzen's S-system in Suppes' format for examples.

Modal formulae are defined for **S4** as any ν-formulae, and for **S5**, additionally as any π-formulae. Hence we have the following rule of necessity introduction:

$(\Box I_S)$ $\Gamma \Rightarrow \varphi$ / $\Gamma \Rightarrow \Box\varphi$,

where for **S4**, Γ consists of ν-formulae only, whereas for **S5** it may contain also π-formulae.

One may easily note that it is just a special case of suitable rule from SC with Δ empty for respective logics. For both of them we additionally need:

$(\Box E_S)$ $\Gamma \Rightarrow \Box\varphi$ / $\Gamma \Rightarrow \varphi$

with no constraints on Γ.

The above definition of modalized formulae provides a simple account of rules but, unfortunately, forces us to construct unnecessarily long and complicated proofs – there is an example in Suppes' format on the next page. The problem illustrated here is not only of practical nature, it has also some important theoretical aspect which we should describe briefly. Prawitz [220] has proved normalization theorem for many logics in ND-formalization. Unfortunately, not all proofs in his systems for **S4** and **S5** may be transformed into normal form. For example, in displayed proof maximum formulae are present in lines 10 (12) and 11.

[4] Corcoran in fact applied horizontal, instead of vertical, manner of displaying derivations, but this is only a slight departure of no real importance.

$\{1\}$	1	$\Box p \wedge \Box q$	*ass.*
$\{1\}$	2	$\Box p$	$(1, \wedge E_S)$
$\{1\}$	3	$\Box q$	$(1, \wedge E_S)$
$\{4\}$	4	$\Box p$	*ass.*
$\{4\}$	5	p	$(4, \Box E_S)$
$\{6\}$	6	$\Box q$	*ass.*
$\{6\}$	7	q	$(6, \Box E_S)$
$\{4, 6\}$	8	$p \wedge q$	$(5, 7, \wedge I_S)$
$\{4, 6\}$	9	$\Box(p \wedge q)$	$(8, \Box I_S)$
$\{4\}$	10	$\Box q \rightarrow \Box(p \wedge q)$	$(9, \rightarrow I_S)$
	11	$\Box p \rightarrow (\Box q \rightarrow \Box(p \wedge q))$	$(10, \rightarrow I_S)$
$\{1\}$	12	$\Box q \rightarrow \Box(p \wedge q)$	$(2, 11, \rightarrow E_S)$
$\{1\}$	13	$\Box(p \wedge q)$	$(3, 12, \rightarrow E_S)$
	14	$\Box p \wedge \Box q \rightarrow \Box(p \wedge q)$	$(13, \rightarrow I_S)$

To avoid the problem Prawitz proposed the second variant, where the definition of modal assumptions is more liberal. He introduces the concept of *essentially modal formula* (shortly emf-formula) which for **S4** is defined as follows:

(a) $\bot$ and $\Box\varphi$, for any φ are emf-formulae
(b) if φ and ψ are emf-formulae, then so are $\varphi \wedge \psi$ and $\varphi \vee \psi$.

For **S5** it is necessary to add in (a) $\Diamond\varphi$, and in (b) $\varphi \rightarrow \psi$; equivalently, and simpler, one can define the set of emf-formulae for **S5** as the set of formulae, where each variable is in the scope of $\Box$. Admissibility of these rules is justified by the following:

Lemma 6.1 *If φ is emf-formula with respect to* **S4** *(***S5***), then $\vdash \varphi \rightarrow \Box\varphi$ in* **S4** *(***S5***)*

Proof, by induction on the length of φ, in [220].

The latter definition allows of a proof for our example in normal form, because the assumption in line 1 is essentially modal. Unfortunately, this variant does not satisfy normalization theorem either. Notice that if in **S4** (or **S5**) $\Gamma \vdash \Box\varphi$, where Γ contains only emf-formulae, then also $\psi, \Gamma \vdash \Box\varphi$, where ψ is not emf-formula, but then $\psi \wedge \Gamma \vdash \Box\varphi$. Obviously $\psi \wedge \Gamma$ is not modalized according to our definition, hence the proof of $\Box\varphi$ on its basis requires some maximum formulae again.

Remark 6.1 Prawitz presented also the third version of ND-systems for **S4** and **S5** which was believed to satisfy normalization theorem, but it is a solution of rather different sort. It is admissible in them to apply $\Box$ introduction to formula based on any assumption, on condition that there are some modalized formulae in the proof connecting these assumptions and a formula in question. Prawitz has proved normalization theorem for this version in the language with no $\vee, \exists, \Diamond$; it was recently extended for **S5** to full language by Martins and Martins [184]. In fact, Prawitz solution is rather a variant of Fitch's approach described in Section 6.3. It is also of no practical importance because it only shows what to check in a completed proof, not how to construct it. However, Sieg and Cittadini [255] provided extension of their intercalation calculus to **S4** based on this solution.

Anyway, it was noticed by Medeiros [189] that original proof of Prawitz for **S4** is erroneous. She provided slightly modified system but her normalization proof is also incomplete, as was noticed by Andou [7]. On the other hand, von Plato [212] has shown that even the first version for **S4** is normalizable if we use in classical basis his general elimination rules; in case of $\Box$ (in F-format) it takes the form:

$(\Box E_G)$ if $\varphi \vdash \psi$ then $\Gamma, \Box\varphi \vdash \psi$ ♣

The serious drawback of this approach is due to its limited scope of application. It is not incidental that such systems were devised only for **S4** and **S5**. These are the logics for which modal rules from SC-formalizations have $\Gamma^\star$ and $\Delta^\natural$ defined as subsets of Γ and Δ; no modification of formulae from $\Gamma \cup \Delta$ is involved. We can obtain adequate formalizations also for regular counterpart of **S4** or for monotonic versions of these logics by demanding nonempty (or singular) set of modalized assumptions for premises in case of $\Box$ introduction. But it is certainly not obvious how to extend this approach to other logics.

We do not claim however that some other extensions of Suppes' format ND are not possible if we additionaly use some rules of different character. In Section 6.4.3 we propose alternative way of formalization suitable for Suppes' system; one may also consider an application of generalized rules for $\Box$ elimination described in Remark 6.5. In [241] Satre proposed ND system in Suppes' format for all modal logics considered in [173] which involves Curry's rules but add also other rules for $\Box$ introduction. Although Satre underlines the influence of Ohnishi/Matsumoto for his work, his rules have a different character. The problem is basically with adaptation of SC rules to ND in Gentzen's S-format. If we are ready to admit rules which perform some operations also on the set of assumptions, then we are free to take any of the SC rules presented in Section 6.1. This is the approach represented in appendix of [136]. But it leads to some problems with the realization of such calculus, if we prefer to use Suppes' format, i.e. not to use all sequents but just formulae (=succedents) with records of numbers of assumptions (=antecedent). In Satre's approach no operations are allowed on assumptions at the cost of having more complex rules. For **K** a suitable rule has the following form:

$$(S\text{-}K)\quad \Gamma_1 \Rightarrow \Box\varphi_1; ...; \Gamma_n \Rightarrow \Box\varphi_n;\ \varphi_1, ..., \varphi_n \Rightarrow \psi\ /\ \Gamma_1, ..., \Gamma_n \Rightarrow \Box\psi$$

By introduction of suitable restrictions on this rule and on $(\Box I)$ and addition of sequent forms of rules corresponding to D or T (like $(\Box E_S)$ stated above) Satre is able to obtain formalizations of all normal and regular logics from [173] (for quasi-regular ones he must add some other rules as well). We omit the details but present a proof of K as an example of application of $(S\text{-}K)$:

$\{1\}$	1	$\Box(p \to q)$	*ass.*
$\{2\}$	2	$\Box p$	*ass.*
$\{3\}$	3	$p \to q$	*ass.*
$\{4\}$	4	p	*ass.*
$\{3, 4\}$	5	q	$(3, 4, \to E_S)$
$\{1, 2\}$	6	$\Box q$	$(1, 2, 5, S\text{-}K)$
$\{1\}$	7	$\Box p \to \Box q$	$(6, \to I_S)$
	8	$\Box(p \to q) \to (\Box p \to \Box q)$	$(7, \to I_S)$

6.2.2 Modalization of Rules

Bull and Segerberg [59] proposed an original method for dealing with modal logics in ND-systems. The starting point is the observation that any rule correct in **CPL** should be still correct in any modal context suitably specified. The notion of a context is then explained in the following way: if $\Gamma \vdash \varphi$ in **CPL**, then the addition of n boxes to all elements of Γ and φ preserves deducibility. Essentially it is an n-ary application of the condition (RR) from Section 5.2. Here (RR) provides a justification of a modalization of any inference rule. For example, if in any F-system the occurrence of φ and $\varphi \to \psi$ in the proof allows us to add ψ by $(\to E)$, then by condition (RR), we can add $\Box^n\psi$ to this proof, if we have already $\Box^n\varphi$ and $\Box^n(\varphi \to \psi)$. Similarly, we can modalize all other inference rules, e.g. for $\wedge$ we obtain:

$(\Box^n \wedge I)$ $\Box^n\varphi, \Box^n\psi$ / $\Box^n(\varphi \wedge \psi)$
$(\Box^n \wedge E)$ $\Box^n(\varphi \wedge \psi)$ / $\Box^n\varphi$ (or $\Box^n\psi$)

Bull and Segerberg did more, because they also modalized all proof construction rules. Such a solution cannot be justified by condition (RR) alone, because it is sufficient only for justification of inference rules. Nevertheless it is in accordance with the starting motivation. Modalization of conditional proof and indirect proof is based on the following principles:

$[\Box^n COND]$ if $\Gamma, \varphi \vdash \psi$, then $\Box^n\Gamma \vdash \Box^n(\varphi \rightarrow \psi)$
$[\Box^n RED]$ if $\Gamma, \neg\varphi \vdash \bot$, then $\Box^n\Gamma \vdash \Box^n\varphi$

Both principles are derivable by **CPL** and (RR).

So in Bull/Segerberg's system there are no introduction and elimination rules for $\Box$; in case $n = 0$ all the rules are simply **CPL**-rules. Introduction of modal context yields a system adequate for **K**.

There is a problem of what realization fits best to such a system. Modalized inference rules may be applied in any format; of course, if it is S-system, then rules in the calculus are defined on sequents, for example for $\wedge$ we need the following:

$(\Box^n \wedge I_S)$ $\Gamma \Rightarrow \Box^n\varphi, \Delta \Rightarrow \Box^n\psi$ / $\Gamma, \Delta \Rightarrow \Box^n(\varphi \wedge \psi)$
$(\Box^n \wedge E_S)$ $\Gamma \Rightarrow \Box^n(\varphi \wedge \psi)$ / $\Gamma \Rightarrow \Box^n\varphi$ (or $\Gamma \Rightarrow \Box^n\psi$)

The problem arises with modalized proof rules. Bull and Segerberg suggested F-T-system, but it is not clear how, in practise, one should mark in such a proof a transition from the set of assumptions Γ to $\Box^n\Gamma$. If we use Jaśkowski's format, the system becomes quite similar to Fitch's approach based on the use of modalized reiteration rule (cf. the next section), with the only difference that there is no special rule for $\Box$ introduction because strict derivations may be entered by $[COND]$ and $[RED]$ with the addition of $\Box$ $n > 0$ times to show-formula.

It seems that this system may be simply modified to obtain a more general solution, independent of the basic format and enabling extensions to logics other than **K**. The point is the redundancy of the system. First of all, we can always keep $n = 1$ in the indices of $\Box$ in the definition of rules. Moreover, we may resign from the modalization of many rules. One of the possible solution is to modalize proof rules only; we may even use only one of them – $[\Box^n RED]$. We will explain how it works in the Remark 6.7 in the next section. But such a modification of Bull/Segerberg's system is not very original; it is in fact a variant of Fitch's system.

A better solution, still in accordance with the original motivation, is to limit the modalization only to inference rules. It is based on the natural interpretation of condition (RR), and ND-system thus obtained is not a variant of Fitch's system anymore, because it forces us to use different proof strategies. To get an adequate formalization for **K** without any modification of proof rules, one must allow of a modalization of inference rules with empty set of premises, which is simply an application of (RG). Although formalization of this kind is format-insensitive, practically it is simpler to combine it with S-format because of the last proviso; in Jaśkowski format it is not immediately evident if a formula is really not dependent on any assumptions.

A separate problem is the possibility of extension of this formalization to other logics. Bull and Segerberg suggest that such basic system for **K** is enough; all extensions should be obtained by axiomatic additions. Still we can consider whether in such a system some extensions can be obtained in different way. One simple solution, which is very natural in case of basic logics, is to introduce for every axiom $A \rightarrow B$ a corresponding inference rule $A \ / \ B$. Other proposal of making extensions may be found in Hawthorn [122] but we omit a presentation because it is rather a nonstandard solution based on the multiplication of types of strict derivations. Instead, we shall pay attention to modalization of theses and consider some variants of (RG). In such a way many normal logics may be formalized with relatively small effort. Consider the following rules:

$(RG^{\Box})$ You may put $\Box$ before φ, if the set of assumptions for φ is empty or φ is ν-formula.

$(RG^{\Diamond})$ You may put $\Box$ before φ, if the set of assumptions for φ is empty or φ is π-formula.

(RG^{M}) You may put $\Box$ before φ, if the set of assumptions for φ is empty or φ is m-formula.

It is evident that addition of $(RG^{\Box})$ yields **K4**, $(RG^{\Diamond})$ – **K5**, and (RG^{M}) – **K45**. Addition of inference rules covering the effect of D or T to any of them extends a formalization to **D, T, KD4, S4, KD5, KD45, S5**, so most of the basic logics are covered. In some cases we may obtain adequate result even on smaller basis, e.g. **KD45** may be simply formalized on the basis of a system with $(RG^{\Diamond})$ only, and a rule for D like $\Gamma \Rightarrow \Box\varphi \ / \ \Gamma \Rightarrow \Diamond\varphi$ (a reader is asked to prove axiom 4 in this system). Obviously, in such an approach we resign from the idea that there are no specific rules for modal functors but it seems that adding axioms to system for **K** is not better.

Remark 6.2 The system where only the inference rules are modalized is still redundant. It is enough to have a variant of (RG) and $(\Box^n \to E_S)$ as primary rules. All other modalized variants of inference rules are easily derivable. Let $\psi_1, ..., \psi_k \ / \ \varphi$ be any inference rule; its modalized variant may be derived in the following way:

$\{1\}$	1	$\Box^n \psi_1$
$\vdots$	$\vdots$	$\vdots$
$\{k\}$	k	$\Box^n \psi_k$
$\{k+1\}$	$k+1$	ψ_1
$\vdots$	$\vdots$	$\vdots$
$\{k+k\}$	$k+k$	ψ_k
$\{k+1, \ldots, k+k\}$	$k+k+1$	φ
$\vdots$	$\vdots$	$\vdots$
	$3k+1$	$\psi_1 \to (\psi_2 \to \ldots (\psi_k \to \varphi) \ldots)$
$\vdots$	$\vdots$	$\vdots$
	$3k+1+n$	$\Box^n(\psi_1 \to (\psi_2 \to \ldots (\psi_k \to \varphi) \ldots)$
$\vdots$	$\vdots$	$\vdots$
$\{1, \ldots, k\}$	$4k+1+n$	$\Box^n \varphi$

where lines from 1 to $k+k$ contain assumptions, line $k+k+1$ is obtained by the application of the rule in question to formulae from lines $(k+1)-(k+k)$, line $3k+1$ results by k applications of $(\to I_S)$ to line $k+k+1$, line $3k+1+n$ is obtained by n applications of (RG) to line $3k+1$, and finally, $4k+1+n$ follows by k applications of $(\Box^n \to E_S)$ to lines $1, \ldots, k$ and $3k+1+n$. ♣

6.3 Modalization of Reiteration Rule

Fitch [89, 91] is the author of the next, and probably the most popular, ND approach to modal logics. His solution is not universal, in the sense that it is not suitable for Gentzen's format; some variant of Jaśkowski's format is presupposed because it is essential in this approach to separate parts of the proof. We will present it as usual in the KM realization. Despite the limitation to Jaśkowski's format, Fitch's solution has one unquestionable advantage – the scope: [89] contains only ND system for **T** and **S4**,[5] [91] provides counterparts for some deontic logics. Siemens [256] extends this formalization further, finally in Fitting [93] one can find a uniform formalization for many regular and normal logics. Fitch's approach was generalized even further: for basic normal logics of strict implication by Cerrato [66], for many relevance logics by Anderson and Belnap [5] and for conditional logics by Thomason [275]. Some extensions, due to Indrzejczak, to bimodal temporal logics [140] and to first-order modal logics [141] will be presented below. These are only some examples of application of Fitch's idea.

We will show that this approach is the closest relative of standard SC and TS for modal logics. In case of basic logics we may even say that these approaches are equivalent in the sense that every logic adequately formalized in SC (or TS) is formalizable in Fitch-style ND and vice versa. This strict correspondence is destroyed for some of these modal logics which use modal SC (or TS) rules with more premises than one.

The basic idea of Fitch's approach is the introduction of special category of subproofs called *strict* or *modal*. In ND-system for **CPL** one can use any U-formula from an open derivation of k-degree in open derivation of any higher degree which is nested in it. In case of modal logic, if new subderivation is strict, then only special sort of U-formulae from outer derivation (or formulae obtained by some operation performed on them) may be used. The logic in question decides what kind of U-formulae (or their derivatives) is admissible. The idea is that one can obtain ND formalizations for different logics only by modeling the set of suitable formulae, keeping all the inference and proof rules intact.

Technically, the problem of control over admissible formulae on the level of realization is solved by the introduction of *reiteration rule* (*Reit*) which regulates the transfer of formulae from a parent derivation to its subderivations. We did not introduce explicitly this rule to KM-**CPL** since transfer of formulae was not restricted and the definition of proof was simpler on condition that we can use any U-formula in current subderivation. In modal setting such a rule must be explicitly stated and every application of any inference rule must be performed only on premises which are present in the current subderivation.

If we restrict our consideration to $\mathbf{L}_\Box$, then to obtain an adequate formalization of **K** we must add to the calculus for **CPL** only one rule of proof construction:

[5] In fact, Fitch did not realize that he formalized **T**; he claimed that his rules gives "almost **S2**".

$[NEC\text{-}K]$ if $\Gamma \vdash \varphi$, then $\Box\Gamma \vdash \Box\varphi$

The realization of this rule in KM format may be displayed as follows:

$$\begin{array}{ll} & \mathcal{D} \\ i & \text{SHØW: } \Box\varphi \\ & \quad\boxed{\begin{array}{l} \mathcal{D}' \\ \vdots \\ \varphi \end{array}} \\ k & \end{array}$$

where: $\Box\Gamma \subseteq \mathrm{U}(\mathcal{D})$, and $\Gamma \subseteq \mathrm{U}(\mathcal{D}')$, moreover we assume that the only elements of $\mathrm{U}(\mathcal{D}')$ are elements of Γ or formulae deduced from them with the help of admissible rules, and that there are no S-formulae in $\mathcal{D}'$.

Informally this rule can be read: if φ is derivable in the strict derivation of degree $k + 1$ based on formulae in Γ, then $\Box\varphi$ is deduced in the parent derivation of degree k from $\Box\Gamma$. The comparison of $[NEC\text{-}K]$ with SC-rule $(\Rightarrow \Box)$ makes clear the relationship between these approaches. The premise of SC rule corresponds to the strict subderivation closed in the box, whereas the conclusion corresponds to the outer derivation. We may generalize this rule in Fitting's style to cover extensions of **K**.

$[NEC]$ if $\Gamma^\star \vdash \varphi$, then $\Gamma \vdash \Box\varphi$

where $\Gamma^\star$ for **K** is like above (i.e. $\{\varphi : \Box\varphi \in \Gamma\}$) and for stronger logics additionally contains:

- $\{\Box\varphi : \Box\varphi \in \Gamma\}$ for transitive logics
- $\{\Diamond\varphi : \varphi \in \Gamma\}$ for symmetric logics
- $\{\Diamond\varphi : \Diamond\varphi \in \Gamma\}$ for Euclidean logics

To obtain a correct $[NEC]$ for e.g. **KB4** we just take a union of the first two sets. It is easy to see that a definition for $\Gamma^\star$ is just like this provided by the table in Section 6.2; the only difference is that instead of ν- and π-formulae we restrict considerations to $\Box$- and $\Diamond$-formulae. In order to cover serial or reflexive logics one must add suitable inference rules:

(D) $\Box\varphi \,/\, \Diamond\varphi$ or (T) $\Box\varphi \,/\, \varphi$

Clearly, a definition of derivation of φ in modal KM system should be changed accordingly. In all clauses describing the application of inference rules we must add a proviso that premises must be present in current subderivation, we must also state additional clauses concerning closure of a subderivation by $[NEC]$ and the application of reiteration.

$[NEC]$: Let $\Box\varphi$ be a show-formula of k-degree subderivation, then we can close this subderivation provided φ has appeared as U-formula in it, and there is no indirect assumption $\neg\Box\varphi$ under the show-line.

$(Reit)$: Let Γ be the set of U-formulae of k-degree subderivation and φ an element of Γ, then φ may be added to $k+1$-degree subderivation if either:

(a) opening show-formula is not of the form $\Box\psi$, or

(b) the first U-formula of this subderivation is an indirect assumption.

If neither (a) nor (b) is satisfied, then φ may be added if it belongs to $\Gamma^\star$.

The following example (a proof of a thesis of **S5**) shows how it works.

1	SHØW: $\Box p \wedge \Diamond q \rightarrow \Diamond\Box(p \wedge \Box\Diamond q)$	$[11, COND]$
2	$\Box p \wedge \Diamond q$	$ass.$
3	$\Box p$	$(2, \alpha E)$
4	$\Diamond q$	$(2, \alpha E)$
5	SHØW: $\Box(p \wedge \Box\Diamond q)$	$[10, NEC]$
6	p	$(3, Reit(K))$
7	$\Diamond q$	$(4, Reit(5))$
8	SHØW: $\Box\Diamond q$	$[9, NEC]$
9	$\Diamond q$	$(7, Reit(5))$
10	$p \wedge \Box\Diamond q$	$(6, 8, \alpha I)$
11	$\Diamond\Box(p \wedge \Box\Diamond q)$	$(5, T)$

In case of modal reiteration, in justification column we put in brackets the name of the corresponding modal axiom which admits this step, e.g. in line 6 we have an application of reiteration admissible already in **K**, whereas in line 7 and 9 it is admissible in **K5**. Applications of reiteration admissible for **CPL** will be justified just by $(Reit)$ with no parameter in brackets. In the last step we have used a contrapositive of (T).

In case of regular logics one should add to $[NEC]$ a condition to the effect that subderivation of $k+1$-degree may be closed by this rule if at least one usable-formula of k-degree derivation is $\Box$-formula. Definition of $\Gamma^\star$ for these regular logics we have considered is the same as in normal case.

It is quite easy to prove that this system is adequate with respect to all basic normal and regular logics. Completeness requires proofs of suitable axioms, which is routine; (RG) and (RR) is simulated by $[NEC]$. Soundness is also not very difficult to prove; we will do it in the next section.

Remark 6.3 Fitting [93] in his formalization for regular logics used different but equivalent solution; instead of $[NEC]$ he proposed a proof construction rule based on the principle:

$[MOD]$ if $\Gamma^\star \vdash \varphi \vee \psi$, then $\Gamma \vdash \Diamond\varphi \vee \Box\psi$ ♣

Remark 6.4 Rules $[NEC]$ and $(Reit)$ may be significantly simplified in our version of KM. First, as we remarked in Chapter 2, one can eliminate from KM both $[RED]$ and the rule for entering indirect assumptions, because these rules are not necessary in the system with our set of inference rules. In so modified system both $[NEC]$ and $(Reit)$ may be formulated as follows:

$[NEC']$ Let $\Box\varphi$ be a show-formula of k-degree subderivation, where φ has appeared as a usable-formula, then we can close this subderivation, provided all its usable-formulae justified by $(Reit')$ belong to $\Gamma^\star$.

$(Reit')$ Let Γ be the set of usable-formulae of k-degree subderivation, then we may add to $k+1$-degree subderivation either:

(a) $\varphi \in \Gamma^\star$, if show-formula of this subderivation is $\Box$-formula, or

(b) $\varphi \in \Gamma$

Even if $[RED]$ and indirect assumptions are kept intact, both rules can be simplified in case of reflexive logics; we can use $[NEC']$ and the following:

$(Reit'')$ Let Γ be the set of usable-formulae of k-degree subderivation, then we may add to $k+1$-degree subderivation φ belonging either to $\Gamma^\star$ or to Γ

This simplification is due to the fact that although in case of $[NEC]$, $(Reit)$ must be restricted to $\Gamma^\star$, then in case of other rules of closing a derivation both elements of Γ and $\Gamma^\star$ are admissible in reflexive logics, because formulae from $\Gamma^\star$ may be inferred by the application of (T). Also the condition that indirect assumption should not be present in the derivation to be closed by $[NEC]$ is not needed because for **T**, and its extensions treated here, one can prove the following as an admissible rule:

$[NEC_T]$ if $\Gamma^\star, \neg\Box\varphi \vdash \varphi$, then $\Gamma \vdash \Box\varphi$

Our primary formulation, despite the complications in the definition of (*Reit*), has one serious advantage. All the necessary restrictions are stated as the conditions to be satisfied before we apply the rule, hence we do not need to check a finished proof whether there are some mistakes. Simpler formulations, often found in literature, usually require some control of correctness after the proof is completed. ♣

Remark 6.5 The drawback of this formalization is the lack of any rule of $\Box$ elimination for logics weaker then **T**; in case of **D** one can use in this role an inference rule (D) $\Box\varphi$ / $\Diamond\varphi$, but it is an ad hoc solution. Some authors, like Garson [105], prefer yet another terminological convention. Given the definition of $\Gamma^\star$ for **K**, it is possible to define $(Reit(K))$ for strict derivation as a kind of $(\Box E)$, while allowing simple transfer of formulae in ordinary subderivations. This solution has also some disadvantages, when we consider how to obtain extensions of **K**. These forms of reiteration hardly may be treated in a similar way, because e.g. in **K4**, $\Box$ is not only eliminated but also the whole $\Box$-formula is put in a strict subderivation. Garson simply applies Bull and Segerberg's solution and add suitable axioms.

A different solution is possible, at least for some logics which are not reflexive, if we consider some variants of $(\Box E)$ defined on modalized formulae, e.g.:

$(\Box E^{\Box})$ $\Box\varphi$ / φ, for any $\Box$-formula φ
$(\Box E^{\Diamond})$ $\Box\varphi$ / φ, for any $\Diamond$-formula φ
$(\Box E^{\Box\Diamond})$ $\Box\varphi$ / φ, for any m-formula φ

$(\Box E^{\Box})$ may be added to **K5** and its extensions, $(\Box E^{\Diamond})$ to **KD4**, which implies that $(\Box E^{\Box\Diamond})$ is a suitable rule for **KD45**. In this way, at least some logics without T have $(\Box E)$ of some sort, but it should be noticed that all these rules are in fact derivable in the basic formalization, what is more, they cannot replace the rule (D) in **KD4**, or **KD5**. The exception is **KD45**, a very important epistemic logic; instead of replacement of $(\Box E)$ by (D) in ND-**S5**, one can use $(\Box E^{\Diamond})$, which is simpler and more natural. Proofs of $K, 4, 5$ run like in ND-**S5**, it is enough to show that D is also provable; as a shortcut we will use a secondary rule $(R5)$ $\Diamond\Box\varphi$ / $\Box\varphi$, which is obviously derivable either.

1	SHØW: $\Box p \to \Diamond p$	$[14, COND]$
2	$\Box p$	$ass.$
3	SHØW: $\Box\Diamond p$	$[13, RED]$
4	$\neg\Box\Diamond p$	$ass.$
5	$\Diamond\Box\neg p$	4,by def.
6	$\Box\neg p$	$(5, R5)$
7	$\Box p$	$(2, Reit.)$
8	SHØW: $\Box\Diamond p$	$[12, NEC]$
9	$\neg p$	$(6, Reit(K))$
10	p	$(7, Reit(K))$
11	$\bot$	$(9, 10, \bot I)$
12	$\Diamond p$	$(11, \bot E)$
13	$\bot$	$(4, 8, \bot I)$
14	$\Diamond p$	$(3, \Box E^{\Diamond})$

Remark 6.6 The definition of $\Gamma^\star$ may be simplified in some cases because it is redundant. **B** and **S5** can serve as examples. Let us define $\Gamma^\star$ for **B** as $\{\Diamond\varphi : \varphi \in \Gamma\}$ and for **S5** as $\{\Diamond\varphi : \Diamond\varphi \in \Gamma\}$. Admissibility is obvious, so it is enough to prove sufficiency of these definitions. Provability of (RG) is intact, B is still provable trivially with this $\Gamma^\star$ in both logics. It suffices to prove K in both systems and 4 in **S5** Notice that in both logics we may add also a secondary inference rule: (RB) $\Diamond\Box\varphi$ / φ. The proof of derivability of (RB) is obvious and its use simplifies a proof of K. Below we present a proof of it in ND-**B**, with weakened $(Reit)$:

1	SHØW: $\Box(p \to q) \to (\Box p \to \Box q)$	$[3, COND]$
2	$\Box(p \to q)$	$ass.$
3	SHØW: $\Box p \to \Box q$	$[6, COND]$
4	$\Box p$	$ass.$
5	$\Box(p \to q)$	$(2, Reit.)$
6	SHØW: $\Box q$	$[11, NEC]$
7	$\Diamond\Box(p \to q)$	$(5, Reit(B))$
8	$\Diamond\Box p$	$(4, Reit(B))$
9	$p \to q$	$(7, RB)$
10	p	$(8, RB)$
11	q	$(9, 10, \beta E)$

Analogously we can prove K in ND-**S5**, the only difference is that in line 4 and 5, we should first apply contrapositive of (T) $(\varphi\ /\Diamond\varphi)$ before it is possible to move the conclusions by $(Reit(S5))$ to the strict derivation. In **S5** one must also prove axiom 4:

1	SHØW: $\Box p \rightarrow \Box\Box p$	$[4, COND]$
2	$\Box p$	*ass.*
3	$\Diamond\Box p$	$(2, T)$
4	SHØW: $\Box\Box p$	$[6, NEC]$
5	$\Diamond\Box p$	$(3, Reit(S5))$
6	SHØW: $\Box p$	$[14, RED]$
7	$\neg\Box p$	*ass.*
8	$\Diamond\neg p$	(7,by def.)
9	SHØW: $\Box\neg\Box p$	$[11, NEC]$
10	$\Diamond\neg p$	$(8, Reit(S5))$
11	$\neg\Box p$	(10,by def.)
12	$\neg\Diamond\Box p$	(9,by def.)
13	$\Diamond\Box p$	$(5, Reit.)$
14	$\bot$	$(12, 13, \bot I)$

One could easily notice that in this way it is possible also to get ND system for **KB** and **KDB**, simply by dropping (T) or replacing it by (D). It is not possible to formalize **KB4** in this way. **KD45** admits a weaker definition of $\Gamma^\star$ either, because the above proof of 4 is also a proof in this logic (line 3 is justified by $(\Diamond I^{\Box})$, which is a dual version of $(\Box E^{\Diamond})$), so it is sufficient to define $\Gamma^\star$ for **KD45** as the union of $\Gamma^\star$ for **K** (instead of **K4**) and for **S5** in the latter version.

Clearly one can define suitable rules for SC and TS with modified definition of $\Gamma^\star$ (and $\Delta^\natural$) on the basis of the above considerations. But calculi with so modified rules may loose some important properties that hold for systems of Takano or Shvarts (cf. discussion in Section 6.2). ♣

Remark 6.7 When discussing Segerberg's approach, we have noticed that one possible extreme is to base it on the modalized form of indirect proof which was stated as:

$[\Box^n RED]$ if $\Gamma, \neg\varphi \vdash \bot$, then $\Box^n\Gamma \vdash \Box^n\varphi$

Here if $n = 0$ we have just ordinary $[RED]$, otherwise we obtain a rule strong enough to obtain ND for **K**. Since it is sufficient to have $n = 1$ in practice it is realized in such a way that to any proof in a system ND-**K**, as defined in this section, we add as an assumption $\neg\varphi$ under each show-formula $\Box\varphi$, and change a justification from $[NEC]$, to $[\Box^n RED]$. Clearly, for stronger logics we must keep suitable modal reiteration rule as defined above, but it still works because the following rule is obviously admissible:

if $\Gamma^\star, \neg\varphi \vdash \bot$, then $\Gamma \vdash \Box\varphi$

In practise it is easy to realise – just add an indirect assumption to every strict subderivation. ♣

6.4 Rules for Possibility

So far we have used $\Diamond$ as a definitional shortcut, which was in accordance with the usual practice of many authors. However, the problem of formalization of $\Diamond$ is sufficiently interesting in itself to be described separately. Moreover, as we shall see, an application of $\Diamond$ as primitive opens the way to define modal ND which is not committed to Jaśkowski's format, but may be used with any other format presented in Chapter 2.

6.4.1 Original Fitch's System

In the original system of Fitch for **T** and **S4**, $\Diamond$ was in fact treated as an independent functor and characterized by the pair of rules of introduction and elimination. On the level of calculus of our ND system the first of them is an introduction rule:

$(\Diamond I)\ \ \varphi\ /\ \Diamond\varphi$

which is of course normal only for reflexive logics. The second one is another proof construction rule:

$[POS]$ if $\psi, \Gamma^\star \vdash \varphi$, then $\Diamond\psi, \Gamma \vdash \Diamond\varphi$

Its application in KM schematically looks like this:

	$\mathcal{D}$	
i	SHØW: $\Diamond\varphi$	
$i+1$		ψ
		$\mathcal{D}'$
		$\vdots$
k		φ

where: $\Box\Gamma \cup \{\Diamond\psi\} \subseteq \mathrm{U}(\mathcal{D})$, $\Gamma \subseteq \mathrm{U}(\mathcal{D}')$, and ψ is a modal assumption. Moreover, we assume that the only elements of $\mathrm{U}(\mathcal{D}')$ are elements of Γ or formulae deduced from $\Gamma \cup \{\psi\}$ with the help of admissible rules, and that there are no S-formulae in $\mathcal{D}'$.

On the level of realization we must add to KM two rules that correspond to $[POS]$; one for closing a derivation and one for entering modal assumption:

$[POS]$ Let $\Diamond\varphi$ be the show-formula of k-degree subderivation with the first U-formula being modal assumption, then we can close this subderivation, provided φ has appeared in it as U-formula.

(*mod.ass.*) If $\Diamond\psi$ is U-formula of k-degree subderivation and $\Diamond\varphi$ is S-formula entering $k+1$-degree subderivation, then we may add ψ as a *modal assumption* of the $k+1$-degree derivation

In fact, also (*Reit*) should be modified; we introduce the version which works for the system in which both $[NEC]$ and $[POS]$ are counted as primitive. Because of the stylistic reasons it will be defined dually to our official formulation of (*Reit*) from Section 6.3.

Let Γ be the set of U-formulae of k-degree subderivation and $\varphi \in \Gamma^\star$, we may put φ into subderivation of $k+1$-degree by (*Reit*) if at least one of the following conditions is satisfied:

(a) show-formula of this subderivation is $\Box\psi$ and the first U-formula of this subderivation is not an indirect assumption

(b) the first U-formula of this subderivation is modal assumption;

if neither (a) nor (b) is satisfied, then we may put into this derivation any $\varphi \in \Gamma$

One should notice that although an introduction of modal assumption is an optional element (we may have some $\Diamond$-formula as show-formula but to try to close this subderivation by different rule), its presence is a necessary condition to close the derivation by $[POS]$.

Introduction of special rules for $\Diamond$ is very convenient in practice. Very often we may produce shorter proofs for many theorems (and usually with relatively smaller effort). But one should notice that the system containing both $[NEC]$ and $[POS]$ in such a version is still incomplete in the language with both modalities taken as primitive. To overcome the problem the original system of Fitch contains also 4 rules of elimination and introduction for negated modal formulae:

$$\begin{array}{ll}
(\neg\Box E) & \neg\Box\varphi \;/\; \Diamond\neg\varphi \\
(\neg\Box I) & \Diamond\neg\varphi \;/\; \neg\Box\varphi \\
(\neg\Diamond E) & \neg\Diamond\varphi \;/\; \Box\neg\varphi \\
(\neg\Diamond I) & \Box\neg\varphi \;/\; \neg\Diamond\varphi
\end{array}$$

The presence of these rules, let us call them *definitional rules*, raises the question: are they really necessary to save completeness? We will return to this question in the next Chapter (Section 7.3.2) – for the time being we just note that one may easily formulate the system in such a way that all these rules are dispensable.

6.4.2 Fitch's System Generalized

In order to avoid definitional rules it is enough to apply Fitting's generalized notation. Both proof construction rules obtain the form:

$$\begin{array}{lll}
 & \Gamma & \\
i & \text{SHØW: } \nu^i & \\
 & & \Gamma^\star \\
 & & \vdots \\
 & & \vdots \\
k & & \nu
\end{array}
\qquad\qquad
\begin{array}{lll}
 & \Gamma \cup \{\pi_1^i\} & \\
i & \text{SHØW: } \pi_2^i & \\
i+1 & & \pi_1 \\
 & & \Gamma^\star \\
 & & \vdots \\
k & & \pi_2
\end{array}$$

where $\Gamma^\star$ is defined exactly as in the table from Section 6.1 specifying sets of formulae in antecedents of sequents of suitable SC rules. It should be noted that with the generalized forms of proof construction rules it is possible to close by $[NEC]$ a derivation starting with SHOW:$\neg\Diamond\varphi$ provided we deduce U-formula $\neg\varphi$; in $[POS]$ we can add as a modal assumption $\neg\varphi$ if we have already $\neg\Box\varphi$ in the proof.

Such a system is very handy in use but quite redundant. In particular, one may construct a system based either on $[NEC]$ – we did it in the last section – or on (some form of) $[POS]$ (cf. Fitting [93]). We will show the mutual eliminability of these rules in the next Chapter after introduction of ND rules for weak modal logics. First, in the next subsection, we will explore what possible advantages can we get by choosing one of the extreme. Yet for practical purposes we rather advice to use a system in its most general form, because usually proof search is easier and completed proofs shorter. So this is our official version of KM for the basic normal and regular logics. Below, we display an example of a proof in **KB** using both forms of proof construction rules; a reader is invited to prove this thesis using only $[NEC]$ or only $[POS]$.

1	SHØW: $\Box(p \to \neg\Diamond q) \land \Diamond p \to \neg q$	$[5, COND]$
2	$\Box(p \to \neg\Diamond q) \land \Diamond p$	$ass.$
3	$\Box(p \to \neg\Diamond q)$	$(2, \alpha E)$
4	$\Diamond p$	$(2, \alpha E)$
5	SHØW: $\neg q$	$[15, RED]$
6	q	$ass.$
7	SHØW: $\Box\Diamond q$	$[8, NEC]$
8	$\Diamond q$	$(6, Reit(B))$
9	$\Box(p \to \neg\Diamond q)$	$(3, Reit)$
10	$\Diamond p$	$(4, Reit)$
11	SHØW: $\neg\Box\Diamond q$	$[14, POS]$
12	p	$m.ass.(10)$
13	$p \to \neg\Diamond q$	$(9, Reit(K))$
14	$\neg\Diamond q$	$(12, 13, \beta E)$
15	$\bot$	$(7, 11, \bot I)$

Note the generalized form of $[POS]$ used in line 11; $m.ass.(10)$ denotes a modal assumption with indication of the line where it comes from.

We already stated that this system is complete even without $[POS]$ but we should anyway demonstrate that it is not too strong. In proving soundness we will apply the technique introduced in Chapter 2. It is fairly easy to check that inference rules like (T) or other considered, are **L**-normal for suitable logics, and that the new proof construction rules are normality preserving with respect to all basic logics. So we just state the first lemma and demonstrate one case for the second.

Lemma 6.2 *Every inference rule is* **L***-normal in suitable modal logic* **L** *and its extensions.*

Lemma 6.3 *Every proof construction rule is normality preserving in all basic normal or regular logics.*

PROOF Consider $[POS]$. Assume that $\Gamma^\star, \varphi \models \psi$ but $\Gamma, \Diamond\varphi \not\models \Diamond\psi$. So in some w all elements of Γ as well as $\Diamond\varphi$ holds, but $w \not\models \Diamond\psi$. Hence for some accessible w', $w' \vDash \varphi$ but $w' \nvDash \psi$. Consider elements of $\Gamma^\star$, all are taken by modal reiteration from Γ, so there is a division of cases according to which logic is under consideration. Take **K45** as an example and some arbitrary $\chi \in \Gamma^\star$. Either $\Box\chi$ is in Γ or χ is in Γ and χ is some m-formula. In all cases $w' \vDash \chi$. The first case holds for every model since $w \vDash \Box\chi$ and $\mathcal{R}ww'$. If χ is ν-formula it holds by transitivity of $\mathcal{R}$; if χ is π-formula it holds by euclideaness of $\mathcal{R}$. So $w' \vDash \Gamma^\star$ which implies that $w' \vDash \psi$ and we have a contradiction. Thus $\Gamma, \Diamond\varphi \models \Diamond\psi$. ■

Now, in order to prove soundness, we must change a bit a definition of justified subproof. In fact, the introduction of reiteration rule leads to simplification of this definition (and a proof) since we do not use as premises of inference rules any formula from outer derivation. Let $\mathcal{D}$ be any proof and consider any subproof $\mathcal{D}'$ of degree $n > 0$ contained in it. We say that this subproof is *justified* iff, $\Gamma \models \psi_n$, where ψ_n is the last formula of $\mathcal{D}'$, and Γ is the set of all formulae of $\mathcal{D}'$ introduced as assumption or by reiteration.

The proof proceeds as in Chapter 2, by double induction: on the depth k of $\mathcal{D}$ and on the length of its subproofs. In the basis we again consider all subproofs of degree k, i.e. with no subproofs inside, and of length n, and show that $\Gamma \models \psi_i$ $(1 \leq i \leq n)$ which implies that they are justified.

Basis: $i = 1$, so ψ_1 is an assumption or a formula introduced by reiteration, and the claim follows by reflexivity and monotonicity of $\models$. Assume for any i such that $i < k \leq n$ the claim holds and consider ψ_k. If ψ_k is by reiteration, then again $\Gamma \models \psi_k$ by reflexivity and monotonicity. Otherwise ψ_k is deduced by some inference rule. By induction hypothesis our claim holds for all premises and by Lemma 6.2. the rule we have used is **L**-normal. Hence by transitivity of $\models$ again $\Gamma \models \psi_k$ which holds for $n = k$ as well and we are done.

In showing that every subproof of degree i is justified we proceed exactly as in the proof of theorem 2.1. Every line of this subproof which is not a canceled S-line is justified by the same reasoning as in the basis, whereas previous S-formulae need additional use of the induction hypothesis that all subproofs of degree $i+1$ are justified. Let ψ_i be the first such a formula; we must show that $\Gamma \models \psi_i$, where Γ is the set of all formulae in this subproof introduced as an assumption or by reiteration. By the induction hypothesis we have $\Gamma' \models \chi_n$, where χ_n is the last line of suitable subproof of degree $i+1$ and all elements of Γ' are introduced as an assumption or by reiteration on the basis of some formulae occurring at this stage in the subproof of degree i. By Lemma 6.3. $\Gamma'' \models \psi_i$ because completion of this subproof was obtained by some proof construction rule, and all rules are normality preserving. Now, Γ'' is not necessarily a subset of Γ; it may contain formulae deduced from some elements of Γ by inference rules. Let $\Gamma'' = \Delta \cup \Sigma$, where $\Delta \subseteq \Gamma$ and Σ is the set of k formulae introduced by some inference rules. Consider the first $\chi \in \Sigma$, by Lemma 6.2. and monotonicity $\Delta \models \chi$ since the applied rule is normal. It holds also for the rest elements of Σ by this lemma, monotonicity and possibly by transitivity of $\models$. So by k applications of transitivity with respect to $\Gamma'' \models \psi_i$ and $\Delta \models \chi_i$, $i \leq k$ we obtain $\Delta \models \psi_i$ and by monotonicity we finally conclude that $\Gamma \models \psi_i$. By the same argument we consecutively demonstrate that other previous S-lines in considered subproof of degree i are also justified. Since it holds for all subproofs, we obtain:

Theorem 6.1 (Soundness of KM for L) *If* $\Gamma \vdash_L \varphi$, *then* $\Gamma \models_L \varphi$, *where* **L** *is every basic regular or normal logic*

Theorem 6.2 (Adequacy of KM for L) *Every basic regular and normal logic* **L** *is adequately characterized by KM-***L***.*

Remark 6.8 It must be said that ordinary method of proving soundness of ND systems[6] run into troubles in Fitch's format ND for modal logics. It is not clear how to transform into such a sequent a formula which is introduced by modal reiteration into a strict subderivation. Certainly if φ' is such a reiterated formula and φ its origin then it is rather not the case that $\Gamma \models \varphi$ implies $\Gamma \models \varphi'$ (e.g. in **K** where φ is some ν^i and φ' is ν). Similar problems apply to justification of rules closing strict subproofs. Interesting innovation of this strategy of soundness proof may be found in [104, 105]. Garson instead of sets of active assumptions uses sequences of them (which, by the way, better complies with ordered character of subderivations in Jaśkowski's format – cf. Section 2.4.3) and for every strict subproof introduces $\Box$ as the corresponding assumption. So for each formula in the proof we define a sequent with this formula in the succedent and the sequence of formulae and boxes in the antecedent. This trick enables to keep the proof by induction on the length of the whole derivation at the cost of small modification of an interpretation of each line. Each sequence of active assumptions (formulae

[6] We mean soundness profs which are based on the transformation of every formula into a sequent containing this formula in the succedent and the record of active assumptions in the antecedent – cf. introductory remarks in Section 2.6.

and boxes) is encoding a $\mathcal{R}$-path in a partial description of a model, where a formula from the succedent holds in the last world of this path. So we do not prove, as in classical case, that $\Gamma \models \varphi$ holds in line i provided some other statements of this sort hold in earlier lines. We rather prove that $\mathfrak{M}, w \vDash \varphi$, provided some other statements of this sort hold in earlier lines. ♣

6.4.3 Modal Assumptions

In the previous subsection we have noticed that adequate formalization of modal logics can be based on only $\Diamond$ as a primitive functor. This possibility is fully realized by Fitting [93], who presented two different ND systems for modal logics. One, called A-system, is based on $[NEC]$, and the other, I-system, is based on some (stronger) form of $[POS]$ which we name $[\bot]^F$.

$[\bot]^F$ if $\Gamma^\star, \psi \ \vdash \ \bot$, then $\Gamma, \Diamond\psi \ \vdash \ \bot$

The reason for preferring such a rule is connected with the fact that, in the context of normal logics, $[POS]$ is too weak for complete characterization, although it is sufficient in weaker logics; we will show this in Chapter 7.

This distinction is quite important for Fitting. He remarked that, with regard to proof construction, A-system is more connected with axiomatic formalizations based on (RG), whereas I-system is rather close to tableau systems. It makes I-system a better candidate as a potential proof-search tool, whereas A-system is easier to use for completeness proof of nonanalytic version. Semantic reasons for the distinction are even more important; strict derivations closed by $[NEC]$ are interpreted in a different way than those closed by $[\bot]^F$. In the former case strict derivation is a counterpart of an arbitrary chosen world (hence the name A-system per analogiam to traditional general-categorical A-statements). In the latter, it is a counterpart of some specific world in a Kripke model (hence the name I-system as particular-categorical I-statement). This informal interpretation is for Fitting a basis for the construction of respective soundness proofs for both systems.

So far we have run a different course, of mixing both systems, but full characterization apart, we can ask a question: what advantages our possibilistic approach, "possibly" offers? Previous presentation presupposed Jaśkowski's format with the apparatus of strict derivations and restricted reiterations. But the presence of modal assumptions can make at least one of these components unnecessary, because all the required information which was transported into strict derivation by (*Reit*) may be incorporated into the modal assumption which is possible due to the properties of logics. Let us display in the next lemma some new rules of $\Diamond$ introduction that are normal in some regular and normal logics (and their extensions, of course):

Lemma 6.4 (Secondary normal rules) *The following rules are normal in respective logics:*
(a) $\Diamond\varphi,\ \Box\psi\ /\ \Diamond(\varphi \wedge \psi)$ *in* **R**
(b) $\Diamond\varphi,\ \Box\psi\ /\ \Diamond(\varphi \wedge \Box\psi)$ *in* **R4**
(c) $\Diamond\varphi,\ \psi\ /\ \Diamond(\varphi \wedge \Diamond\psi)$ *in* **KB**
(d) $\Diamond\varphi,\ \Diamond\psi\ /\ \Diamond(\varphi \wedge \Diamond\psi)$ *in* **S5**

Proof is an easy exercise.

The similarity of these rules to the respective definitions of $\Gamma^\star$ is straightforward. It holds that if we have a definition of $\Gamma^\star$ for some logic, we can define a suitable introduction rule for $\Diamond$ for exactly the same logic. Moreover, if we apply for this logic a formalization with $[\bot]^F$ instead of $[NEC]$ and with suitable rule for $\Diamond$ introduction as primitive, then we can get rid of restricted (*Reit*). The only way of entering strict derivation in such a system is $[\bot]^F$ and (*Reit*) is simply forbidden here. All formulae that we need to close such a proof, and that were so far transported by (*Reit*), may be linked into conjunction with $\Diamond$, which is next turned into a modal assumption of this subderivation. But we can go even further because such a solution no longer pressuposes Jaśkowski's format or Fitch's approach in particular. We do not have special reasons to separate parts of the proof as subderivations, strict or nonstrict or whatever, hence we can combine this approach with any format of ND. Let us illustrate how to get such a system for **R** in Suppes' representation. As a counterpart of $[POS]$ we will introduce the following rule for $(\Diamond E)$:

A	i	$\Diamond\varphi$	
$\vdots$	$\vdots$	$\vdots$	
$\{k\}$	k	φ	modal assumption
$\vdots$	$\vdots$	$\vdots$	
$\{k\}$	n	ψ	
A	$n+1$	$\Diamond\psi$	$(i, k, n, \Diamond E)$

It is justified by the following rule admissible in all considered logics[7]:

If $\Gamma \vdash \Diamond\varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \Diamond\psi$.

where Γ is the set of assumptions corresponding to the assumption-set A on the schema above.

We also need a sequent version of the first of $\Diamond$ introduction rules, displayed above in Lemma 6.4, and some sequent versions of definitional rules. In order to get the extensions of **R** it is enough to provide additional suitable sequent versions of the above rules in the system and to strengthen our version of ($\Diamond E$), replacing ψ by $\bot$, especially if we want to formalize normal logics. Careful reader should not have any problems with detailed exposition, so we keep it as granted. The overall moral is that in general, $\Diamond$ seems to fit better to ND systems as a primitive functor because it is format insensitive.

On the other hand, this formally simpler and format insensitive solution, is practically more complicated. We must be able to build some π-formula (a candidate for modal assumption) before we start a strict subderivation. In practice, it is much easier to see what we need during the construction of strict derivation and apply reiteration when necessary, than to predict in advance what will be needed later. Also, in many cases, we must build rather lengthy conjunction, then decompose it, which makes a proof without use of reiteration much longer. Therefore, in what follows we prefer for practical applications rather ND with both $[NEC]$ and $[POS]$ as primitive rules, and with reiteration, although for theoretical simplicity we often describe only reduct-system based on $[NEC]$ alone (i.e. Fitting's A-system).

6.5 Standard ND for Weak Logics

There is no difficulty in extending Fitch's approach to monotonic and congruent basic logics. In order to get ND-system for **M** it is sufficient to add to KM for **CPL** two strict proof construction rules $[NEC_M]$ and $[POS_M]$:

$[NEC_M]$ if $\nu_1 \vdash \nu_2$, then $\Gamma,\ \nu_1^i \vdash \nu_2^i$
$[POS_M]$ if $\pi_1 \vdash \pi_2$, then $\Gamma,\ \pi_1^i \vdash \pi_2^i$

$[NEC_M]$ is a generalized ND counterpart (with added weakening) of

[7] It is in fact a rule which preserves normality even in monotonic logics – cf. the next section.

suitable SC rule from Section 6.1 of the form:

$$(M)\quad \frac{\psi \Rightarrow \varphi}{\Box\psi \Rightarrow \Box\varphi}$$

The only difference is that SC was formulated in $\mathbf{L}_\Box$, whereas here we use both modalities and generalized notation to cover interdefinability of $\Box$ and $\Diamond$. In fact, we could use only one of them as primitive even in full language because they are not independent (cf. the proof of their eliminability in the next Chapter) We may display their application in KM as follows:

$$\begin{array}{lll} i & \nu_1^i & \\ & \vdots & \\ j & \text{SHØW: } \nu_2^i & \\ j+1 & & \boxed{\begin{array}{c}\nu_1\\ \vdots\\ \nu_2\end{array}} \\ k & & \end{array} \qquad\qquad \begin{array}{lll} i & \pi_1^i & \\ & \vdots & \\ j & \text{SHØW: } \pi_2^i & \\ j+1 & & \boxed{\begin{array}{c}\pi_1\\ \vdots\\ \pi_2\end{array}} \\ k & & \end{array}$$

where there is no show-lines in a box and no reiteration into strict boxes.

But this time some care is needed in the formulation of these rules on the level of realization in KM. First of all, no reiteration (modal or ordinary) is admissible in strict subderivations. The only formula transported from the outer (parent) derivation is a modal assumption which is some ν in case of $[NEC_M]$ or some π in case of $[POS_M]$. Note also that the presence of modal assumption is necessary for completion of a subproof by one of these rules. On the other hand, we should not make an introduction of modal assumption an obligatory element in case of modal formula in show-line, After all, one may start a subderivation with, say S-formula $\Diamond\varphi$, but prefer to proceed with indirect proof for instance. Anyway, the rules of completion of a subproof must clearly put respective constraints (no reiteration and obligatory modal assumption). The following formulation makes it clear:

(*mod.ass.*) If ν_1^i (resp. π_1^i) is U-formula of k-degree subderivation and ν_2^i (resp. π_2^i) is S-formula opening $k+1$-degree subderivation, then we may add ν_1 (resp. π_1) as a *modal assumption* of the $k+1$-degree subderivation

(*Reit*) If φ is U-formula of k-degree subderivation, then it may be repeated as U-formula of $k+1$-degree subderivation, provided there is no modal assumption as the first line of this $k+1$-degree subderivation

$[NEC_M]$ ($[POS_M]$) Let ν_2^i (π_2^i) be the show-formula of k-degree subderivation, where the first U-formula is modal assumption, then we can close this subderivation, provided ν_2 (π_2) has appeared as U-formula in it.

One should note that these rules are in fact format-insensitive, similarly as modified possibilistic approach described in the preceding section. In fact, a formulation of ($\Diamond E$) in Suppes' format displayed therein was just a S-system counterpart of $[POS_M]$ giving adequate formalization of **M** in $\mathbf{L}_\Diamond$. Only the addition of inference rules listed in Lemma 6.4 gives us **R** and some of its regular extensions (obtained by "pumping up" a modal assumption). It shows that we may, in a similar way, use $[NEC_M]$ with modal assumption and no reiteration as a basic ND system for **M** and strengthen it to regular and normal basic logics by addition of suitable inference rules of the sort given in Lemma 6.4. In case of normal logics we must additionally admit "empty" modal assumption to cover (RG). But it is rather theoretical possibility – at the end of the preceding section we have mentioned some practical troubles connected with abandoning reiteration rule.

In case of the weakest congruent logic **E** we have in SC two-premise rule (see Section 6.1.) which in generalized form looks like this:

$$(E') \quad \frac{\nu_1 \Rightarrow \nu_2 \qquad \nu_2 \Rightarrow \nu_1}{\nu_1^i \Rightarrow \nu_2^i}$$

We have discussed in Chapter 4 how to simulate binary branching rules (taken from TS or KE) in KM. Recall that in Jaśkowski's format ND we may display the content of one branch as a new subderivation initiated by S-formula which is the main formula of the second branch. But this technique has some limitations. It is obvious that rules with more than two branches are difficult to direct simulation in such a format, but even in case of binary-branching rules we may encounter some difficulties. It is a case of the above rule schema. In Hintikka-style TS reformulation and with weakening included this rule has the following form:

$$(E') \quad \frac{\Gamma, \nu_1^i, -\nu_2^i}{\nu_1, -\nu_2 \mid -\nu_1, \nu_2}$$

The problem is that in both branches we proceed with only chosen modal subformulae from premise-set; the rest is lost. If we want to simulate such a rule in the way described in Chapter 4 we may of course define a subproof corresponding to one branch (e.g. left) as strict with $\nu_1, -\nu_2$ as modal premises and no other formulae added by reiteration. But then S-formula is $-\nu_1 \wedge \nu_2$ and if we close a subderivation it became a U-formula of outer derivation which means that all formulae from $\Gamma \cup \{\nu_1^i, -\nu_2^i\}$ are at our disposal. Such troubles may be avoided in Gentzen's tree-format by direct simulation of SC rule, where both premises correspond to strict subderivation. On the level of calculus it reads:

$[NEC_E]$ if $\nu_1 \vdash \nu_2$ and $\nu_2 \vdash \nu_1$, then $\Gamma, \nu_1^i \vdash \nu_2^i$

On the level of realization it looks like this:

$$\dfrac{\Gamma \quad \nu_1^i \quad \begin{matrix}[\nu_1] \\ \vdots \\ \nu_2\end{matrix} \quad \begin{matrix}[\nu_2] \\ \vdots \\ \nu_1\end{matrix}}{\nu_2^i}$$

In this place one weak point of Jaśkowski's format should be noted, particularly evident in KM because of the presence of show-lines. This format does not have natural devices for the realization of proof construction rules with more than one subproof involved. On the other hand, a tree-format of Gentzen is very handy in realization of such rules; any rule of the form:

if $\Gamma_1 \vdash \varphi_1, \ldots, \Gamma_n \vdash \varphi_n$, then $\Delta \vdash \psi$

is realized in this way:

$$\dfrac{\Delta \quad \begin{matrix}[\Gamma_1] \\ \vdots \\ \varphi_1\end{matrix} \quad \begin{matrix}\ldots \\ \ldots \\ \ldots\end{matrix} \quad \begin{matrix}[\Gamma_n] \\ \vdots \\ \varphi_n\end{matrix}}{\psi}$$

Similarly for S-systems, also in Suppes' format (i.e. with linear proofs). In KM the realization of such a rule would require a sequence of n boxes, each with its own premise, initiated by one S-line. It is possible to define but rather artificial. In case of $[NEC_E]$ it may be displayed as follows:

	Γ	
i	ν_1^i	
	$\vdots$	
j	SHØW: ν_2^i	
$j+1$		ν_1
		$\vdots$
k		ν_2
$k+1$		ν_2
		$\vdots$
l		ν_1

We omit cumbersome details of realization of such rules in KM. Instead we propose a simpler solution, more suitable for this kind of ND system. It is based on the following rule:

$[NEC'_E]$ if $\neg(\varphi \leftrightarrow \psi) \vdash \bot$, then $\Gamma, \Box\varphi \vdash \Box\psi$

Clearly a subproof starting with assumption $\neg(\varphi \leftrightarrow \psi)$ is strict with reiteration blocked similarly as in ND-**M**. We leave the exact formulation to the reader.

ND-Systems for **M** and **E** may be extended to stronger monotonic or congruent logics but since there are no reiteration rules for strict subproofs we cannot apply Fitting's strategy of defining sets of admissible reiteration-formulae. We are left with two strategies:

- add suitable axioms/inference rules
- transform suitable modal SC-rule into strict proof construction rule.

The first is in essence the program of Bull/Segerberg described in Section 6.2.2. The second approach makes use of characteristic SC-rules from Section 6.1.3. In case of monotonic logics, note that all rules defining basic logics, except (T), fall under one of the following schemata:

(SC-1) $\dfrac{\varphi \Rightarrow \psi}{\varphi' \Rightarrow \psi'}$ (SC-2) $\dfrac{\varphi,\ \psi \Rightarrow}{\varphi',\ \psi' \Rightarrow}$ (SC-3) $\dfrac{\Rightarrow \varphi,\ \psi}{\Rightarrow \varphi',\ \psi'}$

To get suitable proof construction rules we must first transform every SC-rule of the form (SC-2) or (SC-3) into equivalent rule of the form (SC-1), e.g.

(5) $\dfrac{\Rightarrow \Box\varphi, \psi}{\Rightarrow \Box\varphi, \Box\psi}$

is transformed into:

(5') $\dfrac{\neg\Box\varphi \Rightarrow \psi}{\neg\Box\varphi \Rightarrow \Box\psi}$

Now, to every modal SC-rule of the form:

(SC-1) $\dfrac{\varphi \Rightarrow \psi}{\varphi' \Rightarrow \psi'}$

there correspond a proof construction rule [SC-1_M] of KM which may be displayed by the following figure:

i	φ'	
	$\vdots$	
j	SHØW: ψ'	
$j+1$		φ
		$\vdots$
k		ψ

obviously, a subproof in the box is strict with no reiteration allowed.

In case of congruent logics this approach is harder to realize in KM (or any other Jaśkowski's format ND) since all SC-rules (except (T)) have two premises. So in every case we would need two consecutive strict subproofs to justify addition of some modal formulae to outer subderivation. Again, for KM it is simpler to use the following proof construction rules with one strict subproof only:

$[D_E]$:	if $\neg(\varphi \leftrightarrow \neg\psi) \vdash \bot$,	then $\Gamma, \Box\varphi \vdash \neg\Box\psi$
$[4_E]$:	if $\neg(\Box\varphi \leftrightarrow \psi) \vdash \bot$,	then $\Gamma, \Box\varphi \vdash \Box\psi$
$[B_E]$:	if $\neg(\Box\varphi \leftrightarrow \neg\psi) \vdash \bot$,	then $\Gamma, \neg\varphi \vdash \Box\psi$
$[5_E]$:	if $\neg(\Box\varphi \leftrightarrow \neg\psi) \vdash \bot$,	then $\Gamma, \neg\Box\varphi \vdash \Box\psi$

For easier comparison with root SC-rules from Section 6.1.3 we formulated all the rules in $\mathbf{L}_\Box$, but restoring them to generalized form covering both modalities is straightforward; we leave it to the reader.

There is no problem with showing adequacy of our KM system for weak basic logics. Proofs of axioms of respective logics is routine and simulation of rules (RE) and (RM) is direct with our proof construction rules – this yields completeness. Soundness proof proceeds exactly as the proof of theorem 6.1.; we must only show:

Lemma 6.5 *All proof construction rules introduced for weak basic logics are normality preserving in suitable logics.*

PROOF We show as an example that $[5_M]$ is normality preserving for **M5** and leave the other cases to a reader. Assume that $\neg\Box\varphi \models \psi$ but $\neg\Box\varphi \not\models \Box\psi$. Hence $\|\neg\Box\varphi\| \subseteq \|\psi\|$, and for some w, $w \nvDash \Box\varphi$ and $w \nvDash \Box\psi$. So $\|\varphi\| \notin \mathcal{N}(w)$ which, by condition (5) (cf. Section 5.4.5), implies that $\{w' : \|\varphi\| \notin \mathcal{N}(w')\} \in \mathcal{N}(w)$ which means that $\{w' : w' \nvDash \Box\varphi\} \in \mathcal{N}(w)$, which means that $\|\neg\Box\varphi\| \in \mathcal{N}(w)$. This claim together with $\|\neg\Box\varphi\| \subseteq \|\psi\|$ by condition (m) yields $\|\psi\| \in \mathcal{N}(w)$. But then $w \vDash \Box\psi$, a contradiction which shows that $\neg\Box\varphi \models \Box\psi$. ■

6.6 First-Order Modal Logics

The area of nonaxiomatic formalizations of **QML** in general, and ND-systems in particular, is not very rich. Fitting [93] provides some systems but with rules for quantifiers borrowed from TS. Indrzejczak [139, 141] provides a characterization of many logics on the basis of KM, whereas Garson [104, 105] offers solutions based on ND in Fitch's format but with Gentzen's rule for elimination of $\exists$ and with parameters. In what follows we briefly describe in what way several versions of **QML** described in Chapter 5 may be formalized on the basis of ND. From the plethora of ND variants described in Section 2.7 we have chosen for our considerations only two approaches: KM and KMGP (or KM' and KMGP' for free logic). This selection is justified by the fact that they seem to represent diametrically different solutions of some specific questions, and lead to different behavior of the system when modalities are added. In both cases we deal with F-systems based on Fitch's technique of strict subderivations, but it seems that proposed sets of rules are rather format insensitive and one may combine them with other types of ND systems for modal logics described in previous sections. Note that KMGP and KMGP' are equivalent to ND systems of [104, 105] in the sense of results of interaction with added modalities.[8] Our presentation follows

[8]Small differences concern the rule of $\forall$ introduction which is an inference rule in [104, 105] but proof construction rule in KMGP, the formulation of rules for free logic, and the fact that Garson does not apply KM apparatus of show-lines and boxes but Fitch's bars. Also Garson defines modal reiteration only for **K** and add axioms for extensions.

strictly the order introduced in Chapter 5 and applies only to normal logics.

To obtain KM-**QPL-L** we must simply add to KM modal rules characterizing **L**. That this solution yields logics characterized by monotonic frames is evident since we may prove CBF in KM-**QPL-K**:

1	SHØW: $\Box\forall xAx \to \forall x\Box Ax$	$[3, COND]$
2	$\Box\forall xAx$	$ass.$
3	SHØW: $\forall x\Box Ax$	$[5, UNIV]$
4	$\Box\forall xAx$	$(2, Reit.)$
5	SHØW: $\Box Ax$	$[7, NEC]$
6	$\forall xAx$	$(4, Reit(K))$
7	Ax	$(6, \forall E)$

Clearly, if background modal logic is as strong as **KB** we have, by symmetry, systems characterized by frames with (locally) constant domains, and BF is also provable:

1	SHØW: $\forall x\Box Ax \to \Box\forall xAx$	$[3, COND]$
2	$\forall x\Box Ax$	$ass.$
3	SHØW: $\Box\forall xAx$	$[5, NEC]$
4	$\Diamond\forall x\Box Ax$	$(2, Reit(B))$
5	SHØW: $\forall xAx$	$[16, RED]$
6	$\neg\forall xAx$	$ass.$
7	$\exists x\neg Ax$	$(6, \neg\forall E)$
8	$\neg Ay$	$(7, \exists E)$
9	$\Diamond\forall x\Box Ax$	$(4, Reit.)$
10	SHØW: $\Box\neg\forall x\Box Ax$	$[14, NEC]$
11	$\Diamond\neg Ay$	$(8, Reit(B))$
12	$\neg\Box Ay$	$(11, \neg\Box I)$
13	$\exists x\neg\Box Ax$	$(12, \exists I)$
14	$\neg\forall x\Box Ax$	$(13, \neg\forall I)$
15	$\neg\Diamond\forall x\Box Ax$	$(10, \neg\Diamond I)$
16	$\bot$	$(9, 15, \bot I)$

So in case of symmetric logics **QPL-L=Q1-L**; in other basic logics we must add to KM-**QPL-L** either BF or an inference rule corresponding to this axiom:

(BF) $\forall x\Box\varphi \;/\; \Box\forall x\varphi$

Hence KM corresponds exactly to axiomatic formalizations of **QML** contrary to KMGP. Surprisingly enough we are not able to formalize **QPL-L** on the basis of KMGP. One may easily prove not only CBF but also BF in KMGP-**K**, so this version of ND is as strong as **Q1** from the beginning. Here is the proof of BF in KMGP-**K**:

1	SHØW: $\forall x \Box Ax \to \Box \forall x Ax$	$[4, COND]$
2	$\forall x \Box Ax$	$ass.$
3	$\Box Aa$	$(2, \forall E)$
4	SHØW: $\Box \forall x Ax$	$[6, NEC]$
5	Aa	$(3, Reit(K))$
6	SHØW: $\forall x Ax$	$[7, UNIV]$
7	Aa	$(5, Reit.)$

Note that the application of $[UNIV]$ in line 6 is correct since a is not a parameter present in active assumptions although it is present above the line 6. Proof of CBF in KMGP-**K** is an exact copy of such a proof in KM but with a instead of x in lines 5 and 7.

Garson claims that the reason for provability of BF in his system, in contrast to axiomatic systems, is the fact that modal rules of ND are stronger than K and (RG). But the fact that we are unable to prove BF in KM (with modality weaker than B), despite having the same ND modal rules, shows that it is rather a consequence of having more flexible rules for quantifiers. In KM it is impossible to derive in line 3 $\Box Ax$ and proceed like in the proof above – because x is free above, S-formula in line 6 has no chance to be proved. So crucial for this proof is the weaker requirement that universally quantified variable (actually, corresponding parameter) should not be free in active assumptions only.

It depends on our needs whether this feature of KMGP is felt as an advantage or as a drawback. If **QPL** is treated as an artificial system, then certainly KMGP is fine, otherwise KM seems to be better. In both cases when identity is present we must add two inference rules (or just suitable axioms) yielding rigidity of terms:

(LI) $\quad \tau_1 = \tau_2 \ / \ \Box(\tau_1 = \tau_2)$
(LNI) $\quad \tau_1 \neq \tau_2 \ / \ \Box(\tau_1 \neq \tau_2)$

The remaining logics require KM' or KMGP' as a basis since they are all based on free logic.

ND-**QS-L** is obtained by a combination of KM' or KMGP' with modal rules **QPL-L** for suitable **L**. Similarly we obtain ND for **G-L** but with suitable restrictions, both $(F\forall E)$ and $(F\exists I)$ must be weakened: in KM' only variables, and in KMGP' only parameters may be values of τ. The logic **F-L** is obtainable just like **G-L** but E is counted as atomic. Finally, **Q1R-L** without identity is just like **QS-L**, otherwise we must add (LI) and (LNI); $E\tau$ is counted as atomic.

In case of **Q3-L** we proceed as in case of **G-L** but in axiomatic formulation also rules $(G\forall E), (G\forall I)$ and $(G = E)$ were needed. In this respect ND formulations behave better in general but KMGP' is the winer. Both generalized rules for $\forall$ are derivable in KMGP', whereas in KM' only the first of them. We must add $(G\forall I)$ as a primitive rule to KM'-**Q3-L** although such a solution is artificial. In contrast to KMGP', in KM' it cannot be proved because of the similar reasons as with unprovability of BF in KM and provability in KMGP. To secure completeness $(G = E)$ must be added to both ND systems for **Q3-L** but it may be simplified:

$(G = E)' \ \vdash \varphi \rightarrow x \neq \tau \ / \vdash \varphi \rightarrow \bot$

where $x \notin VF(\{\varphi, \tau\})$

It is easy to show that $(G = E)$ is provable in KM' (or KMGP') with $(G = E)'$.

Completeness of these ND systems follows easily from completeness of respective ND systems for first-order logics and for propositional modal logics. Soundness may be proved by combination of suitable proof for propositional modal logics with the result for ND systems for first-order logics from Chapter 2. Clearly in case of KM or KM' we must first make a transformation into KMG or KMG' (cf. Chapter 2) but note that the presence of $[NEC]$ or $[POS]$ does not harm to the proof of Lemma 2.3. We encourage the reader to repeat soundness proof from Section 6.4.2 for various systems enriched with rules for quantifiers.

Chapter 7

Beyond Basic Logics and Standard Systems

This Chapter has a transitional character. We consider several ways of extending standard approach presented in the last Chapter and point out their limitations. Section 7.1. is devoted to an application of standard approach in ND to other modal logics. We discuss them in order of complications they introduce into the structure of ND. Some systems are simulations of solutions from SC or TS (Sections 7.1.2 and 7.1.3), whereas other were originally introduced for ND (7.1.1 and 7.1.5). In particular, some SC and TS solutions, like that for monomodal logics of linear frames (Section 7.1.4) cannot be simulated in ND. The last subsection shows that this problem may be partly overcome if we deal with bimodal (temporal) logics of such frames.

In the next two sections we focus on the limitations of standard approach. First, we consider utility of some sequent and tableau systems as tools of proof-search; it requires some discussion of cut elimination, subformula property, confluency and similar properties. We rather avoid a discussion of proof theoretic features of the rules, or metalogical properties like completeness or applicability in proving decidability, interpolation and the like. Nevertheless, some theoretical problems connected with redundancy of standard ND, as presented in Chapter 6, are discussed in Section 7.3. In general, this presentation of drawbacks and limitations of standard approach is a natural starting point for search of a more suitable solutions.

The rest of the Chapter provides an exposition of some nonstandard deductive systems. They have in common only one feature – the basic apparatus of ordinary deductive system is substantially enriched (sometimes

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*, Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_7,

modified), so all of them may be called hybrid. We are not going to describe all the known formalizations of modal logics; it is far beyond the scope of this book. In short, we omit all these systems for which we could not recognize a clear relationship to ND-systems in their ordinary or generalized form. Criteria of selection and of grouping a material were described more strictly in Introduction. In particular, one of the most popular method of extending the basic formalism, by addition of labels, is not dealt with in this Chapter. Since using labels is the only way of constructing nonstandard ND-systems presented in this book, the technique itself deserves more attention and will be discussed thoroughly in later chapters.

Section 7.4. introduces an extension of standard approach obtained by combination with RND system from Chapter 4. We consider some variants of such a formalization for basic modal logics and their limitations. The last section is restricted to brief presentation of several variants of TS's utilizing some elements of Kripke semantics. The well known approach of Kripke, and some of his successors, is presented first because it may be seen as the first attempt to obtain a hybrid deductive system. More refined ways of combining TS with relational semantics, like higher order systems or trees with boxes, are discussed in the next subsections. Although these systems cannot be simulated in ND (at least we don't know how), they offer some formalizations of linear temporal logics and their knowledge is presupposed for understanding considerations from Chapter 9. Moreover, Kashima system, introduced in Section 7.5.2, is particularly useful as a tool of representation and comparison of several formalizations of linear logics.

7.1 Beyond Basic Normal Logics

Because in the preceding Chapter we have considered only a limited group of modal logics for which the standard approach, and in particular Fitch's ND, works quite well, one may be interested in the real scope of applicability of these techniques. Hawthorn [122] claimed that modification of reiteration works well only for these modal logics which are axiomatizable with the help of formulae of the kind $\mu\varphi \to \Box\delta\varphi$, where μ and δ are finite sequences of modal functors. It is not true however; e.g. in [140] we have provided Fitch's formalization based on the definition of Γ^* for some temporal logics axiomatized by formulae of different shape; we will introduce it in Section 7.1.5. In fact, Fitch's style ND may be reasonably extended if we take under consideration some logics that were formalized on the ground of TS or SC not only by modification of (πE), but with the help of additional special rules. In this section we will examine the extensions of Fitch's format ND for some important modal logics. Some of them are based on the simulation of rules borrowed from standard SC and TS, but there are also solutions that were originally developed for ND (7.1.1 and 7.1.5).

7.1.1 Almost Basic Logics

In Chapter 5, for some axioms defining basic logics their boxed versions were also introduced. They correspond to conditions where ordinary clauses of reflexivity e.t.c. are delayed, in the sense that in rooted models, the origin is not obliged to satisfy this condition. Some of them are important for deontic interpretation. For example, $\Box T$ may be treated as a definition of a classical conception of justice due to stoics, as was noticed by Prior.

The solution we are to present has one strange feature. It is easy to formulate some modifications on the level of realization in KM yielding correct formalization of respective logics, but it is difficult to extend a calculus by suitable rules. Note also that this solution is introduced for modal Fitch-style ND and it is rather difficult to simulate it in the standard SC or TS.[1]

For example, to obtain the result of $\Box T$ we may use the rule of delayed $\Box$ elimination:

$(D\Box E)$: we may apply $(\Box E)$ in a subderivation of k-degree, provided some outer, open subderivation is strict

Similarly, for simulation of $\Box D$ we must use a delayed version of (D). For the rest of axioms of this kind (namely $\Box 4$, $\Box B$) we must restrict suitably reiteration rules since unboxed versions are provable by $[NEC]$. It is sufficient to allow reiteration admissible for transitive and symmetric logics only in case of strict subderivations which are inside some outer strict subderivation. This kind of modification of KM introduces some complications in the proof of soundness. It is not enough to make preliminary steps in modifying proofs of lemmata 6.2 and 6.3 because there are no new rules directly stated in the calculus. We must consider in our induction on the depth of proof two cases of application of $[NEC]$ or $[POS]$. We omit the details.

[1] Although it is not impossible; it would require an addition of some global side conditions on SC proofs to the effect that some rules are correctly applied only if some other are used below them in the tree.

7.1.2 Provability Logics

There are well known SC and TS formalizations of important provability logics **G**, **K4Grz** and **Grz**. SC for these logics are due e.g. to Sambin [238]. Goré [117] considered the following rules of TS:

$$(G)\ \frac{\Box\Gamma, \neg\Box\varphi}{\Gamma, \Box\Gamma, \neg\varphi, \Box\varphi} \qquad (Grz)\ \frac{\Box\Gamma, \neg\Box\varphi}{\Gamma, \Box\Gamma, \neg\varphi, \Box(\varphi \to \Box\varphi)}$$

It is easily seen that these rules do not fall under simple schema of (πE), where only a definition of $\Gamma^\star$ regulates which logic is under consideration, since in (G) and (Grz) we have additional formulae in conclusion. In case of ND, a simulation of (G) and (Grz) is straightforward; we need only a modification of $[NEC]$:

$[NEC\text{-}G]$: if $\nu^i,\ \Gamma^\star \vdash \nu$, then $\Gamma \vdash \nu^i$

$[NEC\text{-}Grz]$: if $\Box(\nu \to \nu^i),\ \Gamma^\star \vdash \nu$, then $\Gamma\ /\ \nu^i$

In both cases $\Gamma^\star$ is defined as for **K4** (to get **Grz** we must also add (T) and simplify definition $\Gamma^\star$ as for **S4**), and the only difference with standard $[NEC]$ is the (optional) addition of modal assumption (ν^i or $\Box(\nu \to \nu^i)$) at the beginning of a strict subproof starting with S-formula ν^i. So in the definition of a derivation it is enough to add one more clause concerning optional introduction of this modal assumption, in particular for **G**:

If the last show-formula is ν^i, then we may add ν^i as a modal assumption of that derivation.

Similarly for **Grz**, but modal assumption is $\Box(\nu \to \nu^i)$. Checking that such ND-system easily simulate TS proof-trees in these logics is straightforward.

7.1.3 Logics with Branching TS Rules

Goré [117] considered also the following rules of TS:

$$(R)\ \frac{\Gamma, \neg\Box\varphi}{\Gamma, \neg\Box\varphi, \neg\varphi \mid \Gamma, \neg\Box\varphi, \Box\neg\Box\varphi, \varphi}$$

$$(S4F)\ \frac{\Delta, \Box\Gamma, \neg\Box\Theta, \neg\Box\varphi}{\Delta, \Box\Gamma, \neg\Box\Theta, \neg\Box\varphi, \Box\neg\Box\varphi \mid \Box\Gamma, \neg\Box\Theta, \neg\Box\varphi, \neg\varphi}$$

$$(S4.2)\ \frac{\Gamma, \neg\Box\varphi}{\Gamma, \neg\Box\varphi, \Box\neg\Box\varphi \mid \Gamma, \neg\Box\varphi, \Box(\neg\Box\neg\Box\varphi)}$$

These rules when added to TS-**S4**, yield the formalization of some important normal logics of epistemic or doxastic interpretation: **S4R**, **S4F**, **S4.2**.[2] The characteristic feature of these rules is the introduction of branching, in contrast to standard rules of (πE). Note that in $(S4F)$ the left conclusion repeats all parametric formulae, but in the left one we have only transfer of modal formulae. Let us call these conclusions normal and strict, respectively; in (R) and $(S4.2)$ both conclusions are normal. We have already noticed in Section 6.5 that only branching rules with both strict conclusions are difficult to simulate in Fitch's format ND – it was just the case of modal rules for congruent logics. So in these cases we may use the kind of simulation described in Chapter 4. In particular, for $(S4F)$ we must display strict conclusion as a strict subderivation, whereas normal conclusion corresponds to the outer derivation and S-formula; in the rest of rules it is an arbitrary choice. In this way we obtain three proof construction rules:

$[POS\text{-}R]$: if $\Gamma,\ \Diamond\varphi,\ \varphi\ \vdash\ \bot$, then $\Gamma,\ \Diamond\varphi\ \vdash\ \Box\Diamond\varphi \wedge \neg\varphi$

$[NEC/POS\text{-}F]$: if $\Gamma^{\star},\ \Diamond\varphi,\ \varphi\ \vdash\ \bot$, then $\Gamma,\ \Diamond\varphi\ \vdash\ \Box\Diamond\varphi$

$[NEC/POS\text{-}2]$: if $\Gamma,\ \Diamond\varphi,\ \Box\Diamond\varphi\ \vdash\ \bot$, then $\Gamma,\ \Diamond\varphi\ \vdash\ \Box\neg\Box\Diamond\varphi$

Note that $\Gamma^{\star}$ in $[NEC/POS\text{-}F]$ is defined as for **S5**. We leave the details of realization of these rules in KM noting only that in all of them we introduce an additional assumption which is obtained by elimination of $\Diamond$ in first two cases and by addition of $\Box$ to some $\Diamond\varphi$ in the last one. One may also observe that the technique we have used to simulate branching rules of Goré leads to creation of other SC variants for **S4R**, **S4F** and **S4.2** based on nonbranching rules. If we a apply standard way of simulation (rules of TS as reverses of SC rules), we get for, e.g., **S4R** the following rule:

$$\text{(SC-R)}\quad \frac{\Gamma,\ \neg\varphi \Rightarrow \Box\varphi \qquad \Gamma,\ \varphi,\ \Box\neg\Box\varphi \Rightarrow \Box\varphi}{\Gamma\ \Rightarrow \Box\varphi}$$

On the other hand, our proof construction rules lead to formulation of the following SC rules:

$$\text{(SC-R}'\text{)}\quad \frac{\Gamma,\ \neg\Box\varphi\ \Rightarrow\ \neg\varphi}{\Gamma,\ \neg\Box\varphi\ \Rightarrow\ \Box\neg\Box\varphi \wedge \neg\varphi}$$

$$\text{(SC-F}'\text{)}\quad \frac{\Gamma^{\star},\ \Diamond\varphi, \Rightarrow\ \neg\varphi}{\Gamma,\ \Diamond\varphi\ \Rightarrow\ \Box\Diamond\varphi}$$

[2]In fact, to obtain a system for **S4F** we must add to TS-**S4** both $(S4.2)$ and $(S4F)$.

$$\text{(SC-2}')\quad \frac{\Gamma,\ \Box\neg\Box\varphi\ \Rightarrow\ \Box\varphi}{\Gamma,\ \Diamond\Box\neg\Box\varphi\ \Rightarrow\ \Box\varphi}$$

These rules are provably equivalent to two-premise rules due to Goré, on the basis of standard SC with cut.

7.1.4 Logics of Linear Frames

In standard SC and TS also some logics of linear frames were formalized. One should recall here the works of Shimura and Goré: [252, 116, 117]. In both cases it is a realization of the same idea which for the first time appeared in [288]. Linearity is obtained by means of a special version of $(\Rightarrow\Box)$ in SC or π-rule in TS. For **S4.3** it has the form:

$$(\Rightarrow\Box^3)\quad \frac{\Box\Gamma\Rightarrow\Box\Delta_1,\varphi_1 \qquad \dots \qquad \Box\Gamma\Rightarrow\Box\Delta_n,\varphi_n}{\Box\Gamma\ \Rightarrow\Box\Delta}$$

where: $\Delta=\{\varphi_1,\dots,\varphi_n\},\Delta_i=\Delta-\{\varphi_i\}$

Its counterpart in Hintikka style TS, given by Goré [116], looks as follows (the meaning of Δ and Δ_i with no changes):

$$(\neg\Box E^3)\quad \frac{\Box\Gamma,\neg\Box\Delta}{\Box\Gamma,\neg\Box\Delta_1,\neg\varphi_1 \mid \dots \mid \Box\Gamma,\neg\Box\Delta_n,\neg\varphi_n}$$

[117] contains also rules for other monomodal logics of linear frames as **K4.3** or **S4.3.1**. All of them are based on the same principle which will be analyzed (and compared with other solutions) in Chapter 9. Additional properties (or their lack) lead however to some modifications; e.g. in **K4.3** the lack of reflexivity results in the following rule:

$$(\neg\Box E^{K3})\quad \frac{\Box\Gamma,\neg\Box\Delta}{S_1 \mid \dots \mid S_r}$$

where: $r=2^n-1$; $S_i=\Gamma,\Box\Gamma,\neg\Box\overline{\Delta_i},\neg\Delta_i$;
$\Delta_1,\dots,\Delta_r$ are all nonempty subsets of Δ, and $\overline{\Delta_i}=\Delta-\Delta_i$.

It is worth noting that the above solution differs considerably from the strategy of extending SC and TS applied for basic modal logics or for logics from the preceding subsection. Extensions are not obtained in a modular way by addition of new ν-rules to formalizations of **S4** or **K4** but by replacement of the basic rule of elimination for π-formula by a new unique rule of the same type.[3] Another interesting feature of these rules is varying

[3]The same remark applies to formalizations of provability logics discussed above.

number of branches. Note that in case we have only one π-formula to deal with, they are reduced to ordinary (πE) rule for **S4** or **K4**.

Presented rules, due to their many-branching character, cannot be simulated in Fitch's style ND. We have noted in Section 6.5 that for many-premise SC rules rather Gentzen's tree-format ND is more suitable. But as we already noted, in such ND we cannot realize Fitch's approach based on the use of modal reiteration rule. So, also in this case we have a problem because reiteration is essential for the solution involved in the above rules. One should conclude that in this case the possibility of easy transfer of results between SC TS, and ND is broken. It seems that there is no natural and practically applicable way of simulation of SC rules for linear logics in ND. But it does not mean that in ND such logics cannot be formalized at all; we will provide different solutions – one of them in the next subsection.

We decided to describe this approach to formalization of linear logics, despite their unfitness for ND, to complete the picture of standard approach. Introduction of this solution is also important for later comparison of several approaches to formalization of linear logics in Chapter 9. Finally, one may note one unexpected advantage of presented rules. In contrast to other standard SC (or TS) for modal logics, these systems are confluent. It is a by-product of the shape of linear rules for π-formulae, since when applying the rule we are not forced to choose one π-formula, but we are dealing independently with all of them at once.

7.1.5 Temporal Logics

One may also extend Fitch's approach to multimodal logics. In particular, we may obtain a formalization of quite extensive range of temporal logics (see Indrzejczak [140]). Interestingly enough, on the field of standard SC's or TS's there were no many proposals of this sort; Nishimura [195] provided SC for **Kt4** i **Kt4D**. We slightly change formulation of his rules putting them in a Fitting-like generalized form:

$$(\Rightarrow \nu^i)\ \ \frac{\Gamma^\star \Rightarrow \Delta^\natural, \nu}{\Gamma \Rightarrow \Delta, \nu^i} \qquad\qquad (\pi^i \Rightarrow)\ \ \frac{\pi, \Gamma^\star \Rightarrow \Delta^\natural}{\pi^i, \Gamma \Rightarrow \Delta}$$

where $\Gamma^\star = \{\nu : \nu^i \in \Gamma\} \cup \{\nu^i : \nu^i \in \Gamma\} \cup \{\pi^j : \pi \in \Gamma\}$, $\Delta^\natural = \{\pi : \pi^i \in \Delta\} \cup \{\pi^i : \pi^i \in \Delta\} \cup \{\nu^j : \nu \in \Delta\}$, and $i \neq j \in \{\mathrm{F}, \mathrm{P}\}$. For **Kt** and **KtD** it is sufficient to get rid of the middle component of the union. One should note that both i and j are present in the definitions to cover the effect of interplay of past and future.

Probably the lack of SC or TS formalizations for other temporal logics is connected with the fact that all of them are in a sense symmetrical (there is a symmetry of future and past flow of time), and already on the ground of monomodal symmetric logics there were enough troubles with finding satisfying solutions. We say more about it in the next section. But if we are interested just in making proofs, not necessarily analytic, but just easy to follow, it is not a real problem. In particular, we can obtain ND formalization of some linear temporal logics. It is in contrast to the negative result of the preceding subsection, where we stated that SC and TS rules for monomodal linear logics are not possible to simulate on the ground of ND. On the other hand, bimodal linearity is formalizable in standard ND and may be simulated by standard SC or TS. We will present a solution from [140].

For the sake of simplicity we again treat P and F as definitional shortcuts and provide a system based on $[NEC]$, but one may easily define temporal analog of $[POS]$. Here we need in fact two rules: $[NEC(H)]$ and $[NEC(G)]$, for H and G respectively, but both are covered by just one schema being an exact copy of $[NEC]$ provided for monomodal logics in Section 6.3. The definition of $\Gamma^\star$ for **Kt, KtD, Kt4, Kt4D** was stated above. A simple example of proof of **Kt**-thesis will help to understand how it works:

1	SHØW: $GHGp \to Gp$	$[3, COND]$
2	$GHGp$	*ass.*
3	SHØW: Gp	$[12, RED]$
4	$\neg Gp$	*ass.*
5	$GHGp$	$(2, Reit.)$
6	SHØW: Gp	$[11, NEC(G)]$
7	$P\neg Gp$	$(4, Reit(Kt))$
8	HGp	$(5, Reit(Kt))$
9	$\neg HGp$	(7,by def.)
10	$\bot$	$(8, 9, \bot I)$
11	p	$(10, \bot E)$
12	$\bot$	$(4, 6, \bot I)$

Note two (**Kt**-admissible) applications of temporal reiteration in lines 7 and 8; one is like in symmetric logics (with addition of possibility-like operator but of different modality than actual S-formula) whereas the second is just like in **K** – by deletion of necessity-like operator (of the same modality as S-formula).

In order to obtain a system for **Kt4.3** or **KtD4.3** we must add to $\Gamma^\star$

yet another set: $\{\nu^j : \{\nu, \nu^i, \nu^j\} \subseteq \Gamma\}$. Perhaps it will be easier to see what kind of formulae one may move by reiteration into strict subderivation, if we put it in other words. Let $H\varphi$ be the current S-formula and Γ the set of all U-formulae above, then we can put into the strict subderivation initiated by $H\varphi$, every ψ which satisfies at least one of the conditions:

- $H\psi \in \Gamma$ (usual modal reiteration)
- $\psi = H\chi$ and $\psi \in \Gamma$ (by transitivity)
- $\psi = F\chi$ and $\chi \in \Gamma$ (by symmetry of past and future)
- $\psi = G\chi$ and $\{\chi, H\chi, G\chi\} \subseteq \Gamma$ (by linearity)

Clearly, for S-formula $G\varphi$ we take duals. One may easily check that proofs of temporal axioms LF and LP are straightforward. But we may also obtain proofs of other axioms applied for linearity, and it is much easier than proving their equivalence with LP and LF in axiomatic system. Below we display a proof of one temporal form of L as an example:

1	SHØW: $G(Gp \wedge p \to q) \vee G(Gq \wedge q \to p)$	$[3, COND]$
2	$\neg G(Gp \wedge p \to q)$	*ass.*
3	SHØW: $G(Gq \wedge q \to p)$	$[5, NEC(G)]$
4	$P\neg G(Gp \wedge p \to q)$	$(2, Reit(Kt))$
5	SHØW: $Gq \wedge q \to p$	$[22, RED]$
6	$\neg(Gq \wedge q \to p)$	*ass.*
7	$Gq \wedge q$	$(6, \alpha E)$
8	$\neg p$	$(6, \alpha E)$
9	Gq	$(7, \alpha E)$
10	q	$(7, \alpha E)$
11	$Gp \wedge p \to q$	$(10, \beta I)$
12	SHØW: $H(Gp \wedge p \to q)$	$[15, NEC(H)]$
13	$\neg Gp$	$(8, Reit(Kt))$
14	$\neg(Gp \wedge p)$	$(13, \beta I)$
15	$Gp \wedge p \to q$	$(14, \beta I)$
16	SHØW: $G(Gp \wedge p \to q)$	$[18, NEC(G)]$
17	q	$(9, Reit(Kt))$
18	$Gp \wedge p \to q$	$(17, \beta I)$
19	SHØW: $\neg P\neg G(Gp \wedge p \to q)$	$[20, NEC(H)]$
20	$G(Gp \wedge p \to q)$	$(11, 12, 16, Reit(Kt4.3))$
21	$P\neg(Gp \wedge p \to q)$	$(4, Reit)$
22	$\bot$	$(19, 21, \bot I)$

Note the crucial application of reiteration for **Kt4.3** in line 20.

The above proof is sufficient for demonstration of completeness of this formalization for linear temporal logics since proofs of other axioms is routine and two forms of temporal (RG) are simulated exactly as in monomodal case. The proof of soundness is an exact copy of a proof from Section 6.4, we need only to show that both versions of $[NEC]$ with Γ^* defined as above are normality preserving with respect to linear frames. We leave it to the reader or advice to consult [140].

7.2 Limitations of Standard Approach

The overall picture of the results from the preceding, and this Chapter, may look quite nice. After all, surveys like that of Goré [117](or even much earlier Zeman's work [288]) show that a lot of logics can be formalized with the help of tableaux or sequent calculi. So, despite the discussed limitation (e.g. transfer of multibranching rules into the context of Jaśkowski's format ND), one may think that standard approach is really extensive and provides a uniform syntactic frame for characterization of many logics. But it is easily seen that many of the systems are based on rather artificial solutions that are incompatible with the most natural requirements concerning practically useful tableau or sequent calculi.

Generally, the schema for construction of new SC, TS or ND (in Fitch's style) system for many logics is similar; we keep a constant rule $(\Rightarrow \Box)$ (or (πE) or $[NEC]$) and modify only $\Gamma^\star$. In some cases it looks pretty, but a definition of this set in many cases is not satisfactory and seems to be an ad hoc construction. Moreover, for many logics (e.g. provability and linear logics) we must either replace old rules by the new ones or to introduce additional rules of a different shape. So the architecture of the system is somewhat broken and one may conclude that these standard systems which may be seen as simple and natural, are rather formalizations of some particular logics, but their generalizations are often not easy to provide. Standard approach in TS and ND is also format sensitive since, in TS we presuppose Hintikka format (sets as data structures), in ND we must have subproofs – at least strict ones.

So it must be said that standard approach to formalization of modal logics is hardly recognized as a uniform syntactic frame comparable in scope to successful semantic framework provided by relational models. It is particularly evident if we search for a system having satisfactory properties.

Already Sambin and Valentini [239] have noted in the context of modal SC:

> It is usually not difficult to choose suitable rules for each modal logic if one is content with completeness of rules. The real problem however is to find a set of rules also satisfying the subformula property.

This remark applies equally well to other features usually required in case of SC or TS. Here, we are concerned neither with philosophical questions nor with theoretical investigations from proof theory, but rather pay attention to practical matters connected with proof search. Hence we will discuss only some chosen properties and address an interested reader to other sources of information. One may find an extensive discussion of these matters in the context of philosophy of meaning involved in the construction of ordinary SC in Poggiolesi [214]. Exhaustive criticism of ordinary modal SC from the point of view of proof theory is provided by Wansing [280]. He shows that modal rules in standard SC usually lack many structural properties, that are actual source of success of SC as a tool of proof analysis. But one should note that at least one point from Wansing's list of complaints may be partly dispelled. SC for congruent and monotonic basic logics presented in Chapter 6 is modular, in contrast to standard SC systems for normal logics. Moreover, one may obtain modular regular and normal SC's on the basis of weaker systems, just by addition of rule (C) or $(C\text{-}3)$ (cf. Section 6.1.3). But it is doubtful if such a solution is better for practical applications.

One of the central issue is certainly cut elimination. We recall the results for basic logics:

Theorem 7.1 *Cut elimination holds for the following SC's for basic logics:*

- *in the class of normal logics:* **K, D, T, K4, D4, S4, K45, KD45***; the same holds for discussed regular counterparts*
- *in the class of monotonic logics:* **M, MD, MT, M4, M5, MT4, MD4, M45, MD45***; the same holds for respective monotonic logics with (N)*
- *in the class of congruent logics:* **E, ED, ET, E5, ET4***; the same holds for respective congruent logics with (N) except* **END**

In the first group most of the results are due to Ohnishi/Matsumoto [197, 197] and Zeman [288]; the last two come from Shvarts [253]. Results for weak logics are provided by Indrzejczak in [152]. It is quite surprising that both **E5** and **M5** (and their N-counterparts) have cut-free formalization in contrast to **K5**. On the other hand, neither **ED5** nor **MD5** (and their N-counterparts) have cut-free formalization, similarly as **KD5**. In general, cut elimination fails for all B-logics, but in case of congruent logics we have a similar situation with 4-logics (the exception is **ET4** and **ENT4**). It is easy to note that congruent logics represent very bad behaviour with respect to cut elimination, in contrast to monotonic logics, that are even better than normal logics. But one should note that standard SC for **MD4** and **MD45** as presented in Section 6.1.3 is not suitable for that. To prove cut elimination one needs additional two rules:

$$(D5)\ \frac{\varphi \Rightarrow \Box\psi}{\Box\varphi \Rightarrow \Box\psi} \qquad\qquad (D4)\ \frac{\varphi,\ \Box\psi \Rightarrow}{\Box\varphi,\ \Box\psi \Rightarrow}$$

Thus cut-free SC'-**MD4** is SC-**MD4** with $(D4)$, whereas cut-free SC'-**MD45** is SC-**MD45** with $(D5)$. Both rules are provable in generic SC's, but their addition is necessary for proof of cut elimination. Many other normal logics were proved to be cut-free as well, in particular all SC (TS) systems presented in Sections 7.1.2, 7.1.3, and 7.1.4.

Although in many cases cut is not eliminable, it is not necessary a disaster. We have already mentioned that admissibility of cut is neither sufficient nor necessary for obtaining an analytic formalization. Takano [269] proposed SC formalizations with analytic cut for all normal B-logics, and (in [270]) for **K5** and **KD5** with cut restricted to applications with modified subformula property. Rautenberg [229] and Goré [117] proposed TS's for these logics with additional rules simulating special applications of cut and satisfying the "superformula"-property; details may be found in [117]. All these solutions are meant to overcome problems generated by semantics of symmetric and Euclidean logics. From the semantical point of view one must have a possibility to come back, from one world to its predecessor, but standard SC or TS is a system admitting only forward moves. Hence special applications of cut or additional rules are devised to keep or to reintroduce some essential information from the predecessor which may be lost otherwise. One should note that we have the same situation in bimodal temporal logics, and probably this is the main reason that these logics were rarely formalized as standard SC's or TS'.

So analycity is somewhat saved, but even in case of cut-free systems we have a defect which, from the point of view of automation, is rather troublesome. Let us focus on some characteristic feature of modal rules considered above (i.e. $(\Rightarrow \Box)$, $(\Diamond \Rightarrow)$ and the like). In general, such rules lead to a loss of a part of information and/or a modification of the rest. It is due to the fact that there is a "narrow gate" leading from the conclusion to the premise(s). Only some parametric formulae, and usually after deleting of modal functors, are released. In case of congruent and monotonic logic this "gate" is even narrower – we admit exactly one parametric formula. Fitting called systems with such rules "destructive", since some information is lost or destroyed. Such rules belong to the wider category of rules that are not pure (in Avron's terminology), where purity means that if from a sequent (or sequents) S a sequent S' is derivable, then it holds also if some formulae are added in the antecedent or in the succedent of premise(s) and conclusion. Clearly, every destructive rule is not pure in this sense.

These features of modal SC rules have an important impact on some properties of the calculi. First of all, invertibility of all rules is lost, because either weakening as a rule is necessary (if we have rules like $(\Rightarrow \Box)$) or its effect is implicitly given in formulation of rules like generalized Fitting's $(\Rightarrow \Box')$. Also, considered systems are in general not confluent, and we may be forced to backtracking during proof search process. In other words, if we finish some branch with nonaxiomatic but atomic leaf, it does not mean that the root-sequent is nonprovable, perhaps some other routes may lead to success. It may be conveniently illustrated with the help of typical (πE) application. Assume that we are at the stage of proof-search where no other rule is applicable, but we have two π-formulae in the current set. We may apply the rule only to one of them, the other is erased because usually it does not belong to $\Gamma^\star$. If we later obtain finished and open branch, it is not enough to build a falsifying model for an input-formula since, in such a model, every π-formula must be satisfied at some point and we do not know if our abandoned π-formula really is.

As an example we may consider a thesis of **K** $\Diamond p \wedge \Diamond(q \rightarrow r) \rightarrow (\Box q \rightarrow \Diamond r)$. In tableau proof-tree, after negation of the formula and some standard transformations we obtain a set $\Gamma = \{\Diamond p, \Box q, \Diamond(q \rightarrow r), \neg\Diamond r\}$. If we apply (πE) to the first π-formula from the left we will get a set $\{p, q, \neg r\}$, and the branch finishes as open. But if this rule will be applied to the second π-formula we obtain $\{q \rightarrow r, q, \neg r\}$, and this set is not satisfiable – since after application of β-rule we immediately obtain two closed branches. So in one case we do not obtain a proof, whereas in the second we do.

To avoid such problems we must have a possibility to go back to the last stage, where (πE) was applied and try again with it. Several solutions were proposed to the effect that backtracking is somewhat formally secured. For example, Goré proposed modified rule of the form:

$$(\pi E') \quad \frac{\Gamma,\ \pi^i}{\Gamma^\star,\ \pi \ \parallel\ \Gamma}$$

Branching displayed with $\parallel$ has a different character than in β-rules or cut; there we have disjunctive branching (premise-set is satisfiable, if at least one conclusion-set is), here it is conjunctive (both conclusion-sets must be satisfied).[4] Other solution, from [142], consists in recursive embedding of the definition of proof tree in the definition of proof-search tree. The simplest solution does not affect a calculus but introduces some meta-rule into definition of proof-search procedure. It was applied e.g. by Sambin [238]); also a decision algorithm for basic monotonic logics sketched in [152] is based on such a meta-rule of Subtree Generation (SG).

$$(SG) \quad \frac{\mathcal{S}_L}{\Gamma,\ \Box\varphi_1, ..., \Box\varphi_l \Rightarrow \Box\psi_1, ..., \Box\psi_k,\ \Delta}$$

where, $l + k > 0, \Gamma, \Delta$ are atomic formulae, and $\mathcal{S}_L$ is not a sequent but the set of sequents called subproof generators. These are defined for each logic having cut-free formalization from characteristic sets of sequents:

$$\mathcal{S}_M = \{\varphi_i \Rightarrow \psi_j \ :\ i \leq l,\ j \leq k\}$$

$$\mathcal{S}_4 = \{\Box\varphi_i \Rightarrow \psi_j \ :\ i \leq l,\ j \leq k\}$$

$$\mathcal{S}_5 = \{\Rightarrow \Box\psi_i,\ \psi_j \ :\ i \leq k,\ j \leq k\}$$

$$\mathcal{S}_D = \{\varphi_i,\ \varphi_j \Rightarrow :\ i \leq l,\ j \leq l\}$$

$$\mathcal{S}_{D5} = \{\varphi_i \Rightarrow \Box\psi_j \ :\ i \leq l,\ j \leq k\}$$

$$\mathcal{S}_{D4} = \{\varphi_i,\ \Box\varphi_j \Rightarrow :\ i \leq l,\ j \leq l\}$$

These combine into the following sets:

[4] It seems that the first who introduced trees with two types of branching was Beth [28] in TS for intuitionistic logic.

Logic	$\mathcal{S}_L$
M, MT	$\mathcal{S}_M$
MD	$\mathcal{S}_M \cup \mathcal{S}_D$
M4, MT4	$\mathcal{S}_M \cup \mathcal{S}_4$
M5	$\mathcal{S}_M \cup \mathcal{S}_5$
MD4	$\mathcal{S}_M \cup \mathcal{S}_D \cup \mathcal{S}_4 \cup \mathcal{S}_{D4}$
M45	$\mathcal{S}_M \cup \mathcal{S}_4 \cup \mathcal{S}_5$
MD45	$\mathcal{S}_M \cup \mathcal{S}_D \cup \mathcal{S}_4 \cup \mathcal{S}_5 \cup \mathcal{S}_{D4} \cup \mathcal{S}_{D5}$

But one should recall that at least Shimura/Zeman SC for **S4.3** (as well as Goré variants for other linear logics) is confluent! Its characteristic rule creates as many premises as we have $\Box$-formulae in the succedent of a conclusion. In this respect it is apparently similar to our meta-rule (SG), but in this case it is the same kind of branching like in β-rules. SC rules for linear logics are still destructive and impure, due to side conditions concerning the antecedent of conclusion-sequent, but nevertheless confluency is saved.

There is one more problem connected with proof-search procedures defined for standard calculi. The risk of loop-generation leading to infinite branches. It is connected with the fact that some of the rules involved, e.g. for transitive logics do not satisfy subformula property in the strict sense (some formulae are rewritten to premise). It should be noticed that in this respect monotonic logics also reveal better behaviour than normal logics. The procedure based on the application of (SG) described above, leads to termination even in case of such 4-logics like **M4**, **MT4**, **MN4** and **MNT4**. It is well known fact that in procedures defined on SC or TS for normal (or regular) 4-logics we can generate infinite branches due to duplication of $\Box$-formulae. But in case of monotonic logics this problem disappears since no such formula is transported at the same time with a box and without it to the premise. For example, instead of $\Box\varphi,\ \varphi \Rightarrow\ \psi$ from some $\Box\varphi \Rightarrow \Box\psi$ in SC for **K4**, in **M4** we may only obtain either $\Box\varphi \Rightarrow \psi$ by (4), or $\varphi \Rightarrow \psi$ by (M) which blocks generation of loops. Nevertheless, the problem of possibly infinite branches reappears in case of these calculi that contain (5) or $(D4)$ (namely: **M5**, **M45**, **MD4**, **MD45**, **MN5**, **MN45**, **MND4** and **MND45**).[5] It is possible because two elements in premises of (5) and $(D4)$ may be the same formula. In such a situation from $\Box\varphi, \Box\varphi \Rightarrow$ ($= \Box\varphi \Rightarrow$) we may deduce by $(D4)$ $\Box\varphi,\ \varphi \Rightarrow$ and then do it again. Fortunately, such loops are extremely easy to detect and we can obtain termination anyway.

[5] In [152] it was noticed only for SC with $(D4)$.

All these problems must have reappear if we would like to use Fitch-style ND for modal logics as a tool for proof search, and practical decision procedure. This kind of formalization, similarly as standard ND for classical logics, is not universal, but it is harder to obtain an analytic version of ND for modal logics, than for **CPL**. Clearly, for ND the lack of cut-elimination for some logics is not a problem, as we have shown in Chapter 4, but still other features may cause problems. For example, how to deal with non-confluency? It would be necessary to have some devices for flagging some strict subderivation as failed (if, e.g., the set of respective U-formulae is downward saturated but consistent), perhaps boxing without cancelation of prefix "SHOW:"? In this way we could mark this part of a proof as unsuccessful, and try different choices. So it seems that due to eventual complications in the definition of a derivation, standard ND are not very good candidates for providing analytic systems. Implementation of such a system would require even more complicated work. Hence we may certainly build more natural, shorter and simpler proofs in ND than in Hilbert systems but the possibility of using such systems as decision procedures or in automated deduction, is an open question. We are afraid that the prospects are not very promising, and labelled systems, that will be introduced in the subsequent chapters, are much better prepared for such tasks.

7.3 Redundancy of Standard Systems

Before we go on with some alternatives to standard systems we focus on some theoretical problems connected with redundancy of Fitch's format ND. It may be skipped by readers interested mainly in practical aspects of its application. We have already mentioned that Fitting's style formulation is redundant, but even the system of Fitch in the original formulation (cf. Section 6.4.1) is redundant. First, the primitive introduction/elimination rules for $\Box$ and $\Diamond$ are not independent. Second, it is interesting whether definitional rules are necessary for completeness.

7.3.1 Admissibility of Proof Construction Rules

We start with the first problem. For Fitch's rules ($\Box E$) and ($\Diamond I$) the situation is clear. The first is just (T) and the second its contrapositive modulo interdefinability of $\Diamond$ and $\Box$. They are normal rules only in logics axiomatizable by T so for weaker logics we do not have in general rules of this kind. As we noted, some authors prefer to treat basic modal reiteration as a kind of □ elimination rule. Another possibility is to use some generalized forms of □ elimination as defined in Remark 6.5; we may easily define their counterparts for ◇ introduction. Anyway, all these rules are interderivable; in particular (T) is provable in the system with $(\neg\Box I)$, and $(\Diamond I)$ is provable in the system with $(\neg\Diamond E)$.

What is more interesting, also $[NEC]$ and $[POS]$ are in fact mutually dependent, although not, in general, interderivable – for that we need to replace Fitch's $[POS]$ with Fitting's $[\bot]^F$. We have already mentioned this fact, when discussing Fitting's two systems; one may easily prove all necessary axioms and rules for himself either in A- or in I-system. But it is of some interest in itself to show that in the full system they are eliminable. For $[NEC]$ and $[POS]$ we can prove the following lemma:

Lemma 7.1 (Admissibility of rules)

1. $[POS]$ *is admissible in* **CPL**$+[NEC]+(\neg\Diamond E)+(\neg\Diamond I)$

2. *Regular-*$[NEC]$ *is admissible in* **CPL**$+[POS]+(\neg\Box E)+(\neg\Box I)$

where Regular-$[NEC]$ means $[NEC]$ with proviso for regular logics (cf. Section 6.3).

Proof it is provided by the following schemata of elimination. Every part of a proof containing an application of $[POS]$ of the shape:

$$
\begin{array}{lll}
 & \vdots & \\
i-j & \Gamma & \\
j+1 & \Diamond\psi & \\
 & \vdots & \\
k & \text{SHØW: } \Diamond\varphi & [n, POS] \\
k+1 & \quad \boxed{\psi} & mod.ass(j+1) \\
(k+2)-l & \quad \boxed{\Gamma^\star} & (i-j, Reit.) \\
 & \quad \boxed{\vdots} & \\
n & \quad \boxed{\varphi} &
\end{array}
$$

may be recursively (starting with the innermost application of $[POS]$) replaced by the following:

	$\vdots$	
$i-j$	Γ	
$j+1$	$\Diamond\psi$	
	$\vdots$	
k	SHØW: $\Diamond\varphi$	$[o+4, RED]$
$k+1$	$\neg\Diamond\varphi$	*ass.*
$k+2$	$\Box\neg\varphi$	$(k+1, \neg\Diamond E)$
$k+3$	$\Diamond\psi$	$(j+1, Reit.)$
$(k+4)-l$	Γ	$(i-j, Reit.)$
$l+1$	SHØW: $\Box\neg\psi$	$[m+2, NEC]$
$(l+2)-m$	$\Gamma^\star$	$(k+4-l, Reit.)$
$m+1$	$\neg\varphi$	$(k+2, Reit.)$
$m+2$	SHØW: $\neg\psi$	$[o+2, RED]$
$m+3$	ψ	*ass.*
$(m+4)-n$	$\Gamma^\star$	$(l+2-m, Reit.)$
	$\vdots$	
o	φ	by assumed proof
$o+1$	$\neg\varphi$	$(m+1, Reit.)$
$o+2$	$\bot$	$(o, o+1, \bot I)$
$o+3$	$\neg\Diamond\psi$	$(l+1, \neg\Diamond I)$
$o+4$	$\bot$	$(k+3, o+3, \bot I)$

similarly, every application of $[NEC]$ admissible in every regular logic, of the shape:

	$\vdots$	
$i-j$	Γ	
$j+1$	$\Box\psi$	
	$\vdots$	
k	SHØW: $\Box\varphi$	$[n, NEC]$
$(k+1)-l$	$\Gamma^\star$	$(i-(j+1), Reit.)$
	$\vdots$	
n	φ	

may be replaced by:

	$\vdots$				
$i - j$	Γ				
$j + 1$	$\Box\psi$				
	$\vdots$				
k	SHØW: $\Box\varphi$				$[n + 4, RED]$
$k + 1$		$\neg\Box\varphi$			*ass.*
$k + 2$		$\Diamond\neg\varphi$			$(k + 1, \neg\Box E)$
$(k + 3) - l$		Γ			$(i - j, Reit.)$
$l + 1$		SHØW: $\Diamond\neg\psi$			$[m + 1, POS]$
$l + 2$			$\neg\varphi$		$mod.ass(k + 2)$
$(l + 3) - m$			$\Gamma^\star$		$(k + 3 - l, Reit.)$
$m + 1$			SHØW: $\neg\psi$		$[o + 1, RED]$
$m + 2$				ψ	*ass.*
$m + 3$				$\neg\varphi$	$(l + 2, Reit.)$
$(m + 4) - n$				$\Gamma^\star$	$(l + 3 - m, Reit.)$
				$\vdots$	
o				φ	by assumed proof
$o + 1$				$\bot$	$(m + 3, o, \bot I)$
$o + 2$		$\neg\Box\psi$			$(l + 1, \neg\Box I)$
$o + 3$		$\Box\psi$			$(j + 1, Reit.)$
$n + 4$		$\bot$			$(o + 2, o + 3, \bot I)$

One may easily check that these proofs of elimination hold also for monotonic versions of respective proof construction rules, namely for $[NEC_M]$ and $[POS_M]$. In the first case it is enough to delete all occurrences of $\Gamma^\star$ in both schemata and change the justification of line $m + 1$ (in the second schema) from reiteration into modal assumption. In the second case we must in the schema of eliminated subproof change $\Gamma^\star$ into ψ justified as modal assumption and delete occurrences of $\Gamma^\star$ in the second schema. ■

It is evident that $[NEC]$ and $[POS]$ are equivalent in regular logics (and monotonic as well); $[POS]$ in itself is too weak to capture in ND the effect of (RG) – we must have some $\Box$-formula in outer proof (here it is $\Box\psi$ in line $j + 1$ which is essential to close the replacing proof). But we can use a variant (in fact two variants) of $[POS]$ which is sufficiently strong for keeping equivalence also in normal logics. We have already (in

Section 6.4.3) introduced the rule $[\bot]^F$ due to Fitting [93] (it is displayed below for convenience); earlier Wisdome [282] proposed a rule $[\bot]^W$:

$[\bot]^W$ if $\Gamma^\star, \psi \vdash \bot$, then $\Gamma, \Diamond\psi \vdash \varphi$
$[\bot]^F$ if $\Gamma^\star, \psi \vdash \bot$, then $\Gamma, \Diamond\psi \vdash \bot$

Both variants are equivalent; $[\bot]^W$ follows from $[\bot]^F$ in **CPL**, and the latter is a particular form of $[\bot]^W$, hence we will simply use the name $[\bot]$ for any of them. The equivalence of logics formalized with $[NEC]$ or with $[\bot]$ is stated in the next lemma:

Lemma 7.2 (Admissibility of rules)

1. $[\bot]$ *is admissible in* **CPL**+$[NEC] + (\neg\Diamond I)$
2. $[NEC]$ *is admissible in* **CPL**+$[\bot] + (\neg\Box E)$

Proof by similar schemata as in the previous lemma.

So there is no need to modify $[POS]$ in regular logics or $[POS_M]$ in monotonic logics to obtain systems equivalent with formalizations based on $[NEC]$ (or $[NEC_M]$) but for normal logics we should rather use $[\bot]$ instead of $[POS]$.

Remark 7.1 Note that one may keep a uniform formalization of monotonic, regular and normal logics based on $[POS]$ but at the expense of addition of one more inference rule to every normal logic: $(\Diamond\bot)$ $\Diamond\bot$ / $\bot$. This rule is not normal in weaker logics but in normal logics it saves completeness of I-formulations because we may prove $\Diamond\bot$ by $[POS]$ with $\varphi = \bot$ and then apply $(\Diamond\bot)$ to obtain the effect of $[\bot]$. ♣

Remark 7.2 In case of normal logic **D** and its extensions there is no need to add modal assumption in the strict derivation initiated by $[POS]$, because in **D** we have admissible rule:

if $\Gamma^\star \vdash \psi$, then $\Gamma \vdash \Diamond\psi$.

Hence, it is possible in all extensions of **D** to simplify $[POS]$, making an introduction of modal assumption an optional element, similarly as with assumptions for $[COND]$ and $[RED]$. Moreover, it enables to obtain an adequate formalization for **D**, without any special rule of inference such as (D), which is now simply derivable. If we admit such a simplification in the definition of $[POS]$ on the ground of regular logics we must add similar proviso as in the case of the definition of $[NEC]$ in them, namely that in the outer derivation at least one U-formula should be ν-formula. ♣

7.3.2 Interdefinability Problem

The second question we want to consider is whether definitional rules are necessary. One may check that although the system is redundant, the rules for introduction and elimination of $\Box$ and $\Diamond$ are not enough for completeness. We cannot get rid of all definitional rules, but we may use only some of them because they are interderivable in the following way:

Lemma 7.3 (Derivability of rules)

1. $(\neg\Diamond E)$ *is derivable in* **CPL**$+[POS_M]+(\neg\Box E)$
2. $(\neg\Diamond I)$ *is derivable in* **CPL**$+[POS_M]+(\neg\Box I)$
3. $(\neg\Box E)$ *is derivable in* **CPL**$+[NEC_M]+(\neg\Diamond E)$
4. $(\neg\Box I)$ *is derivable in* **CPL**$+[NEC_M]+(\neg\Diamond I)$
5. $(\neg\Box I)$ *and* $(\neg\Diamond I)$ *are derivable in* **CPL**$+[\bot]$

PROOF we will provide schemata of derivations justifying 1., 4. and one of 5.; for remaining cases proofs are analogous.

1	$\neg\Diamond\varphi$	*premise*
2	SHØW: $\Box\neg\varphi$	$[9, RED]$
3	$\neg\Box\neg\varphi$	*ass.*
4	$\Diamond\neg\neg\varphi$	$(3, \neg\Box E)$
5	SHØW: $\Diamond\varphi$	$[7, POS_M]$
6	$\neg\neg\varphi$	$mod.ass(4)$
7	φ	$(6, \neg\neg)$
8	$\neg\Diamond\varphi$	$(1, Reit.)$
9	$\bot$	$(5, 8, \bot I)$

1	$\Diamond\neg\varphi$	*premise*
2	SHØW: $\neg\Box\varphi$	$[9, RED]$
3	$\Box\varphi$	*ass.*
4	$\Diamond\neg\varphi$	$(1, Reit.)$
5	SHØW: $\Box\neg\neg\varphi$	$[7, NEC_M]$
6	φ	$mod.ass(3)$
7	$\neg\neg\varphi$	$(6, \neg\neg)$
8	$\neg\Diamond\neg\varphi$	$(5, \neg\Diamond I)$
9	$\bot$	$(4, 8, \bot I)$

1	$\Diamond\neg\varphi$	*premise*
2	SHØW: $\neg\Box\varphi$	$[5, RED]$
3	$\Box\varphi$	*ass.*
4	$\Diamond\neg\varphi$	$(1, Reit.)$
5	SHØW: $\bot$	$[8, \bot]$
6	$\neg\varphi$	$mod.ass(4)$
7	φ	$(3, Reit(R))$
8	$\bot$	$(6, 7, \bot I))$

Note that although we provided proofs of 1. and 4. in KM-**M** they are correct also for regular and normal logics. We must simply change in both proof-schemata a justification of some lines. In the former we use $[POS]$ in line 5, in the latter $[NEC]$ in line 5 and line 6 is not a modal assumption but an application of $(Reit(K))$ or $(Reit(R))$ from line 3. On the other hand point 5. of Lemma 7.3. holds only for regular and normal logics because the use of modal reiteration is essential. ■

The following observation is a simple consequence of this lemma:

- KM for any monotonic, regular, normal logic based on $[NEC]$ requires $(\neg\Diamond I)$ and $(\neg\Diamond E)$
- KM for any monotonic, regular, normal logic based on $[POS]$ requires $(\neg\Box I)$ and $(\neg\Box E)$
- KM for any monotonic, regular, normal logic based on $[NEC]$ and $[POS]$ requires $(\neg\Box I)$ and $(\neg\Diamond E)$, or $(\neg\Box E)$ and $(\neg\Diamond I)$, or $(\neg\Box I)$ and $(\neg\Diamond I)$, or $(\neg\Box E)$ and $(\neg\Diamond E)$

- KM for any regular, normal logic based on [$\bot$] requires ($\neg\Diamond E$) and ($\neg\Box E$)

- KM for any regular, normal logic based on [NEC] and [$\bot$] requires ($\neg\Diamond E$)

The last point is worth noting. Using both [$\bot$] and [NEC] as primitive rules of the system makes only one definitional rule necessary. One may check that elimination of this one rule is impossible however. It was proved as derivable by [POS] (see point 1. in the proof above) so one may think that it should be provable by stronger rule as well. But if we try to eliminate [POS] in favor of [$\bot$] in the way shown in the preceding subsection, we find that ($\neg\Diamond E$) must be used in replacing proof schema. So either we must add this rule or [POS] to the basis. This shows some inherent asymmetry in the construction of the analyzed rules, similarly as in standard SC modal rules. It seems that the problem is solved in a satisfying way only in deductive systems of nonstandard character, where radical enrichment of the basic structural tools makes possible greater flexibility e.g. like in display logic (see [280]) or in multisequential sequent calculus (see [143]). In standard ND these rules are (trivially) derivable only if we adopt a generalized terminology of Fitting, at least with respect to definition of formulae admissible for modal reiteration. Otherwise at least one definitional rule is necessary.

It seems that if we prefer a formulation with $\Diamond$ as primitive (and perhaps the only modal functor), [$\bot$] is a better choice. It is not only stronger than [POS] and, in consequence, more universal (covers also normal logics as a sufficient rule, whereas [POS] is adequate only for weaker logics), but also more economical if [NEC] is present. The drawback of such a system is that [$\bot$] is not a good representative of a modal rule, which is particularly evident in KM, where it looks like a kind of indirect proof. Only the rule of introduction of modal assumption has some flavor of a modal rule but it is independent of [$\bot$], at least on the level of realization. On the other hand, in the system like AND1, where (analytic) [RED] is the only proof construction rule, the addition of [$\bot$] fits nicely in contrast to [NEC]. Such a version of modal AND1 with [$\bot$] as the only additional proof construction rule would be better prepared to simulate proof-search strategies from modal tableau systems. Still, the problems connected with nonconfluency and backtracking, discussed in the preceding section, remain.

Remark 7.3 One should notice that, at least with respect to some stronger normal logics, like **S4** or **S5**, we may obtain a formalization based on some variant of $[POS]$ instead of $[\bot]$. It may be of some interest for those who look for a rule better suited as a representative of the basic rule for $\Diamond$. System of this sort for **S5** is present in Hazen [123]; slight modifications make possible to obtain a system for **S4**. In a system for **S5**, we admit that reiteration in strict derivations is limited to m-formulae, $[NEC]$ is in the standard form and $[POS]$ is based on the following rule:

if $\Gamma^\star, \psi \ / \ \varphi$, then $\Gamma, \Diamond\psi \ / \ \varphi$, where φ is any m-formula

A variant for **S4** is similar but reiteration in case of $[NEC]$ and $[POS]$ is restricted to ν-formulae and φ in the schema of $[POS]$ must be any π-formula. Definitional rules are dispensable in both systems if we use generalized notation of Fitting. If we restrict, e.g. in **S4**, reiteration to $\Box$-formulae and φ in the schema of new $[POS]$ to $\Diamond$-formulae, then some definitional rules are necessary. It is left to the reader to establish which ones. ♣

7.4 RND for Modal Logics

In the search for alternative, and perhaps more satisfying, systems we first consider RND introduced in Chapter 4. We have already mentioned that among a few approaches to modal logics in ND, the Fitch's format seems to be the most extensive, so we consider only a combination of RND with this approach. Fortunately we can refine the Fitch's technique to comply with RND – we sketch such a modification in this section.

7.4.1 RND Systems for M, R and K

We start with a definition of a system adequate for the weakest logic in each group (except congruent logics), then we discuss how to obtain extensions. Clearly we must introduce the category of strict derivations and to block unrestricted transfer of formulae into them. Technically it is executed as in ordinary ND, by reiteration rule that specifies what kind of formulae are admissible for transfer. In the formulation of RND from Chapter 4 we did not refer explicitly to reiteration rule because we admitted, in the definition of a derivation, that we can use any U-formula in the current subderivation. In Fitch's approach it is necessary to introduce reiteration explicitly. So in general, modal RND follows quite closely ordinary Fitch's ND. There is one important cost on the side of simplicity of RND – one proof construction rule is not enough to govern two types of derivation. Except $[SUB]$ we must add specific modal proof construction rules. Let us consider two additional proof construction rules being clausal variants of modal rules $[NEC]$ and $[POS]$. First we define simpler variants suitable for RND-**M**:

$[CNEC_M]$ if $\Upsilon_1 \vdash \nu$, then $\Gamma, \Upsilon_1^i \vdash \Gamma, \Upsilon_2^i$, provided $\nu \in \Upsilon_2$
$[CPOS_M]$ if $\Pi_1 \vdash \pi$, then $\Gamma, \Pi_1^i \vdash \Gamma, \Pi_2^i$, provided $\pi \in \Pi_2$

Remember (cf. Chapter 5) that Υ^i denotes any clause containing only ν-formulae (of i-modality in multimodal case), whereas Π^i – only π-formulae. Υ and Π are sets obtained by deletion of the first modal operator in every element of Υ^i, Π^i. Γ, Υ^i is a clause, where except ν-formulae from Υ^i we have a (possibly empty) set of other formulae in Γ, similarly for Γ, Π^i.

The completion of these rules in RND[6] may be shown on diagrams in the following way:

i	Γ, Υ_1^i		i	Γ, Π_1^i	
	$\vdots$			$\vdots$	
j	SHØW: Γ, Υ_2^i		j	SHØW: Γ, Π_2^i	
$j+1$		Υ_1	$j+1$		Π_1
		$\vdots$			$\vdots$
k		$\nu \in \Upsilon_2$	k		$\pi \in \Pi_2$

Both subproofs in boxes are strict and no reiteration into them is allowed. The only U-clauses taken from the outer derivation are modal assumptions in line $j+1$ being subclauses of U-clauses from line i with respective modal functors deleted.

For **R** and **K** we must admit also modal reiteration of other Υ-clauses. It is possible to define suitable rules in full generality, where Υ-parts of any clauses are separated and reiterated into strict subderivation; exactly as it was stated in the preceding rules. But such rules are very complex to state, so we rather propose simpler variants, where only Υ-clauses with no other formulae are considered. One should note that this solution is sufficient since every clause of the form Γ, Υ^i may be separated into two clauses Γ and Υ^i with the help of (suitably many) applications of $[SUB]$ or

[6]Strictly speaking we should say RND-KM, but we usually omit this parameter since KM is practically the only variant of ND we are using in this book for the presentation of modifications of standard approach.

one application of admissible rule $[SEP]$ (cf. Chapter 4).[7] So the rules may be stated as follows:

$[CNEC]$ if $\Upsilon_1;...;\Upsilon_n \vdash \nu$, then $X \vdash \Upsilon^i_{n+1}$, provided for each $k \leq n$, $\Upsilon^i_k \in X$ and $\nu \in \Upsilon_{n+1}$

$[CPOS]$ if $\Pi_1;\ \Upsilon_1;...;\Upsilon_n \vdash \Pi_2$, then $X \vdash \Pi^i_3$, provided for each $k \leq n$, $\Upsilon^i_k \in X$, $\Pi^i_1 \in X$ and $\Pi_2 \subseteq \Pi_3$

The completion of these rules in RND on diagrams look as follows:

$$\begin{array}{lll} & \mathcal{D} & \\ i & \text{SHØW: } \Upsilon^i_{n+1} & \\ & & \boxed{\begin{array}{l}\mathcal{D}' \\ \nu \in \Upsilon_{n+1}\end{array}} \\ k & & \end{array} \qquad \begin{array}{lll} & \mathcal{D} & \\ i & \text{SHØW: } \Pi^i_3 & \\ & & \boxed{\begin{array}{l}\mathcal{D}' \\ \Pi_2 \subseteq \Pi_3\end{array}} \\ k & & \end{array}$$

where: $X \subseteq$U$(\mathcal{D})$ and the only U-clauses in U$(\mathcal{D}')$ are either Υ_k obtained by reiteration from $\Upsilon^i_k \in X$ or clauses inferred from them. Additionally in $[CPOS]$ we have $\Pi_1 \in$U$(\mathcal{D}')$ in line $i+1$ as a modal assumption.

Thus in both cases subderivations in boxes are strict since only direct subformulae of ν-formulae (and the chosen π-formulae in $[CPOS]$) from the outer derivation are transferred by reiteration to them, all other are forbidden.

We omit a detailed formulation of reiteration rule and a definition of derivation for RND-**L**; it may be stated less formally as in Chapter 6 for standard modal ND, or more formally as respective definition for classical RND from Chapter 4, but with the addition that premises of inference rules must be present in the same subproof and with an explicit clause for reiteration. We leave it to the reader and instead we provide an example of a proof in RND+[CPOS] for **R**. If we want to show that $\Box p \vee \Box q, \Box(r \rightarrow p) \rightarrow \Diamond s, \Diamond q \rightarrow \Box t \vdash_R \Box r \rightarrow \Diamond s \vee \Diamond t$, we build a derivation for a clause $\Gamma = \neg(\Box p \vee \Box q), \neg(\Box(r \rightarrow p) \rightarrow \Diamond s), \neg(\Diamond q \rightarrow \Box t), \Box r \rightarrow \Diamond s \vee \Diamond t$. It is displayed below:

[7] It is in contrast to Fitting's [94] system of destructive resolution, where a special rule enabling such separation must be introduced.

1	SHØW: Γ	$[16, SUB]$
2	$\Box p \vee \Box q$	$ass.$
3	$\Box(r \rightarrow p) \rightarrow \Diamond s$	$ass.$
4	$\Diamond q \rightarrow \Box t$	$ass.$
5	$\Box p, \Box q$	$(2, C\beta)$
6	$\neg\Box(r \rightarrow p), \Diamond s$	$(3, C\beta)$
7	$\neg\Diamond q, \Box t$	$(4, C\beta)$
8	SHØW: $\neg\Box r, \Diamond s, \Diamond t$	$[14, CPOS]$
9	$\neg(r \rightarrow p), s$	$m.ass(6)$
10	p, q	$(5, Reit)$
11	$\neg q, t$	$(7, Reit)$
12	$r \rightarrow p, q$	$(10, C\beta)$
13	q, s	$(9, 12, Res)$
14	t, s	$(11, 13, Res)$
15	$\neg\Box r, \Diamond s \vee \Diamond t$	$(8, C\beta)$
16	$\Box r \rightarrow \Diamond s \vee \Diamond t$	$(15, C\beta)$

Note also that these rules are in fact clausal counterparts of $[NEC\text{-}K]$ and $[POS\text{-}K]$, not their general versions, since we admitted only minimal transfer of formulae characteristic for **K**. But both rules are adequate also for **R** if we insist for $[CNEC]$ that $n \geq 1$ (i.e. at least one Υ-clause is reiterated), and for $[CPOS]$ that there is $\Pi_1^i \in X$ yielding modal assumption; in **K** these conditions are not required. If it is important to distinguish both versions we will write $[CNEC_R], [CNEC_K]$ respectively.

These rules give us suitable elimination and introduction rules for π- and ν-clauses in RND but, similarly as in ordinary ND, they are interderivable, which is stated in the following:

Lemma 7.4 (Equivalence of proof construction rules)

1. *$[CPOS]$ is admissible in RND+$[CNEC]$ in* **M, R, K**
2. *$[CNEC]$ is admissible in RND+$[CPOS]$ in* **M, R, K**

Proof It is worth giving at least a proof of one part of the lemma, since suitable schemata of elimination show that, despite theoretical dispensability, in practice we may significantly shorten proofs by using them both. We demonstrate the first case in version for **R** and **K**, because it is slightly more complicated than for **M** and than elimination of $[CNEC]$. Let us consider a schema of a part of a proof containing an application of $[CPOS]$:

1	$\Pi_1^i = \{\pi_1^i, \ldots, \pi_k^i\}$	
2	Υ_1^i	
	$\vdots$	
$n+1$	Υ_n^i	
$n+2$	SHØW: $\Pi_3^i = \{\pi_{k+1}^i, \ldots, \pi_m^i\}$	$[o, CPOS]$
$n+3$	Π_1	*ass.*
$n+4$	Υ_1	$(2, Reit)$
	$\vdots$	
$n+3+n$	Υ_n	$(n+1, Reit)$
	$\mathcal{D}$	
o	$\Pi_2 \subseteq \Pi_3$	(a result of $\mathcal{D}$)

A subproof starting with line $n+3$ and justified by $[CPOS]$ may be replaced by the following derivation displayed on the next page. Since it is rather complicated we provide some comments:

When analysing this elimination schema, one should note that only $[SUB]$ and $[CNEC]$ are applied in it. In subderivation starting with line $n+3$ this time we must write down complements of all elements of Π_3^i as assumptions and additionally by ordinary reiteration we put inside clauses $\Upsilon_1^i - \Upsilon_n^i$ from outer subderivation (lines $2 - n+1$ from simulated proof). Note that lines $n+3 - o+n$ contain clauses built from ν-formulae only. Lines $o+n+2 - s+1$ contain subderivation $\mathcal{D}_1$ which provides justification of $-\pi_1^i$ (the complement of the first formula from clause Π_1^i from line 1). We apply here modal reiteration to all modal clauses from lines $n+3 - o+n$, i.e. we rewrite them all, deleting every $\Box$ (or $\Diamond$ preceded by negation). Inside $\mathcal{D}_1$ we open one more subderivation (lines $p+n+2 - s+1$) which justifies $-\pi_1$. In this subproof we infer a clause Π_1 from the assumption π_1 by k applications of (W), and next by reiteration we add clauses $\Upsilon_1 - \Upsilon_n$. It allows repetition of $\mathcal{D}$ which, in simulated proof schema, yields Π_2. By reiteration we rewrite also all clauses from lines $o+n+2 - p$, so we have a complement of every element of Π_2. It yields $\bot$ by suitable number of application of resolution, and closure of subderivation from lines $p+n+2 - s+1$ by $[SUB]$. Derived $-\pi_1$ leads to closure by $[CNEC]$ of all subderivation $\mathcal{D}_1$ and justifies $-\pi_1^i$. We repeat these steps deriving one by one all unit clauses $-\pi_2^i - -\pi_k^i$ with the help of subderivations $\mathcal{D}_2 - \mathcal{D}_k$, which are exact copies of $\mathcal{D}_1$. In this way we obtain complements of all elements of Π_1^i which, by k applications of resolution leads to $\bot$ and closes a proof of Π_3^i by $[SUB]$. ■

$n+2$	SHØW: $\Pi_3^i = \{\pi_{k+1}^i, \ldots, \pi_m^i\}$	$[u, SUB]$
$n+3$	$-\pi_{k+1}^i$	*ass.*
	$\vdots$	
o	$-\pi_m^i$	z
$o+1$	Υ_1^i	$(2, Reit)$
	$\vdots$	
$o+n$	Υ_n^i	$(n+1, Reit)$
$o+n+1$	SHØW: $-\pi_1^i$	$[p+n+1, CNEC]$
$o+n+2$	$-\pi_{k+1}$	$(n+3, Reit)$
	$\vdots$	
p	$-\pi_m$	$(o, Reit)$
$p+1$	Υ_1	$(o+1, Reit)$
	$\vdots$	
$p+n$	Υ_n	$(o+n, Reit)$
$p+n+1$	SHØW: $-\pi_1$	$[s+1, SUB]$
$p+n+2$	π_1	*ass.*
	$\vdots$	
q	Π_1	$(p+n+2, W)$
$q+1$	Υ_1	$(p+1, Reit)$
	$\vdots$	
$q+n$	Υ_n	$(p+n, Reit)$
	$\mathcal{D}$	
r	$\Pi_2 \subseteq \Pi_3$	(a result of $\mathcal{D}$)
$r+1$	$-\pi_{k+1}$	$(o+n+2, Reit)$
	$\vdots$	
s	$-\pi_m$	$(p, Reit)$
	$\vdots$	
$s+1$	$\perp$	$(r, r+1 - s, Res)$
$s+2$	SHØW: $-\pi_2^i$	$[CNEC]$
	$\mathcal{D}_2$	
	$\vdots$	
$s+k$	SHØW: $-\pi_k^i$	$[CNEC]$
	$\mathcal{D}_k$	
t	Π_1^i	$(1, Reit)$
	$\vdots$	
u	$\perp$	$(t, o+n+1, s+2, \ldots, s+k, Res)$

Note also that in RND we avoid problems of full equivalence of such rules characteristic for standard ND, where $[POS]$ is equivalent to $[NEC]$ in monotonic and regular logics; for normal logics, stronger rule $[\bot]$ was necessary. But $[CPOS]$ covers both variants: with Π_2 nonempty it corresponds to $[POS]$ and with Π_2 empty, to $[\bot]$. In fact, we may obtain a fully uniform formalization of **M, R** and **K** if we change a bit a formulation of suitable rules for **M** by deleting nonmodal Γ, and $[POS]$ for **R** and **K** by letting a clause $\Pi_2 \subseteq \Pi_3$ which closes a subproof to be only unit clause (as in $[POS_M]$). The latter change is necessary since although in regular and normal logics any Π_2 is admissible, in monotonic logics only a version with unit clause preserves normality. Such rules are sufficient since, as we remarked above, a separation of modal part of any clause is always possible. Thus we have only one pair of constant rules for all these logics – the version with modal assumption; for **R** additionally reiteration is admitted, and for **K** modal assumption for $[CNEC]$ is stated as optional.

Also, due to Lemma 7.4., we can use either of these rules to obtain A- or I- RND-formalization of **M, R, K**, similarly as in standard ND. Indeed both rules are of interest; $[CPOS]$ makes RND-system closer to tableaux, so it may be better for describing proof-search, $[CNEC]$ is more useful for a comparison with ordinary axiom systems and makes the completeness proof simpler what we use in the proof of:

Theorem 7.2 (Adequacy) *RND+$[CNEC_L]$ is adequate for* **L** *being* **M, R, K**

Proof For soundness it is enough to show that $[CNEC]$ is normality preserving. We demonstrate both versions. For $[CNEC_M]$ assume that ν follows, in the sense of local consequence, from $\vee\Upsilon_1$ and that $\vee\{\Gamma, \Upsilon_2^i\}$ does not follow from $\vee\{\Gamma, \Upsilon_1^i\}$. Hence $\| \vee \Upsilon_1\| \subseteq \|\nu\|$, and at some point say w_1, in some neighbourhood model $\vee\{\Gamma, \Upsilon_1^i\}$ is true but $\vee\{\Gamma, \Upsilon_2^i\}$ is false. Since all disjuncts of $\vee\{\Gamma, \Upsilon_2^i\}$ (including ν^i) are false in w_1, then $\|\nu\| \notin \mathcal{N}(w_1)$ and $w_1 \vDash \vee\Upsilon_1^i$. But in **M**, $\vee\Upsilon_1^i$ implies $\Box(\vee\Upsilon_1)$, hence $w_1 \vDash \Box(\vee\Upsilon_1)$ and $\| \vee \Upsilon_1\| \in \mathcal{N}(w_1)$. From this by $\| \vee \Upsilon_1\| \subseteq \|\nu\|$ and condition (m) we get $\|\nu\| \in \mathcal{N}(w_1)$ which yields a contradiction.

For $[CNEC]$ assume that ν follows, in the sense of local consequence, from $\vee\Upsilon_1, ..., \vee\Upsilon_n$. If $\vee\Upsilon_{n+1}^i$ does not follow from X, then at the same point, say w_1, in some Kripke model, all disjuncts in $\vee\Upsilon_{n+1}^i$ (including ν^i) are false, whereas all clauses from X are true. So for each disjunct in $\vee\Upsilon_{n+1}^i$, there is a point $\mathcal{R}_i$-accessible from w_1 where the corresponding formula with canceled modal operator is false; in particular, let ν be false in w_2. Since

w_2 is $\mathcal{R}_i$-accessible from w_1, then for each $\vee\Upsilon_k^i \in X$ we have $\vee\Upsilon_k$ true in w_2 which yields a contradiction.

For completeness: interderivability of boxes and diamonds is captured by our definition of ν- and π-formulae, we omit an easy proof of K in RND-**R**, finally, any application of $(RM), (RR), (RG)$ is easily simulated by $[CNEC]$ in respective logics. ■

7.4.2 RND for Other Modal Logics

The above system may be generalized to several normal logics by suitable modification of reiteration rule in the same way as ordinary ND. We have already discussed the advantages and disadvantages of this approach. Therefore, although suitable reiteration rules may be refined to comply with clausal form, we follow here a different line of development. Partly it is stated here for the sake of variety and partly because this approach may be directly transferred to labelled systems that will be introduced later.

We can extend RND-**K** in two ways, either by adding a kind of expansion rule:

$(Exp\text{-}A)$ $\Gamma, \varphi \ / \ \Gamma, \psi$

or a kind of generalized resolution rule:

$(Res\text{-}A)$ $\Gamma, \varphi \ ; \ \Delta, -\psi \ / \ \Gamma, \Delta$

In both cases we will call such a system MRND (modal RND). Note that $(Exp\text{-}A)$ and $(Res\text{-}A)$ are schemata of many rules, where a parameter A is instantiated by the name of some axiom, whereas φ and ψ are defined accordingly for each axiom A. Essentially $(Res\text{-}A)$ is a resolution rule modulo some unification of formulae specified in the table displayed below, whereas $(Exp\text{-}A)$ is rather tableau-like form of extending a system. Using either type of rule yields the same effect since they are interderivable.

Lemma 7.5 (Equivalence of $(Res\text{-}A)$ and $(Exp\text{-}A)$)

1. *$(Exp\text{-}A)$ is derivable in RND+$(Res\text{-}A)$*

2. *$(Rez\text{-}A)$ is derivable in RND+$(Exp\text{-}A)$*

Proof

1. Assume Γ, φ in RND+$(Res\text{-}A)$. We write down Γ, ψ as a Show-line and $-\psi$ as the only assumption of this subderivation. From Γ, φ and $-\psi$

we obtain, by (*Res*-*A*), Γ which is sufficient to close this subderivation by [*SUB*] and makes Γ, ψ U-clause inferred from our first assumption only.

2. If we assume both Γ, φ and $\Delta, -\psi$, then from the first one we deduce, by (*Exp*-*A*), Γ, ψ, and this clause together with the second assumption gives us, by (*Res*), a clause Γ, Δ. ■

Although interderivable, it seems that especially MRND constructed with the help of (*Res*-*A*) is a very natural way to make use of properties of RND, since we do not need to introduce other kind of rules. In case of **M, R, K** we have a simple instance of (*Res*) i.e. $\varphi = \psi$; for extensions the table specifies respective values of both formulae for some axioms.

Axiom	φ	ψ	side condition
D	ν^i	π^i	$\nu{=}\pi$
DC	π^i	ν^i	$\nu{=}\pi$
T	ν^i	ν	–
TC	π^i	π	–
4	π_1^i	π_2^i	$\pi_1{=}\pi_2^i$
$4C$	ν_1^i	ν_2^i	$\nu_1{=}\nu_2^i$
5	π^i	ν^i	$\nu^i{=}\pi$
B	π^i	ν	$\nu^i{=}\pi$
B-Te	π^i	ν	$\nu^j{=}\pi,\ i \neq j \in \{F, P\}$

For example, by (*Res*-*B*) we may deduce a clause Γ, Δ from two premises: $\Gamma, \Diamond\neg\Diamond p$ and Δ, p, because $\Diamond\neg\Diamond p$ is our π^i, its π is $\neg\Diamond p$ which is our ν^i (side condition), and its ν is $\neg p$, so p is $-\nu$ required in the schema of the rule.

With the help of suitable instances of (*Exp*-*A*) or (*Res*-*A*) one may obtain in a modular way a formalization of all basic (monotonic, regular, normal) logics, as well as some normal logics of functional and dense (axiom $4C$) accessibility relation. Moreover, with the help of (*Res*-*B*-*Te*) (or (*Exp*-*B*-*Te*) and possibly some other rules one may obtain a formalization of some temporal logics like **Kt, Kt4, Kt4D**.

To prove completeness of our systems of MRDN is nothing more than to compare values of φ and ψ in the table with suitable axioms collected together in Chapter 5 (cf. the table in Section 5.3) On the basis of the fact that RND+[*CNEC*] (or [*CPOS*]) suitably stated is equivalent to axiomatic **M, R, K** we can prove equivalence with any axiomatic formalization of **L** over one of these logics in a modular way. We use (*Exp*-*A*) for that. Assume we have added an axiom $A = \varphi \to \psi$ to RND+$[CNEC]$, we can simply prove derivability of the corresponding (Exp-A). From A by $(C\beta)$ we get $-\varphi, \psi$, if we assume a premise of our rule Γ, φ, then by (Res) we obtain Γ, ψ. On the contrary, having (Exp-A) we can easily prove A. Assume negation of A, then by $(C\alpha)$ we obtain φ and $-\psi$. But from φ we deduce, by (Exp-A), ψ, which by (Res) leads to $\perp$.

In RND one may simulate some resolution systems invented for modal logics; we briefly compare our solution with an approach of Fitting [94]. Fitting's system operates on so called blocks that are disjunctions of conjunctions of disjunctions, so his rules are a bit complicated. He needs some mechanism for splitting chosen disjunction, for example if we have $X;\ \Gamma, \Delta$ we must be able to obtain $X;\ \Gamma$ or $X;\ \Delta$. This is something like branching in tableau systems and the only reason for using two levels of disjunctions instead of simple normal forms. The suitable rule is called Special Case Rule.

In RND things are simpler: our clauses correspond to basic disjunctions (“,” stands for $\vee$) and sets of U-clauses present in the same subderivation correspond to conjunctions (“;” stands for $\wedge$). In order to simulate the next level of disjunction (blocks) and to simulate the branching effect of Special Case Rule we do not need any special rules. Every application of Special Case Rule is easily simulated by $[SUB]$ – any time we need to split some clause, say Γ, Δ, we simply start S-line for Γ and write down all assumptions. By successive application of (Res) we get Δ, hence this subderivation corresponds to $X;\ \Delta$ and if it closes we have $X;\ \Gamma$, since Γ becomes U-clause after closing a subproof.

The system of Fitting is called destructive, since rules for modals cause that some clauses are deleted in blocks (and some are modified). His rule for **K** allows every set of clauses of the form $\{\Delta_1, ..., \Delta_i, \Upsilon_1^i, ..., \Upsilon_k^i, \Pi^i\}$ to be replaced by $\{\Upsilon_1, ..., \Upsilon_k, \Pi\}$. Now it is evident why blocks are needed and splitting of some clauses; we must separate subsets containing only ν- or π-formulae from the rest, before we may apply modal rules. In order to simulate this effect in MRND we must simply apply $[POS]$ with reiteration.

[94] provides also rules for **T**, **K4** and **S4**, that are easy to simulate in MRND.[8] Although our approach is more extensive and easier to deal with than Fitting's system, it has also some disadvantages. Similarly as standard ND, MRND is not confluent, so even if in some cases a proof search may run faster, in general it is also not a system well suited for model extraction

[8]In fact, Fitting's rule for **T** is just our (Exp-T).

from open (and finished) derivations. In case of MRND we have lost also some nice feature of classical RND – the possibility of building "flat" proofs (of degree=1). It is mainly a consequence of introduction of additional proof construction rules which must be used in essentially modal proofs, but also a necessity of using $[SUB]$ to split mixed clauses into modal and non-modal parts.

We leave to the reader a details of extending MRND to cover congruent or first-order modal logics. In the latter case one may use RND rules for **CQL** and **FQL** from Chapter 4. In the former, the simplest way is just to enrich RND with proof construction rule $[NEC'_E]$ stated in Section 6.5, to formalize **E**, and use (*Res*-*A*) or (*Exp*-*A*) for extensions. Note however that some generalization of $[NEC'_E]$ to clausal form is possible.

7.5 Nonstandard Deductive Systems

Restricted application of standard proof methods to modal logics generated two strategies: either construct nonstandard proof system better suited to formalization of modal logics, or change the language into something more sensitive to the application of standard methods. The second choice has led to invention of hybrid logics[9] that – as we shall see – represent far better behavior when formalized with standard tools. We will present hybrid logics and some of their formalizations in Chapters 11 and 12.

For the time being we focus on the first strategy. Nonstandard approaches usually based on the use of richer metalogical apparatus, appeared very fruitful and, in particular on the ground on the methodology of sequent calculi, have led to the invention of many interesting general frameworks suitable not only for modal logics. We can mention here for example the method of *hypersequent calculi* due to Avron, or Belnap's general theory of *display calculi* (in particular for modal logics, the presentation of these and many other approaches may be found in Wansing [280, 281] or Poggiolesi [214]). We are not going to describe these systems because most of them cannot be used as a source of modification of ND system – which is our main goal in this book. There are two exceptions however. The first consists of labelled systems in the wide sense of the word; for presentation of this approach we reserve the rest of the book, since it may be combined in a variety of ways with ND systems. As for many other approaches we restrict our attention only to some group of nonstandard TS's which, despite many

[9]But not only; we can mention also description logics in this context despite its different origin.

differences, represent some structural similarity. All of them are hybrid systems in the sense of combining TS with some graphical interface representing semantical environment of modal logics. In fact, these solutions are rather not possible to combine with ND but offer an interesting way of formalizing temporal logics including linear ones, and this is the reason for inclusion of their brief presentation in this book. Moreover, one of these proposals, namely Kashima nonstandard TS, is a handy tool for representing several rules introduced for linear logics and will be applied throughout for this aim, especially in Chapter 9.

7.5.1 Semantic Tableaux of Kripke

It is worth noting that the system of Hintikka-style tableaux for modal logics described in Chapter 6 is not the only solution that may be called standard. One may remember that in Chapter 3 we have presented three popular variants of this method, due to Hintikka, Beth and Smullyan. In fact, classical works of Kripke [169, 170], which are treated as a milestone in the development of relational semantics, introduced also the extension of the method of Beth diagrams to many modal logics. Further extension of this type of tableaux may be found in Zeman [288]. The solution proposed by Kripke may seem rather obvious from the point of view of relational semantics; for every point in a model we build a separate diagram, whereas external (with respect to deductive system) information concerning properties of accessibility relation governs the process of rewriting some chosen formulae from one diagram to another.

In such TS semantical elements are even more important than syntactic base. So Kripke system belongs to the class of hybrid systems in the sense of combining a deductive system with explicitly represented elements of semantics. Such an approach has many advantages: a definition of rules is invariant, in case of logics with symmetric accessibility relation, there is no problem with moving back to earlier states (diagrams), e.t.c.. This approach is also format insensitive: one may combine with every state not only Beth diagrams, but Smullyan's trees or simply a form of truth-table test (like in [135]). In fact, there is no reasons why not to combine this approach with any deductive system, like KE or ND. Such an approach is also evident in the way of presentation of tableaux for description logics due to Horrocks ([133] or [134]), where an internal representation of a deduction in states is rather not essential; an external tree of states is the main concern of the authors. Similar solution is applied by Heuerding, Seyfried and Zimmermann [130], where trees of worlds are built with sets of formulae as nodes.

But in this approach there is no special deductive system but rather an embedding of such a system in some graphically represented semantic frame. In tableau systems of this kind a natural relationship to modal SC's is lost. Fitting [93] noticed also that systems of this sort may cause some practical problems, since instead of one proof-tree we may easily generate a whole forest. But it is rather a problem of pen and paper realization not that of automation. There are well known implementations of this technique with very good performance, e.g. due to Horrocks, Sattler and Tobies for description logic in [134]).

7.5.2 Tableaux with Boxes

It is quite natural that some approaches were proposed that represent the idea of Kripke solution in a more compact way. Slightly more subtle variant of Kripke's approach may be found in Rescher and Urquhart [231]. They present an extension of Smullyan's TS for temporal logics, where Jaśkowski's boxes are applied for separation of parts of proof-tree that hold in the same point. So we have only one tree but a node may be either a formula or a box containing a subtree, where further boxes may be embedded. Every box is labelled; if the box labelled with t' is inside the box with label t, it means that $\mathcal{R}tt'$. Essentially the same solution is applied also by Boolos in his TS for **G** [50]. Also Garson's system of diagrams in [105] belongs to this category. Systems of this sort are quite close to standard ND systems for modal logics due to Fitch, but there is one difference. The tree of boxes in such a version of TS corresponds strictly to the tree of worlds in attempted falsifying model, whereas in Fitch's style ND only some boxes (i.e. strict subproofs) realize this function.

From the standpoint of practical application Rescher/Urquhart system has one serious disadvantage. Every time we want to move some ν-formula (or its direct subformula) to other temporarily finished box, we must rewrite all the diagram.[10] Once again, in case of implementation, it may be no problem at all. We are not going to present this system in detail, but at least the solution for logics of linear time is worth describing. It is postponed however to Chapter 9 for easier comparison with other approaches.

[10]It is avoided in Garson's system but at the expense of dividing boxes corresponding to one world into several parts.

7.5.3 Systems of Higher Level

This approach in general is based on the idea of using nested structures as basic items. It has at least four independent and very different manifestations. Došen [80] has applied such a solution to **S4** and **S5** building SC, where sequents appear as elements of other sequents. Kashima [161] on the basis of some ideas of Sato [240] applied similar solution to obtain formalization of some temporal logics. Stouppa [264, 265] defined a system of deep inference for **S5**, extending the calculus of structures due to Guglielmi. Recently Poggiolesi [214] proposed tree-hypersequent calculi for basic normal modal logics and **G**, where ideas of Avron's hypersequent calculus are considerably generalized.

We restrict a presentation to the system of Kashima because it covers logics which are of interest for us. Kashima himself calls his system SC, but in accordance with the distinctions introduced in Chapter 3 we must rather say that it is a generalization of Schütte's format TS. For easier comparison with solutions introduced earlier we will present his system in dual form, i.e. as a generalization of TS in Hintikka format.

Building blocks of Kashima's TS are just families of sets of formulae, called K-sequents.[11] The notion of K-Sequent may be defined inductively:

Definition 7.1

- any finite set of formulae is a K-sequent;
- if X is K-sequent, then $\{^F X\}$ and $\{^P X\}$ are also K-sequents;
- if X and Y are K-sequents, then X,Y is a K-sequent.

The application of braces allows us to encode in a single K-sequent the whole attempted model, not just one point of its domain as in standard modal TS. For example, a K-sequent $\Gamma\{^F\Delta\{^P\Sigma\}\{^F\Pi\}\}$ denotes a model containing four points: t_1 satisfies Γ, t_2 satisfies Δ, t_3 satisfies Σ, and t_4 satisfies Π, moreover for **Kt**, $\mathcal{R}t_1t_2$, $\mathcal{R}t_3t_2$ and $\mathcal{R}t_2t_4$; for stronger logics there may be additional clauses, e.g. for **Kt4** it will be transitive closure of $\mathcal{R}$, i.e. $\mathcal{R}t_1t_4$ and $\mathcal{R}t_3t_4$. This is the basic idea; temporal rules are the following in **Kt**:

$(FE)\ \dfrac{X\ [F\varphi]}{X[\{^F\varphi\}]}$ $\qquad$ $(PE)\ \dfrac{X\ [P\varphi]}{X[\{^P\varphi\}]}$

[11] Kashima uses lists of formulae not sets which forces him to use also some structural rules. From the point of view of our needs it is unnecessary complication, but as a result also our version of Kashima's rules for temporal connectives differ somewhat from the original.

$$(G\text{-}Exit)\ \frac{X\ [G\varphi\ \{^F Y\}]}{X\ [G\varphi\ \{^F \varphi, Y\}]} \qquad (H\text{-}Exit)\ \frac{X\ [H\varphi\ \{^P Y\}]}{X\ [H\varphi\ \{^P \varphi, Y\}]}$$

$$(G\text{-}Enter)\ \frac{X[\{^P G\varphi, Y\}]}{X[\varphi\ \{^P G\varphi, Y\}]} \qquad (H\text{-}Enter)\ \frac{X[\{^F H\varphi, Y\}]}{X[\varphi\ \{^F H\varphi, Y\}]}$$

We have used [ ] for indication that elements inside are on the arbitrary level of embedding. It is not necessary since any operations may be done on the level 0 (i.e. no braces), but for that we need two additional rules:

$$(turn)\ \frac{\{^P X\}\ Y}{X\ \{^F Y\}} \qquad \frac{\{^F X\}\ Y}{X\ \{^P Y\}}$$

In the presence of (*turn*) both rules (*Enter*) are of course derivable. Kashima distinguishes both systems and shows their equivalence with regard to formalization of temporal logics; it should be noted however that the first approach is – from the standpoint of potential applications – more general solution. One may admit only one sort of braces and in this way obtain a formalization of monomodal logics, but then rules like (*turn*) are correct only for symmetric logics.

Kashima's system is an interesting extension of Sato's ideas and has potentially wide scope of application. It is worth noting that this TS is confluent, analytic and cut-free for many logics, where standard approach fails to have these features. Also in this kind of a system we may simulate easily many solutions introduced on the ground of other systems which makes it particularly useful for comparison of different techniques. This feature of Kashima's system will be particularly valuable for our interests. For example, every diagram of Rescher/Urquhart may be easily rewritten as a K-sequent, and the opposite also holds. For us the former is more important, since we may use Kashima's system for concise representation of Rescher/Urquhart's rules. On the other hand, if Kashima's system is considered as a tool for actual pen and paper proof search it seems not to be very handy. The necessity of rewriting all parametric formulae in every application of a rule is undoubtedly a disadvantage.

Chapter 8

Labelled Systems in Modal Logics

One of the most important nonstandard approaches to formalization of modal logics is based on the application of labels. This technique is connected not only with modal logics but has a really wide scope of application in several branches of logic. In modal logic labels extend a language with a representation of states in a model. Their addition considerably increase the flexibility of expression.

Section 8.1 introduces some preliminary taxonomy of solutions based on the application of labels. Roughly speaking, they fall into external and internal approaches. A presentation of the internal approach, represented in particular by the use of hybrid languages, will be postponed to Chapters 11 and 12. For the time being, we focus on the external approach, where labels are metalinguistic devices. Existing labelled deductive systems of this sort may be divided additionaly on weak, medium and strong, according to the strength of labeling commitment.

The application of labels in weakly labelled systems is very restrictive. They usually work as an additional mechanism supporting the proof construction but not sufficient for extraction of falsifying models for invalid formulae. We will discuss some of such solutions briefly in Section 8.2.

The situation with strong labelling is similar to that of hybrid languages. The latter are in itself strong enough to express everything usually represented by algebra of labels in strongly labelled systems. In fact, hybrid languages are even stronger. Because of that, our discussion of strong labelling in Section 8.2 is very short and the potential capacities of such an approach will be presented in Chapters 11 and 12.

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*, Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_8,

In this Chapter, and in the next two, we focus on the approach of Fitting, called here medium labelling. It is one of the most popular solution for modal deduction – simple and natural. In this approach labels linked to formulae are finite sequences of natural numbers encoding, at the same time, the name of a state where this formula is evaluated and the place of this state in a (falsifying) model we are searching for. The technique and its generalization to multimodal logics is introduced in Section 8.3.

Although, historically (and also statistically), labelling is associated with tableau methods, it must be stressed that this technique is independent of the kind of a proof system we use. Clearly, our attention will be paid on ND systems. Sections 8.4 and 8.5 contain labelled ND system (shortly LND) for **K** and its extensions to other normal and regular logics. LND for first-order modal logic and for some temporal logics are also presented. In Section 8.6 we show how this technique may be extended to weak modal logics characterized by neighbourhood semantics. In the last section we present two kinds of labelled RND (LRND) systems for modal logics.

8.1 Kinds of Labelling

Generally labels are very handy if we deal with information having complex structure, especially when different sorts of data need different forms of processing. Dov Gabbay in his general theory of LDS's (labelled deductive systems) considered several applications from different fields. For example, labels may be used to represent:

1. fuzzy reliability value n ($0 \leq n \leq 1$) used mainly in expert systems,
2. the situation where the infon holds in situation semantics,
3. the set of assumptions for a formula (e.g. Anderson/Belnap [5] ND-systems for relevant logics),
4. truth values or the sets of truth values for a formula (e.g. Carnielli [61] or Hähnle [119] tableau systems for many-valued logics),
5. possible world (point of time) satisfying a formula in modal (temporal) logics.

Of course, for our aims, the last item is the most important. No doubts, the main breakthrough in the development of modal logic was theinvention of relational semantics. Simple, natural, philosophically motivated semantics is still considered as the basic tool for model theoretic investigations on modal logic, but five decades of research has shown that the correspondence with old syntactical tradition is far from being perfect. Carlos Areces has pointed out (as mentioned in [33]) that the very source of the problem is an asymmetry between local perspective of relational semantics and global perspective of standard modal language. Namely, states in a model which are essential in relational semantics, are not represented in modal syntax. But what's wrong with this? We can mention at least two undesirable results of this situation:

1. the lack of adequate representation for many semantic features
2. problems with suitable modal proof theory

The second item was already discussed in the preceding Chapter, whereas the first one was touched upon in Chapter 5. We have pointed out that standard modal languages have no mechanisms for naming particular *states* (*worlds*) in a model, asserting or denying equality of states, talking about accessibility of one state from another. All these things lie at the heart of modal model theory but there is no way of representing them in standard modal syntax. The situation is striking; especially if we compare it with the situation in classical first order logic, where elements of a model have direct representation in a language. In effect, many important properties of relational frames are expressible in a very roundabout way, while many others are not expressible at all in the standard modal language.

Hence the natural question arises how to find a remedy for the problem of discrepancy between a syntax and a semantics. One possibility is just to introduce an explicit syntactic representation of states in a model. Such an extension is needed to increase a flexibility of expression but it leads to the next question. In what way we can realize this task? It must be said that even in case of the application of labels in modal logics there is a lot of possible solutions.

Blackburn [32] distinguishes three kinds of labelled deduction systems:

1. external – labels as an additional technical apparatus,
2. internalized – labels as a part of a language (in particular, nominals in hybrid languages),
3. mixed – both nominals (in a language) and labels (metalinguistic devices) are present.

In the external approach we use additional metatheoretic apparatus connected to the language in question. In case of modal logics the most popular solution was the addition of the machinery of prefixes to formulae, due to Fitting ([92] in fact refers to earlier note of Fitch [90] as a source of inspiration). The best advocate of this approach in its generalized form is Dov Gabbay with his general theory of formalization of logics as labelled deductive systems [99].

The internalized approach consists of the enrichment of the object language obtained via sorting (of the atoms) and addition of the new operators and/or modalities. It is the way of doing hybrid logic. Logics of this sort and some of their formalizations will be presented in Chapters 11 and 12, where we also briefly describe mixed approach.

For the rest of the Chapter we focus on the external approach. Even in this group we can distinguish a variety of different solutions, according to the strength of semantical commitment expressed by labels. We divide them on three groups:

1. Weak labelling – labels as a very limited technical device supporting proof construction, e.g. tableau systems of Marx, Mikulas, Reynolds [183] for linear tense logics based on the use of three labels, multisequent calculi of Indrzejczak [146] for tense logics.

2. Strong labelling – a system of labels as an exact representation of an attempted falsifying model. Strongly labelled deductive system is a fusion of 2 systems: object language calculus + calculus for the algebra of labels, e.g. Gabbay's theory of labelled systems, Russo [237] ND-systems for modal logics, Basin, Matthews, Vigano [21, 22] ND-systems for nonclassical logics.

3. Medium labelling – with no special calculus for labels but still sufficient for construction of a falsifying model e.g. Fitting's [93] prefixed tableau calculi for modal logics or single-step tableaux of Massacci [185, 186], explored by Goré [117] under the name explicit systems.

In what follows we will concentrate mainly on medium approach because, in our opinion, it has a lot od advantages. It is quite simple and natural since the technical apparatus is kept in reasonable bounds. Fitting's labels (prefixes) are not as direct way of encoding semantics as strong labels or internalized approach, so one of Avron's condition of good deductive framework from [16] seems to be satisfied. Still they may be easily used for construction of falsifying models. ND systems with this kind of labels are free of many drawbacks of standard ND discussed previously, so they may be used as practical decision procedures and even applied in automated deduction. Moreover, this approach is quite extensive – Fitting's original systems cover a lot of normal and regular logics, Massacci's version formalizes even more; we extend it to basic weak modal logics, and to temporal logics (and multimodal logics in general). Some logics, like e.g. linear modal and temporal logics, are characterized by frame-conditions which seem to be hardly expressible in terms of Fitting's prefixes. One of the contribution of this book is to show that this technique may be used even to obtain a satisfactory formalization of this class of logics. So, in contrast to strongly labelled systems, we obtain quite satisfactory result with the help of relatively modest apparatus.

But first, we briefly discuss some systems representing weak and strong labelling. As usual we have chosen only a few systems that satisfy some of the criteria laid down in the Introduction and in the preceding Chapter. In particular, strongly labelled systems are presented in a sketchy way, because all the solutions from this area may be easily simulated by hybrid logics, and these will be discussed in the last two chapters.

8.2 Weak and Strong Labelling

8.2.1 Some Weakly Labelled Systems

There is a lot of deductive systems, where labels of several sort are applied in a very limited way. Probably the earliest one is SC of Kanger [160] for **S5**, where labels are linked only to propositional variables. Also TS of Rescher/Urquhart [231], described in the preceding Chapter, may be seen as a weakly labelled system because every box on the tree has a label denoting time point. Here we introduce two weakly labelled systems formalizing some bimodal temporal logics: TS of Marx, Mikulas, Reynolds [183] and SC of Indrzejczak [146]. What is of particular importance for us is the fact that both systems contain different rules for logics of linear time. We treat them as weakly labelled systems because in both cases labels play only a supporting role in a deduction, separating some parts of a derivation. Their motivation is in fact semantical but only a very small part of interpretation is involved. The apparatus of labels in itself is in both approaches too weak to help building a falsifying model, in contrast to medium and strong labelling, where we can directly extract a model from labels.

Nonstandard TS

TS of Marx, Mikulas and Reynolds [183], is in Hintikka format, but instead of sets of formulae, every node of a proof-tree is an ordered pair or a triple of sets of formulae. Parts of such triples are indexed with three labels: l(eft), m(iddle) and r(ight). They serve to indicate the relative position of three time points in a linear model, i.e. $\mathcal{R}I(l)I(m)$ and $\mathcal{R}I(m)I(r)$, where $I(k), k \in \{l, m, r\}$ denotes a state being a value of respective label. It is important that suitable label is added to some set only temporary and may change its position during proof construction. So labels "name" time points but not in an absolute sense like in labelled systems of strong and medium character. It is "relative" naming, showing at some stage of search of falsifying model, where this point is located with respect to some other points (immediate neighbours). We list below for illustration only rules for temporal constants, since extensional constants are treated by standard α-, β-rules realized in a set labelled by one of l, m or r.

$$(FE)\quad \frac{m:\ \Gamma, G\Delta, F\varphi}{l:\ \Gamma, G\Delta, F\varphi\ ;\ r:\ G\Delta, \Delta, \varphi}$$

$$(PE)\quad \frac{m:\ \Gamma, H\Delta, P\varphi}{l:\ H\Delta, \Delta, \varphi\ ;\ r:\ \Gamma, H\Delta, P\varphi}$$

An informal interpretation of (FE) is such that if a conjunction of elements of $\Gamma, G\Delta$ and $F\varphi$ is satisfied in a point labelled temporary by m, then we introduce a point labelled by r, where a conjunction of elements of $G\Delta, \Delta$ and φ is satisfied. Renaming of m for l is an evidence of establishing an accessibility relation between them. (PE) is dual rule.

Interesting rules are provided for weak connectedness:

$$(3FE)\quad \frac{l:\Gamma, G\Delta, F\varphi;\quad r:\Lambda, H\Sigma, \neg F\varphi, \neg\varphi}{l:\Gamma, G\Delta, F\varphi;\quad m:G\Delta, \Delta, \varphi, H\Sigma, \Sigma;\quad r:\Lambda, H\Sigma, \neg F\varphi, \neg\varphi}$$

$$(3PE)\quad \frac{l:\Gamma, G\Delta, \neg P\varphi, \neg\varphi;\quad r:\Lambda, H\Sigma, P\varphi}{l:\Gamma, G\Delta, \neg P\varphi, \neg\varphi;\quad m:G\Delta, \Delta, \varphi, H\Sigma, \Sigma;\quad r:\Lambda, H\Sigma, P\varphi}$$

One may note that in both rules the new point with label m is inserted between sets labelled with l and r. In $(3FE)$ it is created by $F\varphi$ from l-set, whereas in $(3PE)$ by $P\varphi$ from r-set. Inclusion of $G\Delta, \Delta, H\Sigma, \Sigma$ in m-set is necessary for providing assumed interpretation of a set m, i.e. that $\mathcal{R}I(l)I(m)$ and $\mathcal{R}I(m)I(r)$. It should be stressed that it is one of the few formalizations of linear logics which does not use branching rules to express connectedness. Marx, Mikulas and Reynolds provide also rules for other linear temporal logics but to obtain complete formalization they must use some form of analytic cut.

Remark 8.1 In fact, the system of Marx, Mikulas and Reynolds may be classified as a member of higher-arity proof systems. It is a family of non-standard systems that multiply the number of parts of a sequent. The natural place for such a solution was of course in many-valued logics (cf. [236, 61]), where the number of arguments corresponds to the number of truth values. But this approach has also some representation in modal logics, where application of more arguments is not always based on so direct semantical motivation. Except aforementioned ternary ST, we may distinguish at least two groups of solutions.

The first group contains SC's of Sato [240], and of Humberstone and Blamey [42] for some modal logics where sequents have 4 parts, and SC of Nishimura [195] for temporal logics with 6-ary sequents. The motivation is to distinguish formulae simply true from necessary true (necessary true in the past or future in temporal case).

The second group is a collection of several SC's, where in a standard sequent we have a separated part (set, multiset or list of formulae) encoding a "history" of attempted falsifying model. We can mention here SC of Heuerding, Seyfried and Zimmermann [130] for **S4** and **Kt4**, which are implemented in WorkBench program; a system of Mouri [192] for **K4** and **S4** (xpe program) and a system of Goré and Bonette for **Kt4** [49]. In all of them, despite the differences, a partition of a sequent facilitates control over loops and better managing of ν-formulae. In WorkBench and xpe, except traditional branching connected with β-formulae, an additional conjunctive branching is introduced to make a system confluent.[1] ♣

Multiple Sequent System

This system, presented in [146], was provided for temporal logics, similarly as the TS of [183]. The main feature of MSC is that each sequent has a label attached, being an ordered pair of natural numbers. In general, sequents are of the form $\Gamma < i, j > \Rightarrow \Delta$; if $i = j = 0$ it is an ordinary Gentzen's sequent, otherwise it is an intensional sequent of some sort. The basic motivation is semantic in nature; in intensional sequent an antecedent and a succedent refer to two different time points in attempted model. The label of a sequent shows how far apart these points are, via the future-oriented accessibility

[1] Cf. remarks in Section 7.2 on the lack of confluency in most of standard modal SC's.

relation. For example, a sequent $\Gamma \ <3,2>\Rightarrow \Delta$ shows that we have some points, say t_0, t_3 and t_2, such that t_3 is accessible from root-point t_0 in three steps and t_2 is accessible from t_0 in two steps, Γ is satisfied in t_3 and Δ is satisfied t_2. Hence, a semantic encoding of labels in MSC is not connected with each formula in a sequent, but with the whole sequent that corresponds to a partial description of the attempted model. It makes MSC labels only proof-supporting device not sufficient to obtain a description of a falsifying model.

In some respects MSC is more similar to such sequent systems like display calculus, where a number of structural rules is needed to characterize modal constants (cf. Wansing's exposition in [280]) MSC also contains plenty of rules with some of them, in a sense, structural. We display below as an illustration only the basic rules for temporal connectives:

$$(G\Rightarrow)\ \frac{\varphi\ <0,j>\Rightarrow \Delta}{\mathrm{G}\varphi\ <0,j+1>\Rightarrow \Delta} \qquad (\Rightarrow G)\ \frac{\Gamma\ <i,j+1>\Rightarrow \varphi}{\Gamma\ <i,j>\Rightarrow \mathrm{G}\varphi}$$

$$(F\Rightarrow)\ \frac{\varphi\ <i+1,j>\Rightarrow \Delta}{\mathrm{F}\varphi\ <i,j>\Rightarrow \Delta} \qquad (\Rightarrow F)\ \frac{\Gamma\ <i,0>\Rightarrow \varphi}{\Gamma\ <i+1,0>\Rightarrow \mathrm{F}\varphi}$$

$$(H\Rightarrow)\ \frac{\varphi\ <i,j>\Rightarrow \Delta}{\mathrm{H}\varphi\ <i+1,j>\Rightarrow \Delta} \qquad (\Rightarrow H)\ \frac{\Gamma\ <i+1,0>\Rightarrow \varphi}{\Gamma\ <i,0>\Rightarrow \mathrm{H}\varphi}$$

$$(P\Rightarrow)\ \frac{\varphi\ <0,j+1>\Rightarrow \Delta}{\mathrm{P}\varphi\ <0,j>\Rightarrow \Delta} \qquad (\Rightarrow P)\ \frac{\Gamma\ <i,j>\Rightarrow \varphi}{\Gamma\ <i,j+1>\Rightarrow \mathrm{P}\varphi}$$

$$(G^j\Rightarrow)\ \frac{\Gamma\ <0,j>\Rightarrow \Delta-\varphi}{\mathrm{G}^j\varphi\Gamma\ <0,j>\Rightarrow \Delta} \qquad (\Rightarrow F^i)\ \frac{-\varphi\Gamma\ <i,0>\Rightarrow \Delta}{\Gamma\ <i,0>\Rightarrow \Delta\ \mathrm{F}^i\varphi}$$

$$(H^i\Rightarrow)\ \frac{\Gamma\ <i,0>\Rightarrow \Delta-\varphi}{\mathrm{H}^i\varphi\Gamma\ <i,0>\Rightarrow \Delta} \qquad (\Rightarrow P^j)\ \frac{-\varphi\Gamma\ <0,j>\Rightarrow \Delta}{\Gamma\ <0,j>\Rightarrow \Delta\ \mathrm{P}^j\varphi}$$

Even this set of logical rules is not enough to obtain a formalization of **Kt**; one needs four structural rules:

$$(KF)\ \frac{\Gamma\ <i,0>\Rightarrow}{\Gamma\ <i+1,0>\Rightarrow} \qquad (KG)\ \frac{<0,j>\Rightarrow \Delta}{<0,j+1>\Rightarrow \Delta}$$

$$(KP)\ \frac{\Gamma\ <0,j>\Rightarrow}{\Gamma <0,j+1>\Rightarrow} \qquad (KH)\ \frac{<i,0>\Rightarrow \Delta}{<i+1,0>\Rightarrow \Delta}$$

MSC is confluent but lacks a general proof of cut-elimination although some simpler versions devised e.g. for **S5** (cf. [142]) were proved to be adequate in cut-free version.

Remark 8.2 In fact MSC may be seen as belonging to the wider category of SC's using more than one type of a sequent. Curry [75, 76] was the first to introduce the calculus with two types of sequents for **S4**, a formalization of this sort for **S4.4** is also due to Zeman [288]. Avron, Honsell, Miculan and Paravano in [17] considered a calculus where two types of sequents correspond to different deducibility relations. Indrzejczak [142] presented double sequent calculus for **S5** and more general constructions in [143], where two calculi were defined: GSC I has three types of sequents, whereas GSC II has a denumerable family of sequents of different *grades*, where a grade is simply a natural number. Although GSC I was sufficient to deal with regular and normal logics axiomatized by means of D, T, 4, its apparatus was rather poor to provide a cut-free formalization for symmetric logics, like **KTB**. It was the reason to introduce GSC II, where the machinery of grades enables syntactic representation of "moving back" to the preceding worlds in Kripke models – a characteristic feature of symmetric logics. MSC evolved as a system where advantages of both approaches may be combined. The first approach to combine GSC I and GSC II was rather trivial. There were three types of sequents: one classical, and two modal (like in GSC I) and, moreover, both modal types were ordered by grades (as in GSC II). However, this obvious combination, was not sufficient to capture such properties of Kripke models like Euclideaness or weak connectedness. In order to obtain a more expressive formalization it was necessary to mix both modal types in one sequent which was done in MSC by replacement of a grade by a label being a pair rather, than a single number. In consequence, MSC is technically more complicated, but has a significantly wider scope of application. ♣

8.2.2 Strong Labelling

The opposite solution to the question of building hybrid systems with the help of labels is represented by labelled systems in Gabbay's tradition [99]. In such an approach labels save as much as possible from suitable semantics, but in contrast to internalized approach, labels are not part of a language. Instead we have a composition of two languages: an object language of a logic and a language of an algebra of labels. In deductive system of this sort except rules for logical constants we have also rules governing the behaviour of labels, and usually some rules which correlate both levels. This form of labelled calculi is in fact very close to indirect (translational) approach to resolution for modal logics (cf. Section 3.3).

Such an approach is very popular. One may find several simplified variants of easy-in-use TS's of this kind for many nonclassical logics, including modal ones, in several textbooks (cf. e.g. books of Girle [111] and of Priest [222]). There are also more theoretically oriented works investigating labelled SC's for modal logics, e.g. Castellini and Smaill [63, 62], Negri [194]; TS for multimodal logics is provided by Baldoni [18, 19]. All these systems offer general results for wide classes of logics and will be discussed in more detail in the next Chapter and in Chapter 12; first, we discuss particular instances needed for linear logics, and then we introduce more general solutions in the context of deductive systems for hybrid logics.

The application of strong labelling was also used in the construction of ND systems for modal logics, independently by Russo [237] and Basin, Matthews and Vigano ([21] (also for other nonclassical logics in [22] and [287]).[2] In such a system, except labelled formulae (l-formulae for short), we have formulae of a relational language (r-formulae) which directly express properties of accessibility relation in suitable models. Let x, y, z denote labels, $x : \varphi$ an l-formula, xRy – an r-formula, and Γ – a set of formulae of any kind. For the sake of illustration and further comparison, we briefly describe ND system of Basin/Matthews/Vigano. They provide the following rules for **K**:

[$L\Box I$] If $\Gamma,\ xRy \vdash y : \varphi$, then $\Gamma \vdash x : \Box\varphi$, where $y \neq x$ is a label not occurring in any assumption in Γ

($L\Box E$) $x : \Box\varphi\ ,\ xRy\ /\ y : \varphi$

($L\Diamond I$) $y : \varphi\ ,\ xRy\ /\ x : \Diamond\varphi$

[$L\Diamond E$] If $\Gamma,\ y : \varphi\ ,\ xRy \vdash z : \psi$, then $\Gamma\ ,\ x : \Diamond\varphi \vdash z : \psi$, where y ($z \neq y \neq x$) is a label not occurring in any assumption in Γ

In this system the extensions of **K** are obtained in a modular way by adjoining rules defined on r-formulae only, e.g. to get a symmetry we need a rule: $xRy\ /\ yRx$. Although in this way one may define suitable rules for almost all normal modal logics introduced in Chapter 5, the interests of Basin, Matthews and Vigano are limited to these logics which may be formalized by Horn clauses in relational language, because their normalization theorem works for this class. We will take a closer look at the possibility of uniform formalization offered by so strong engagement of semantics in Chapters 11 and 12. In the meantime we will investigate the scope of application of systems which use labels in a more limited way.

[2] One may mention also ND systems with strong labelling for other kinds of temporal logics not discussed in this book, like **PLTL** or **CTL** provided by Renteria and Hausler [230] or by Bolotov, Basukoski, Grigoriev and Shangin in [47, 48].

8.3 Medium Labelling – Fitting's Approach

The popular approach of Fitting [93] is situated between the extrema of weak and strong labelling. There are no operations on labels performed in extra language like in Gabbay style systems – labels are always linked to formulae of an object language. On the other hand, labels have structure of their own which helps to build a model using them as building blocks. It is possible because each label is not only a name of a state in a model, but its structure encodes the place of this state in a falsifying model we are attempting to build.

From the technical point of view labels are nonempty finite sequences of natural numbers, separated with dots, with 1 as the first digit. We will use σ, τ, θ for labels or their parts (any strings of integers, not necessarily with 1 as the first digit and not necessarily nonempty), and i, j, k for natural numbers. Let LAB represent the set of all labels; formally it may be defined as follows:

Definition 8.1 (Labels)

1. $1 \in$LAB;
2. If $\sigma \in$LAB, then $\sigma.k \in$LAB.

$\sigma.k$ denotes the label with k as the last digit, $\sigma\tau$ represents the label being the concatenation of two strings, We will call σ a parent and $\sigma.i$ a child; except the root label 1 (which is not a child of any label), all other labels are children. We say that σ is an extension of τ, if $\sigma = \tau\theta$ for some nonempty string θ (recall that we do not assume that strings are always nonempty like labels). The length of a label σ (or its part) which is the number of digits in σ will be referred to as $\mid \sigma \mid$.

Hence 1, 1.1.3.2.1, 1.2.1.1.5 are all examples of labels, whereas 4.1.3.7 or 1.3.0.5 are not. Informally, if a label σ is a name of a state w in a model, then a structure of σ shows what points in this model are the $\mathcal{R}$-ancestors of w. For instance, the third example of a label (i.e. 1.2.1.1.5) may be read as a (partial) description of a model, where $1, 1.2, 1.2.1, 1.2.1.1$, and 1.2.1.1.5 belong to $\mathcal{W}$, and ordered pairs $\langle 1, 1.2\rangle, \langle 1.2, 1.2.1\rangle, ..., \langle 1.2.1.1, 1.2.1.1.5\rangle$ belong to $\mathcal{R}$. In general, for any label and its child it means that $\sigma.i$ is accessible from σ by $\mathcal{R}$.

The set of labelled formulae (or shortly l-formulae) – LFOR is defined as follows:

Definition 8.2 (Labelled Formulae – LFOR) If $\varphi \in$ FOR of $\mathbf{L_M}$, and $\sigma \in$ LAB, then $\sigma : \varphi$ is labelled formula ($\sigma : \varphi \in$ LFOR).

In what follows l-formulae will be represented by A, B, C, and their sets by X, Y, Z. LAB(X) stands for the set of all labels of l-formulae in X, and FOR(X) for the set of all formulae from X but with deleted labels.

Intuitively $\sigma : \varphi$ means that φ is satisfied at a point of a model denoted by σ.

Introducing 1 as the fixed first item of every label is strictly speaking not necessary. There are systems, e.g. SC-like system of Leszczyńska [175], where labels are essentially Fitting's style, but this requirement is not satisfied, at least locally. But insistence on having the root label simplifies adequacy proofs because the set of labels of any derivation has some important property:

Definition 8.3 (Strongly Generated Set) Let X be a set of l-formulae, then LAB(X) is *strongly generated*, iff:

- There is some root-label (namely 1).
- For any label σ, such that $\mid \sigma \mid \geq 2$, its parent also belongs to this set.

If LAB(X) is strongly generated, then X will be also called strongly generated.

Fitting with the help of such labels (called by him prefixes) obtained TS's in Smullyan's format for many monomodal normal and regular logics. These *prefixed systems* were then improved by Massacci [185] and extended to additional logics as *single step tableaux*. The difference between these two variants will be discussed in remarks at the end of Section 8.5; here we only point out that it is connected with the shape of rules and details of adequacy proof rather than with the notion of a label. So we will simply call such TS's as Fitting's format of labelled tableaux. A detailed presentation of this class of systems, under the name *explicit systems*, is given by Goré [117].

In many respects labelled systems are superior to other formalizations of modal logics discussed so far. Fitting's format tableaux are confluent[3] and, in a sense, analytic (cf. Section 10.1) in contrast to most of the standard systems. There is also no problems with the formalization of symmetric

[3]Massacci [186] provides a formal proof of this fact.

logics which is hard for standard systems. Branches of proof tree in Hintikka style TS are divided on separated levels corresponding to points of a model. If we jump to the next point (by application of (πE)) it is impossible to get back to the preceding one which is essential for symmetric logics. In Fitting's approach we may have formulae with different labels as immediate neighbours on the same branch, and we may walk freely from one point of a model to another and back again.[4] The last point is also realizable in Kripke format TS's but at the cost of introducing many trees instead of one. Labelled systems may be seen as an improvement of the last technique (e.g. [117] is putting things this way); we have only one tree (as in Hintikka style TS's), but labels give us more freedom in walking on the states of attempted models. This saves confluency because we do not loose any information (as we do in Hintikka TS). In consequence, we always need only one proof-tree, whereas in Hintikka style TS for establishing satisfiability (invalidity of a root-formula) we are usually forced to construct many open tableaux as building blocks of a model (cf. remarks on the lack of confluency in Section 7.2). Shortly, labelled tableaux always need only one tree, Hintikka TS for validity needs one tree but for satisfiability usually it is not enough, whereas Kripke's TS usually needs many trees in both cases. But one should add that the flexibility of Fitting's approach has also some costs in case of implementation for needs of automated deduction; we will explain this claim in Chapter 10. (in particular, Sections 10.1.2 and 10.5).

Although the original labelled systems mentioned above deal with monomodal logics, the technique may be extended to some multimodal cases. It is unproblematic in case of homogenous logics with no interaction, so we will provide generally systems for basic normal logics in multimodal homogenous version. A formalization of interactive logics with the help of medium labelling generally suffers from some limitations which may be overcome by strong labelling. We will discuss limitations at the end of the next Chapter but for the time being we provide systems for some bimodal temporal logics. By the way, it is rather surprising that labelled systems for temporal logics were not proposed earlier, taking into account their simplicity with handling symmetry in models.

Generally, in case of n-modalities we must divide the set of labels onto $n+1$ classes. There is a singleton $\{1\}$ because root-label is neutral and n sets of other labels corresponding to n accessibility relations. The membership of a label will be pointed out by inserting $[i]$ $(1 \leq i \leq n)$ between a label

---

[4]This feature makes Fitting's approach similar to Suppes' format ND as compared to Jaśkowski format – cf. the Remarks in Section 2.4.3.

and a formula. It means that this label (called i-label) is accessible from its parent via relation $\mathcal{R}_i$. For instance, if $\sigma.k$ is an i-label, (i.e. in a derivation we have for some φ, $\sigma.k\ [\ i\ ] : \varphi$, it means that $\langle \sigma, \sigma.k \rangle$ belongs to $\mathcal{R}_i$. Note that this solution does not require changes in the definition of a label, or labelled formula; only pointing out in the description of some rules to which class a label belongs. For example: let $\varphi := p \wedge q$, $i := 3$ and $\sigma := 1.2.1$, then a clause "$\sigma : \varphi$, where σ is i-label" is a metalinguistic description of a formula "1.2.1 [3] : $p \wedge q$". Clearly, in case of monomodal logics, there is no need to use these additional devices; we will make use of that in examples taken usually from monomodal logics. But we will make a substantial use of this device with respect to bimodal temporal logics. In this particular case all children labels (except 1) will be divided into sons (F-labels) and daughters (P-Labels) by inserting $[F]$ or $[P]$ between a label and its formula. Since we prefer models with one flow-time relation $<$, this division serves to indicate the direction of a relation. If $\sigma.i$ is an F-label, it means that $\sigma < \sigma.i$; if $\sigma.i$ is a P-label, it means that $\sigma.i < \sigma$. One should note that so defined sets of labels for multimodal logics are also strongly generated.

One may find some other (medium) labeled systems for multimodal logics, using more complicated labels where the additional information is included in the structure of a label. Such a solution may be found in Bonette and Goré [49] SC for **Kt**, where they keep track of the direction of time flow in each item in their labels. Also Fitting [97] uses similar labels in TS for epistemic logics; each item (except the last) is a pair: a number and a name of an agent who "sees" the next item. Other solutions are also possible, e.g., Marx' system [182] where labels are pairs of natural numbers; we do not present this approach because it is used in the formalization of logics not discussed in this book. Our solution, first presented in [151], is simpler since we keep an information concerning time flow (or generally, a type of accessibility relation) as an extra sign, not contained in the labels themselves. Due to the one-step character of rules we will provide (c.f. next sections; in particular, remarks on alternative rules), it is sufficient to mark the direction only with respect to the last two digits in a label.

Finally, we should say that the technique of Fitting's labels is essentially independent of the kind of deductive system at use; it is rather accidental that usually it goes with tableaux.[5] In this and the next chapters it will be applied to ND systems.

[5] Although not always; for example Tapscott [271] is using essentially Fitting's labels in ND-system.

8.4 Labelled ND-K

In the next three sections we introduce labelled versions of ND system for many modal logics. We still keep KM as our basis but we call the system more generally LND since the proposed solution is in fact format-insensitive. In particular, no strict subderivations are demanded since their role is performed by labels. The reader preferring other ND format may easily adapt the calculus introduced below to other form of realization, or even transform it into S-system. We investigate only LND based on the kind of labels introduced in the preceding section. There are at least three reasons for paying an attention to such a solution:

- The additional technical apparatus of Fitting's labels is kept in reasonable limits, so we may still claim convincingly that ND system of this sort is natural and simple.
- Despite some limitations, it strongly extends the scope of application of ND covering many logics hardly formalizable in standard Fitch's approach.
- It may be easily modified to obtain an analytic version which provides handy decision procedures for many modal logics. We will focus on this question in Chapter 10, where several proof search procedures will be examined.

8.4.1 LND System for K

We start with LND for homogenous multimodal version of **K**. Primitive rules of LND for **K** are divided as usual into two groups.

Inference rules:

($L\alpha E$) $\sigma : \alpha \ / \ \sigma : \alpha_i$, where $i \in \{1, 2\}$
($L\alpha I$) $\sigma : \alpha_1$, $\sigma : \alpha_2 \ / \ \sigma : \alpha$
($L\beta E$) $\sigma : \beta$, $\sigma : -\beta_i \ / \ \sigma : \beta_j$, where $i \neq j \in \{1, 2\}$
($L\beta I$) $\sigma : \beta_i \ / \ \sigma : \beta$, where $i \in \{1, 2\}$
($L\neg\neg$) $\sigma : \neg\neg\varphi \ // \ \sigma : \varphi$
($L\bot I$) $\sigma : \varphi$, $\sigma : -\varphi \ / \ \bot$
($L\bot E$) $\bot \ / \ \sigma : \varphi$, for any σ and φ
($L\pi E$) $\sigma : \pi^i \ / \ \sigma.k : \pi$, where $\sigma.k$ is a new i-label in a derivation
($L\pi I$) $\sigma.k : \pi \ / \ \sigma : \pi^i$, where $\sigma.k$ is any i-label in a derivation
($L\nu E$) $\sigma : \nu^i \ / \ \sigma.k : \nu$, where $\sigma.k$ is any i-label in a derivation

Proof construction rules:

[$LCOND$]: If $X,\ \sigma : -\beta_i \vdash \sigma : \beta_j$, then $X \vdash \sigma : \beta$, where $i \neq j \in \{1, 2\}$

[$LRED$]: If $X,\ \sigma : \varphi \vdash \bot$, then $X \vdash \sigma : -\varphi$

[$LNEC$]: If $X,\ \sigma.k : \top \vdash \sigma.k : \nu$, then $X, \vdash \sigma : \nu^i$, where $\sigma.k$ is a new i-label in a derivation

It is evident that rules for extensional constants do not differ essentially from nonlabelled version introduced in Chapter 2, since no operations on labels are performed (clearly it must be the same label for all premises and conclusion). Such rules will be called *static* (no move beyond given label) in contrast to rules for modal constants, called *transfer rules* because some operation is performed on labels (i.e. a move from one label to another).

Both modal elimination rules are due to Fitting, whereas rules of introduction (including [$LNEC$] which is just a multimodal version of [$L\Box I$]) are essentially from Basin/Matthews/Vigano system, with a slight modification. The lack of r-formulae forces us in [$LNEC$] to introduce, in a subproof, an "empty" assumption $\sigma.k : \top$. It should be noted that the introduction of this assumption is obligatory if we want to close this subproof by [$LNEC$], on the other hand, one should note that this subproof is not strict, in contrast to its counterpart in Fitch's approach. Generally, since all subproofs are ordinary, we again resign from explicit use of any reiteration rule in the realization of LND systems, but we continue to use Jaśkowski's format in the version of Kalish/Montague. Accordingly, the schema of [$LNEC$] looks as follows in this realization:

	$\mathcal{D}$	
i	SHØW:	$\sigma : \nu^i$
$i+1$		$\sigma.k[\ i\ ] : \top$
		.
		.
		.
n		$\sigma.k[\ i\ ] : \nu$

where: $X \subseteq \mathrm{U}(\mathcal{D})$, and $\sigma.k$ is a new i-label.

We may extend this system to cover both local and global deducibility of φ from assumptions. If ψ is a local assumption, then we may always add $1 : \psi$ to a derivation; if ψ is a global assumption, then we may always add to a derivation $\sigma : \psi$ for any label σ which occurs in a derivation.[6]

[6] A justification for such rules may be found in [93].

We do not provide a formal definition of a derivation; it may be done along the lines of the definition from Section 2.5.3 $\vdash_{LND-K} \varphi$ means that there is a proof (a closed derivation) of φ in LND-**K**. Here is an example of such a proof in monomodal **K**:

1	SHØW: $1 : \Box\Diamond p \wedge \Box\Box q \rightarrow \Box\Diamond(p \wedge q)$	$[5, LCOND]$
2	$1 : \Box\Diamond p \wedge \Box\Box q$	*ass.*
3	$1 : \Box\Diamond p$	$(2, L\alpha E)$
4	$1 : \Box\Box q$	$(2, L\alpha E)$
5	SHØW: $1 : \Box\Diamond(p \wedge q)$	$[12, LNEC]$
6	$1.1 : \top$	*m.ass*
7	$1.1 : \Diamond p$	$(3, L\nu E)$
8	$1.1 : \Box q$	$(4, L\nu E)$
9	$1.1.1 : p$	$(7, L\pi E)$
10	$1.1.1 : q$	$(8, L\nu E)$
11	$1.1.1 : p \wedge q$	$(9, 10, L\alpha I)$
12	$1.1 : \Diamond(p \wedge q)$	$(11, L\pi I)$

To demonstrate a completeness of LND-**K** we must only prove the axioms of **K**, which is simple. To show that it is closed under (RG) it is sufficient to observe that whenever we have a proof of φ in LND, then we may transform it into a proof of $\Box_i\varphi$ by suitable rewriting of labels in a proof of φ. As the first two lines of a new proof we write down SHOW: $1 : \Box_i\varphi$ and an assumption $1.1\ [\ i\ ] : \top$; then we insert a proof of φ, in which every label σ is renamed into $1.\sigma$. It is enough to close the main subproof by $[LNEC]$. So it holds:

Theorem 8.1 (Completeness of LND-K) *If* $\models_K \varphi$, *then* $\vdash_{LND-K} \varphi$

Soundness proofs provided for labelled tableau systems (e.g. in [93, 186, 117]) may be applied also for LND. Because they are considerably different from soundness proofs proposed in this book, we rather keep the general strategy applied so far. In order to prove the soundness of our system in the similar way as in Chapter 2 we need however some new notions permitting suitable generalization. Although we demonstrate it in this section only for **K**, the definitions will be stated for any (multi-)modal logic **L** in order to prepare the ground for extensions.

Definition 8.4 (Interpretation and satisfaction) Let X be strongly generated set, and $\mathfrak{M} = \langle \mathcal{W}, \{\mathcal{R}_i\}, V\rangle$ any **L**-model: *an interpretation of* X *in* $\mathfrak{M}$ is a function $\Im$:LAB$(X) \longrightarrow \mathcal{W}$ satisfying condition:

if σ and $\sigma.k$ belong to LAB(X) and $\sigma.k$ is i-label, then $\langle \Im(\sigma), \Im(\sigma.k)\rangle \in \mathcal{R}_i$.
X is **L**-*satisfiable under interpretation* $\Im$, if $\Im(\sigma) \models \varphi$ for every $\sigma : \varphi \in X$
X is **L**-*satisfiable*, if it is **L**-satisfiable under some interpretation $\Im$.

Let us introduce a generalized notion of labelled (local) entailment:

$$\models_L^{LAB} \subseteq \mathcal{P}(LFOR) \times LFOR.$$

Definition 8.5 (Labelled Entailment) Let Y be strongly generated set, and $X \cup \{\tau : \varphi\} \subseteq Y$, then: $X \models_L^{LAB} \tau : \varphi$ iff for any **L**-model and interpretation, if X is **L**-satisfiable, then $\tau : \varphi$ is also **L**-satisfiable under the same interpretation, namely:

$$\forall_{\mathfrak{M} \in MOD(\mathbf{L})} \forall_{\Im} \text{ (if } \forall_{\sigma:\psi \in X} \Im(\sigma) \models \psi \text{ , then } \Im(\tau) \models \varphi)$$

We may now redefine key semantic properties of rules in terms of labelled entailment.

Definition 8.6 (Correctness of rules) Let Z be a strongly generated set, and $X \cup Y \cup \{A, B\} \subseteq Z$, then:

1. A rule of inference $X \ / \ A$ is **L**-(LAB)normal iff, $X \models_L^{LAB} A$
2. A rule of proof construction "if $X \vdash A$, then $Y \vdash B$" is **L**-(LAB)normality preserving iff, whenever $X \models_L^{LAB} A$, then $Y \models_L^{LAB} B$

It is easy to demonstrate the following two lemmata:

Lemma 8.1 *Every inference rule in LND-***K** *is* **K**-*(LAB)normal*

Proof Let's consider the case of $(L\pi I)$. Assume indirectly that for some model and interpretation we have $\Im(\sigma.k) \models \pi$, but $\Im(\sigma) \not\models \pi^i$, where $\sigma.k$ is an i-label. Since $\Im(\sigma) \not\models \pi^i$, then no w which satisfies π is $\mathcal{R}_i$-accessible from $\Im(\sigma)$. But $\Im(\sigma.k)$ is such a point – contradiction, so $(L\pi I)$ is **K**-(LAB)normal.

Other cases are similar. ■

Lemma 8.2 *Every rule of proof construction in LND-***K** *is* **K**-*(LAB)normality preserving.*

PROOF We consider the case of $[LNEC]$. By assumption, $X, \sigma.k : \top \models_K^{LAB} \sigma.k : \nu$ and for any $\tau : \varphi \in X$, $\Im(\tau) \vDash \varphi$ under any interpretation $\Im$; we must show that $\Im(\sigma) \vDash \nu^i$. It means showing that $w \vDash \nu$, for any w such that $\mathcal{R}_i(\Im(\sigma), w)$. Let w be an arbitrary such world, since $\sigma.k$ does not belong to $LAB(X)$, then we may extend $\Im$ letting $w = \Im(\sigma.k)$. Clearly, $w \vDash \top$ (it holds for every point) so, by assumption, $w \vDash \nu$ which means that $X \models_K^{LAB} \sigma : \nu^i$. ■

Both lemmata are sufficient for demonstrating soundness of multimodal LND-**K** along the lines of the proof given in Chapter 2. So we have:

Theorem 8.2 (Soundness of LND-K) *If* $\vdash_{LND-K} \varphi$ *, then* $\models_K \varphi$

Remark 8.3 (Other proof construction rules) One may note that instead of inference rule $(L\pi I)$ we may use suitable proof construction rule on a pair with $[LNEC]$, similarly as we did in standard ND. We mean the following one:

$[LPOS]$: If $X,\ \sigma : \pi_1^i,\ \sigma.k : \pi_1 \ \vdash\ \sigma.k : \pi_2$, then $X,\ \sigma : \pi_1^i \ \vdash\ \sigma : \pi_2^i$, where $\sigma.k$ is a new i-label in a derivation.

A system with this rule is equivalent to our official one with $(L\pi I)$. That every subproof closed by $[LPOS]$ is eliminable in favor of $(L\pi I)$ (preceded by the application of $(L\pi E)$ on $\sigma : \pi_1^i$ introducing a new label $\sigma.k$) is obvious. In the other direction, note that every application of $(L\pi I)$ on, e.g. $\sigma.k : \varphi$, must have been preceded by the application of $(L\pi E)$ on, e.g. $\sigma : \Diamond\psi$, which introduced $\sigma.k$. Let $\mathcal{D}$ be the part of a proof ending with $\sigma : \Diamond\psi$ and $\mathcal{D}'$ be the part of a proof with the first line $\sigma.k : \psi$ (the conclusion of this $(L\pi E)$ application) and the last line $\sigma.k : \varphi$ (the premise of analyzed application of $(L\pi I)$). We may take $\mathcal{D}$, add "SHØW: $\sigma : \Diamond\varphi$" and put $\mathcal{D}'$ in a box immediately below. One may easily check that it is a correct application of $[LPOS]$ with $\psi = \pi_1$ and $\varphi = \pi_2$. Clearly, it is also possible to define a counterpart of $[\Diamond E]$ (cf. Section 8.2.2) of Basin/Matthews/Vigano in terms of Fitting's labels. We leave it to the reader.

Other combinations are also possible. Tapscott [271] presented a labelled ND-systems for **T, B, S4, S5**. In his system there are two specific transfer rules for introducing modal assumption and for closing a subderivation which may be stated (to comply with our form of presentation) as follows:

(*Advancement*) If there is U-formula $\sigma : \Diamond\varphi$, then start a subderivation with an assumption $\sigma.k : \varphi$, where $\sigma.k$ is new in a derivation.

[*Impossibilitation*] If $\bot$ was inferred in a subderivation with assumption $\sigma.k : \varphi$, where $\sigma.k$ is new in a derivation, then close this subderivation introducing $\sigma : \neg\Diamond\varphi$ as a new U-formula.

It is easy to note that they work together, hence they are just realization of one proof construction rule which on the level of a calculus may be stated as follows:

[*Impossibility I*] If $X,\ \sigma : \Diamond\varphi,\ \sigma.k : \varphi \vdash \bot$, then $X,\ \sigma : \Diamond\varphi \vdash \sigma : \neg\Diamond\varphi$

The specific shape of Tapscott's rules follows from the fact that they are modelled directly for a system which is used as ND realization of some proof-search procedure. ♣

8.5 Other Logics

This approach is quite extensive – in fact much more extensive than tableau systems from [93, 185] or [117] may suggest. In this section we adopt ready rules for basic normal logics from labelled TS's to LND and discuss some variations of them. Direct extensions to some temporal logics and to first-order logics will be also included. In the next sections and chapters more substantial modifications will be provided to capture also weak modal logics and (temporal) logics of linear frames.

8.5.1 Basic Normal Logics

The following list contains inference rules sufficient to formalization of all basic normal logics in a modular way. They come from Massacci [185, 186] where they are stated for monomodal logics. Except (LT) and (LD) all other are transfer rules. Because every rule has in premises some ν-formula we will call them collectively as ν-rules.

(LD)	$\sigma : \Box_i\varphi \ / \ \sigma : \Diamond_i\varphi$ and $\sigma : \neg\Diamond_i\varphi \ / \ \sigma : \neg\Box_i\varphi$
(LT)	$\sigma : \nu^i \ / \ \sigma : \nu$
($L4$)	$\sigma : \nu^i \ / \ \sigma.k : \nu^i$, where $\sigma.k$ is any i-label in a derivation
(LB)	$\sigma.k : \nu^i \ / \ \sigma : \nu$, where $\sigma.k$ is any i-label in a derivation
($LB4$)	$\sigma.k : \nu^i \ / \ \sigma : \nu^i$, where $\sigma.k$ is any i-label in a derivation
($L5\Box$)	$1.k : \nu^i \ / \ 1 : \Box_i\nu^i$, where $1.k$ is any i-label in a derivation
($L5.4$)	$\sigma : \nu^i \ / \ \sigma.k : \nu^i$, where $\mid \sigma \mid >1$ and $\sigma.k$ is any i-label in a derivation

Formalizations of all basic normal logics may be obtained in the following way:

D	LND-**K**$\cup\{(D)\}$	**KD5**	LND-**K5**$\cup\{(D)\}$
T	LND-**K**$\cup\{(T)\}$	**S4**	LND-**T**$\cup\{(4)\}$
K4	LND-**K**$\cup\{(4)\}$	**B**	LND-**T**$\cup\{(B)\}$
KB	LND-**K**$\cup\{(B)\}$	**KB4**	LND-**KB**$\cup\{(4),(B4)\}$
K5	LND-**K**$\cup\{(B4),(5\Box),(5.4)\}$	**K45**	LND-**K4**$\cup\{(5\Box),(5.4)\}$
KD4	LND-**D**$\cup\{(4)\}$	**KD45**	LND-**K45**$\cup\{(D)\}$
KDB	LND-**D**$\cup\{(B)\}$	**S5**	LND-**S4**$\cup\{(B4)\}$

Proof of soundness requires only showing that all the rules are **L**-(LAB)normal in suitable logic; one can find them e.g. in Goré [117]. For completeness we must only prove suitable axioms of **L**, we leave it to the reader. Hence we state:

Theorem 8.3 (Adequacy of LND-L) *For every basic normal (homogenous multimodal) logic* **L** *it holds:* $\models_L \varphi$ *iff* $\vdash_{LND-L} \varphi$

For the sake of illustration we prove 5 in LND-**K5**.

1	SHØW: $1 : \Diamond p \rightarrow \Box\Diamond p$	$[4, LCOND]$
2	$1 : \Diamond p$	$ass.$
3	$1.1 : p$	$(2, L\pi E)$
4	SHØW: $1 : \Box\Diamond p$	$[6, LNEC]$
5	$1.2 : \top$	$m.ass.$
6	SHØW: $1.2 : \Diamond p$	$[9, LRED]$
7	$1.2 : \neg\Diamond p$	$ass.$
8	$1 : \neg\Diamond p$	$(7, LB4)$
9	$\bot$	$(2, 8, L\bot I)$

8.5.2 Regular Basic Logics

This approach may be extended to all basic regular logics we have considered. It is enough to change a bit two rules. In $(L\pi E)$ we should add a proviso that there is some ν-formula with label σ in a derivation,[7] or restate the rule introducing this formula as an additional premise:

$(L\pi E')$ $\sigma : \nu^i,\ \sigma : \pi^i\ /\ \sigma.k : \pi$, where $\sigma.k$ is a new i-label in a derivation

[7] This is essentially the solution of Fitting [93].

Similarly for $[LNEC]$ we must have some (other) ν^i as U-formula in outer derivation if we want to close a subproof by this rule. Because of that, instead of using $\top$ as a premise, we make use of immediate component of respective ν^i. So we admit the following rule of proof construction:

$[LNEC_R]$: If $X,\ \sigma : \nu_1^i,\ \sigma.k : \nu_1 \ \vdash\ \sigma.k : \nu_2$, then $X,\ \sigma : \nu_1^i \ \vdash \sigma : \nu_2^i$, where $\sigma.k$ is a new i-label in a derivation.

One may easily note that these two rules are correct in **R**; a careful reader can make suitable adjustments in proofs of adequacy stated above.[8] Note that the notion of interpretation does not change, we only restrict consideration to regular models. All other rules are correct, including rules for serial, reflexive and transitive extensions. As for completeness one may show that we can simulate every application of (RM) similarly as we did it for (RG), by embedding a proof $\mathcal{D}$ of $\varphi \to \psi$ into additional two boxes and renaming labels of $\mathcal{D}$. A new proof-schema looks like this:

1	SHØW: $1 : \Box\varphi \to \Box\psi$	$[3, LCOND]$
2	$1 : \Box\varphi$	*ass.*
3	SHØW: $1 : \Box\psi$	$[n, LNEC_R]$
4	$1.1 : \varphi$	*ass.*
5	SHØW: $1.1 : \varphi \to \psi$	
	$\mathcal{D}'[\sigma/1.\sigma]$	
n	$1.1 : \psi$	$(4, 5, L\beta E)$

where $\mathcal{D}'[\sigma/1.\sigma]$ is an exact copy of $\mathcal{D}$ but with every label σ replaced with $1.\sigma$.

As a result we have an adequacy theorem for LND-**L**, where **L** is one of the regular basic logics described in Chapter 5.

8.5.3 Temporal Logics

A formalization of temporal logics (and other interactive multimodal logics) requires additional rules. Indrzejczak [151] provides the formalization of **Kt**, where except rules for bimodal **K** with $i \in \{P, F\}$, we have the following ν-rules:

[8] By the way, in regular logics we may even get rid of $(L\pi E')$ if we admit $[LPOS]$ from Remark 8.3. as primitive, but we left the proof of this fact to the reader.

$(LB\nu^i)$ $\sigma.k : \nu^i \ / \ \sigma : \nu$, where $\sigma.k$ is any j-label in a derivation and $i \neq j \in \{P, F\}$

For **Kt4** we must clearly add:

$(LB4\nu^i)$ $\sigma.k : \nu^i \ / \ \sigma : \nu^i$, where $\sigma.k$ is any j-label in a derivation and $i \neq j \in \{P, F\}$

We may also add rules for seriality for both modalities or only one of them.

Here is an example of a proof in LND-**Kt**:

1	SHØW: $1 : p \wedge Fq \to F(Pp \wedge q)$	$[11, LRED]$
2	$1 : \neg(p \wedge Fq \to F(Pp \wedge q))$	*ass.*
3	$1 : p \wedge Fq$	$(2, L\alpha E)$
4	$1 : \neg F(Pp \wedge q)$	$(2, L\alpha E)$
5	$1 : p$	$(3, L\alpha E)$
6	$1 : Fq$	$(3, L\alpha E)$
7	$1.1[F] : q$	$(6, L\pi^F E)$
8	$1.1[F] : \neg(Pp \wedge q)$	$(4, L\nu^F E)$
9	$1.1[F] : \neg Pp$	$(7, 8, L\beta E)$
10	$1 : \neg p$	$(9, LB\nu^P)$
11	$\perp$	$(5, 10, L\perp I)$

Showing (LAB)normality of these rules demands a small modification of the definition of interpretation:

Definition 8.7 (Temporal Interpretation) Let X be strongly generated set and $\mathfrak{M} = \langle \mathcal{T}, <, V \rangle$ be any **L**-model: an interpretation of X in $\mathfrak{M}$ is a function $\Im$:LAB(X)$\longrightarrow \mathcal{T}$ satisfying condition: if σ and $\sigma.k$ belong to LAB(X), then:

(a) if $\sigma.k$ is F-label (a son), then $\Im(\sigma) < \Im(\sigma.k)$.

(b) if $\sigma.k$ is P-label (a daughter), then $\Im(\sigma.k) < \Im(\sigma)$.

With this adjustment we may easily prove soundness and completeness of this formalization similarly as in the preceding cases. Hence LND-**L** is adequate for **L**$\in \{$**Kt**, **Kt4**, **KtD**, **KtD4**$\}$. LND-systems for logics of linear time will be considered in the next Chapter.

8.5.4 Some Other Logics

We briefly discuss some other extensions which are easily obtainable in this framework. First, it is easy to formalize all the logics considered in Section 7.1.1 which are axiomatized with the help of $\Box D$, $\Box T$, $\Box B$ or $\Box 4$ (cf. Massacci [185]). It is enough to add to rules $(LD), (LT), (LB)$ and $(L4)$ a side condition that they are applicable if $\mid \sigma \mid > 1$.

With the help of Fitting's labels one may formalize also logics of provability. Goré [117] proposes the following two rules[9]:

$(L\pi G)$	$\sigma : \neg\Box\varphi \ / \ \sigma.k : \neg\varphi \ , \ \sigma.k : \Box\varphi$, where $\sigma.k$ is a new label
$(L\pi Grz)$	$\sigma : \neg\Box\varphi \ / \ \sigma.k : \neg\varphi \ , \ \sigma.k : \Box(\varphi \to \Box\varphi)$, where $\sigma.k$ is a new label

After addition of $(L\pi G)$ to LND-**K4** and $(L\pi Grz)$ to LND-**S4** they yield adequate LND systems for **G** and **Grz**. Note that they are not additional ν-rules but rather alternative rules for $(L\pi E)$.

An extension to first-order logics is also unproblematic. Note that all rules for quantifiers are local (no transition to other states is involved), so it is enough to add a label σ to premises and conclusions of all quantifier inference rules considered in Chapter 2 in order to obtain suitable static rules. Similarly for proof construction rules we must add the same label to all formulae displayed in the schema of the rule. For example, labelled counterparts of both proof construction rules used in formalization of KMGP' look like this:

$[LF\exists E^{p'}]$	If $X,\ \sigma : \exists x\varphi,\ \sigma : Ea \wedge \varphi[x/a]\ \vdash\ \sigma : \psi$, then $X,\ \sigma : \exists x\varphi\ \vdash\ \sigma : \psi$, provided a is a parameter with no occurrence in φ, ψ and undischarged assumptions
$[LFUNIV^{p'}]$	If $X,\ \sigma : Ea\ \vdash\ \varphi[x/a]$, then $X\ \vdash\ \sigma : \forall x\varphi$, provided a is a parameter with no occurrence in φ and undischarged assumptions

Fitting in [93] (or with Mendelsohn in [96]) applies a different strategy in labelled tableaux for logics of varying domains. They provide distinct sort of parameters for each label, simply by indexing parameters with labels. But in case we use existence predicate this is superfluous, since instead of introducing parameters of the form a_σ as arguments of predicates we

[9] The former is in fact present in [93].

use $\sigma : Ea$ to convey exactly the same information. The advantage of an approach represented in this book is that we have a constant set of quantifier rules established once and for all in Chapter 2 and used throughout with no changes in standard modal ND (cf. Chapter 6) and with only cosmetic change (an addition of a label) in LND. As a consequence, adequacy result holds for LND formalizations of all sorts of **QML** introduced in Chapter 5.

The list of possible extensions is not complete but some cases, like weak modal logics or logics of linear frames, require more detailed treatment because of substantial modifications of rules or interpretation of labels. We describe the former in the next section and the latter in the next Chapter; this section we end up with few remarks concerning alternative rules for basic logics.

Remark 8.4 (Serial Logics) Instead of a static rule (LD) one may use Fitting's transfer rule:

(LD') $\sigma : \nu^i$ / $\sigma.k : \nu$, where $\sigma.k$ is any i-label in a derivation, including new ones.

We prefer (LD) because it makes a completeness proof for analytic version provided in Chapter 10 easier. ♣

Remark 8.5 (Transitive Logics) Also the original Fitting's rules used to formalization of transitive logics are different. He has just one such a rule for every logic of this kind:

$(L4')$ $\sigma : \nu^i$ / $\sigma.\tau : \nu$, where $\mid \tau \mid \geq 1$

With this rule we can make "long jumps" from one point to the other which is accessible in long distance, in one step. It is also stronger than $(L4)$ since it covers also $(L\nu E)$ as a particular case (this rule is in fact redundant in **S4** because of the presence of (LT)). The rules which we prefer are due to Massacci [185, 186] and, in contrast to original Fitting's rules, are "single step" rules (i.e. we can go only to the point which is an immediate neighbour). Massacci provides several arguments in favor of single step rules concerning their modularity and better computational behaviour. We are not going to repeat them, but rather add one more: Fitting's rule is inconvenient for multimodal logics. Note, that the mere fact that some θ is i-label and an extension of σ, does not guarantee that it is i-accessible from σ. For example: let $\theta = \sigma.k.n$ be i-label and $\sigma.k$ be

j-label for some $j \neq i$. In such a situation, a deduction of $\theta : \varphi$ from $\sigma : \Box_i\varphi$ by $(L4')$ would be incorrect. If we formalize multimodal logics with the help of labels where only the transition between the label and its parent-label is classified (i.e. it is indicated what accessibility relation is involved), then only "single step" rules of Massacci are admissible. Of course, it is possible to use more complicated labels, where for each digit in a label we keep track of the corresponding relation of accessibility; such a solution was applied e.g. by Goré and Bonette [49]. ♣

Remark 8.6 (Euclidean Logics) It is of some interest why the set of rules for Euclidean logics (except **S5**) is so complicated. In case of monomodal logics we may obtain a suitable formalization much easier; only one rule is needed:

$(L5)$ $\sigma : \nu^i$ / $\tau : \nu$, where $\mid \sigma \mid >1$ and $\mid \tau \mid >1$

This rule is also of "long jump", so if we have at least two different modalities, then it is incorrect because we have no guarantee that all other labels are i-accessible from σ. For example, we have a model $\mathfrak{M} = \langle\{1, 1.1, 1.2, 1.3\}, \mathcal{R}_a, \mathcal{R}_b\rangle$ with two Euclidean relations, where $\mathcal{R}_a(1, 1.1)$, $\mathcal{R}_b(1,\ 1.2)$ and $\mathcal{R}_b(1,\ 1.3)$. Then by Euclidean property we have $\mathcal{R}_b(1.3,\ 1.2)$ but $\mathcal{R}_b(1.3,\ 1.1)$ does not hold. Yet $(L5)$ allows us to infer from $1.3\ [\ b\ ] : \Box_b\varphi$ both $1.2\ [\ b\ ] : \varphi$ (which is correct) and $1.1\ [\ a\ ] : \varphi$ which is incorrect. Because of that, in multimodal Euclidean logics, we must also use only "one step" transfer rules.

Massacci [186] provided also alternative formalizations of Euclidean transitive logics based on the following rule used instead of $(L4)$:

$(L4\pi)$ $\sigma.k : \pi^i$ / $\sigma : \pi^i$

Also, instead of $(L5\Box)$ he considered a more general rule:

(Cxt) $\sigma.k : \nu^i$ / $\sigma : \Box_i\nu^i$ with $\sigma = 1$

which provides a formalization of McCarthy's logic of contextual reasoning (added to **K**), but for Euclidean logics, a restricted version is sufficient. ♣

We have provided LND-system in full form with a variety of rules for introduction and elimination of constants. One should note however that it may be restricted to an analytic form, similarly as ND-**CPL** in Chapter 4. In particular, it is easy to obtain a labelled counterpart of AND1 because for most considered logics such restricted ND is sufficient to simulate every tree in labelled TS's. We return to this question in Chapter 10.

8.6 LND for Weak Modal Logics

Fitting's style labels are directly modeled on relational semantics. It is an interesting question whether this form of labelling can be extended to some congruent and monotonic modal logics, which are not characterizable by Kripke frames. But neighbourhood frames are also a kind of more general relational semantics so it should be possible. For our needs it is important that the domains in neighbourhood frames are of the same kind as in Kripke frames which is sufficient to apply labels in the same way, as it is done in stronger logics – as names for states in a model. The main problem is how to apply them to cover logics with different interpretation of modalities.

In particular, in normal and regular logics labels correspond in a straightforward manner to the structure of an attempted falsifying model. It makes this method very handy for defining procedures of model extraction. Unfortunately, neigbourhood functions are not as easy to handle as accessibility relations. But this difficulty may be easily overcome. One possible solution was provided by the author in [154] for TS; in what follows we apply our approach to LND systems.

Similarly as in Fitting's style systems, the label is not only a name of a state in a model but additionally its structure encodes the place of this state in a falsifying model we are trying to build. The difference lies in the sense which is captured by a label of the form $\sigma.k$. In our system it means that for some $\sigma : \Box\varphi$ we have $\sigma.i \in \|\varphi\|$. We continue using generalized notation good for multimodal logics although [154] deals only with monomodal case. As far as we are talking about homogenous logics there is no difference.

Let us consider the following inference rules taken from [154][10]:

(LM)	$\sigma : \nu^i,\ \sigma : \pi^i\ /\ \sigma.k : \nu,\ \sigma.k : \pi$, where $\sigma.k$ is a new label
(LMD)	$\sigma : \nu_1^i,\ \sigma : \nu_2^i\ /\ \sigma.k : \nu_1,\ \sigma.k : \nu_2$, where $\sigma.k$ is a new label
$(LM4)$	$\sigma : \nu^i,\ \sigma : \pi^i\ /\ \sigma.k : \nu^i,\ \sigma.k : \pi$, where $\sigma.k$ is a new label
$(LM5)$	$\sigma : \pi_1^i,\ \sigma : \pi_2^i\ /\ \sigma.k : \pi_1^i,\ \sigma.k : \pi_2$, where $\sigma.k$ is a new label
(LMB)	$\sigma : \pi_1,\ \sigma : \pi_2^i\ /\ \sigma.k : \pi_1^i,\ \sigma.k : \pi_2$, where $\sigma.k$ is a new label
$(LMD4)$	$\sigma : \nu_1^i,\ \sigma : \nu_2^i\ /\ \sigma.k : \nu_1,\ \sigma.k : \nu_2^i$, where $\sigma.k$ is a new label
$(LMD5)$	$\sigma : \nu^i,\ \sigma : \pi^i\ /\ \sigma.k : \nu,\ \sigma.k : \pi^i$, where $\sigma.k$ is a new label

[10] Except (LMB), since B-logics were not considered in [154] because of the lack of analytic formalization.

Note that in all cases we have in fact two rules, since there are two conclusions stated. In practise we should write both of them in every application of a rule, although it is not, strictly speaking, obligatory.

To obtain LND-**M** we need to add only one inference rule (LM) to a labelled version of rules for **CPL**. None of the four modal rules from Section 8.4. characterizing **K** is needed; in fact, they are incorrect in **M**. For other basic monotonic logics we must add some of the rules from the above set – every rule (LMA) corresponds in a modular fashion to a suitable axiom A. Rules $(LMD4)$ and $(LMD5)$ are in fact not necessary for completeness, but they are necessary for completeness of an analytic versions of serial transitive ($(LMD4)$ is needed) and serial Euclidean ($(LMD5)$ is needed) logics, since they are modeled on SC-rules from Indrzejczak [152] required for the proof of cut elimination (cf. a discussion in Section 7.2). To obtain formalizations of reflexive logics we need simply a static rule (LT),[11] moreover, we may add $(L\pi E)$ to all these ND systems and obtain in this way formalizations of all basic monotonic logics with necessitation (or Gödel's) rule (RG) – 30 logics in total.

Completeness of our LND systems for monotonic logics is easy to establish. A simulation of (RM) with the help of (LM) is similar as in case of regular logics (by twice boxing of a given proof and renaming labels). Proofs of interdefinability of $\Diamond$ and $\Box$, and of all axioms, with the help of suitable rules is straightforward. In order to prove soundness we must first change a bit definitions of interpretation and satisfiability.

Definition 8.8 (Neighbourhood Interpretation) Let X be a set of labelled formulae and $\mathfrak{M} = \langle \mathcal{W}, \mathcal{N}, \mathcal{V} \rangle$ a neighbourhood **L**-model. X is satisfiable in $\mathfrak{M}$ if there is an interpretation function $\Im$:LAB$(X) \longrightarrow \mathcal{W}$ such that:

- if σ and $\sigma.k$ belong to LAB(X), then $\Im(\sigma.k) \in \|\varphi\|$ for some $\sigma : \Box\varphi \in X$
- if $\sigma : \varphi \in X$, then $\Im(\sigma) \vDash \varphi$.

Lemma 8.3 *Every inference rule for monotonic logics is (LAB)normal for respective logic.*

Proof We will demonstrate two cases: (LM) and $(LMD5)$:

[11] In fact, we may also use a static rule (LD) stated above for normal and regular logics instead of a transfer rule from the list.

Assume that $\sigma : \Box\varphi$ and $\sigma : \neg\Box\psi$ are satisfied, i.e. $\Im(\sigma) \vDash \Box\varphi$ and $\Im(\sigma) \vDash \neg\Box\psi$ in some model. By definition of a model we have that $\|\varphi\| \in \mathcal{N}(\Im(\sigma))$ and $\|\psi\| \notin \mathcal{N}(\Im(\sigma))$. By condition (m) this implies $\|\varphi\| \nsubseteq \|\psi\|$, so there is a world w' such that $w' \vDash \varphi$ and $w' \nvDash \psi$. We can extend $\Im$ to $\Im'$ by letting $\Im'(\sigma.k) = w'$, since $\sigma.k$ is not in LAB(X) (it is a new label introduced by an application of (LM). Then $\Im'$ is also an interpretation satisfying both $\sigma.k : \varphi$ and $\sigma.k : \neg\psi$.

In case of $(LMD5)$ assume that $\sigma : \Box\varphi$ and $\sigma : \neg\Box\psi$ are satisfied. So, as in the previous case, we have that $\|\varphi\| \in \mathcal{N}(\Im(\sigma))$ and $\|\psi\| \notin \mathcal{N}(\Im(\sigma))$. By condition (5), we have $\{\Im(\sigma') : \|\psi\| \notin \mathcal{N}(\Im(\sigma'))\} \in \mathcal{N}(\Im(\sigma))$ which means that $\{\Im(\sigma') : \Im(\sigma') \nvDash \Box\psi\} \in \mathcal{N}(\Im(\sigma))$, which means that $\|\neg\Box\psi\| \in \mathcal{N}(\Im(\sigma))$. By condition (d) $\|\neg\varphi\| \notin \mathcal{N}(\Im(\sigma))$. By condition (m) $\|\neg\Box\psi\| \nsubseteq \|\neg\varphi\|$, so there is w' such that $w' \vDash \neg\Box\psi$ and $w' \nvDash \neg\varphi$. We can extend $\Im$ to $\Im'$ by letting $\Im'(\sigma.k) = w'$, for new $\sigma.k$ introduced by $(LMD5)$. Then $\Im'$ is also an **MD5**-interpretation satisfying both $\sigma.k : \varphi$ and $\sigma.k : \neg\Box\psi$. ■

(LAB)normality of all the rules (with respect to suitable classes of models) is enough to provide soundness proof for our systems, so we have:

Theorem 8.4 (Adequacy of LND-L) *For all basic monotonic (homogenous multimodal) logics* **L** *(including those closed under (RG)) it holds:* $\vDash_L \varphi$ *iff* $\vdash_{LND-L} \varphi$

One may note that we can obtain formalizations of suitable regular logics just by addition of $(L\nu E)$ to any LND system for monotonic logic without $(LMB), (LM5), (L\pi E)$. That these three rules are enough to obtain normal logics if $(L\nu E)$ is present, is obvious from the previous considerations.

The proposed formalization of monotonic basic logics is simple in practice, and may be easily used to obtain complete analytic versions in many cases (Indrzejczak [154] provides decision procedures on the basis of labelled TS for 18 monotonic basic logics). But from the standpoint of architecture of ND-systems it may seem worse than formalization provided in the preceding sections. Clearly, we may obtain ND-system for **M** of more ordinary character if instead of (LM) we use two proof construction rules: $[LNEC]$ in the version for regular logics, and $[LPOS]$ from Remark 8.3. We do not justify our claim; that it is correct should be evident for any careful reader of Chapter 6 (Section 6.5), since these rules are exact labelled counterparts of rules stated there (in particular, the lack of $(L\nu E)$ plays the function of the lack of reiteration into strict subproofs). So such proof construction rules of LND just simulate suitable rules of standard ND. Interested reader may consider similar transformation of other inference rules into proof construction rules in accordance with the recipe stated in Section 6.5.

It is evident that our approach to LND for monotonic modal logics is based on labelled TS from [154], and the latter is modeled on standard SC rules presented in Chapter 6. The same applies to congruent logics, so one may first recall these SC-rules to grasp the idea intuitively. All SC-rules (except (T)) for this class of basic logics have two premises; their TS counterparts in [154] are (binary-)branching rules of the form:

(LE_2) $\sigma : \Box\varphi,\ \sigma : \neg\Box\psi\ /\ \sigma.k : \varphi,\ \sigma.k : \neg\psi\ \mid\ \sigma.k : \neg\varphi,\ \sigma.k : \psi$
(LD_2) $\sigma : \Box\varphi,\ \sigma : \Box\psi\ /\ \sigma.k : \varphi,\ \sigma.k : \psi\ \mid\ \sigma.k : \neg\varphi,\ \sigma.k : \neg\psi$
$(L4_2)$ $\sigma : \Box\varphi,\ \sigma : \neg\Box\psi\ /\ \sigma.k : \Box\varphi,\ \sigma.k : \neg\psi\ \mid\ \sigma.k : \neg\Box\varphi,\ \sigma.k : \psi$
$(L5_2)$ $\sigma : \neg\Box\varphi,\ \sigma : \neg\Box\psi\ /\ \sigma.k : \neg\Box\varphi,\ \sigma.k : \neg\psi\ \mid\ \sigma.k : \Box\varphi,\ \sigma.k : \psi$
(LB_2) $\sigma : \neg\varphi,\ \sigma : \neg\Box\psi\ /\ \sigma.k : \neg\Box\varphi,\ \sigma.k : \neg\psi\ \mid\ \sigma.k : \Box\varphi,\ \sigma.k : \psi$

where $\sigma.k$ is a new label in all cases

We have already shown how to simulate binary branching rules in ND (Chapter 4) but also analyzed the problem of transformation of such rules in ND for modal logics in case they lose some data. In the context of labelled systems this problem disappears and we may apply our technique of converting branching rules into proof construction rules with no difficulty. We will do it in one step. First, all the rules are instances of just one schema (where $\sigma.k$ is a new label, and A is one of the $\{E, D, 4, 5, B\}$):

(LA_2) $\sigma : \varphi,\ \sigma : \psi\ /\ \sigma.k : \varphi',\ \sigma.k : \psi'\ \mid\ \sigma.k : -\varphi',\ \sigma.k : -\psi'$

It is converted into a schema of proof construction rule:

$[LA_2]$ If $X,\ \sigma : \varphi,\ \sigma : \psi,\ \sigma.k : -\varphi' \wedge -\psi' \vdash \bot$, then $X,\ \sigma : \varphi,\ \sigma : \psi \vdash \sigma.k : \varphi' \wedge \psi'$

where $\sigma.k$ is a new label and $\wedge$ was used to combine two pairs of conclusions of (LA_2)

It is characteristic for all these rules that we introduce immediately S-formula and the corresponding assumption for performing $[LA_2]$ on the basis of two modal U-formulae present in the current subderivation (except (LB_2), where one formula is not necessarily modal). To obtain LND-**E** one must add only $[LE_2]$ to classical basis; for remaining ones also other instances of $[LA_2]$ are necessary (and for congruent logics axiomatized with T also an inference rule (LT)). To cover congruent logics closed under (RG)

one should add $(L\pi E)$, similarly as in monotonic case. We leave the details of a formulation suitable for realization in KM, providing only a diagram of application of $[LA_2]$ for illustration:

	X	
i	$\sigma : \varphi$	
	$\vdots$	
j	$\sigma : \psi$	
	$\vdots$	
k	SHØW: $\sigma.k : \varphi' \wedge \psi'$	$[n, LA_2]$
$k+1$	$\sigma.k : -\varphi' \wedge -\psi'$	*ass.*
	$\vdots$	
n	$\perp$	

We leave to the reader proofs of (LAB)normality preservation of all these rules in respective neighborhood models (use the definition of interpretation formulated for monotonic logics) required for soundness proof. Completeness demands proofs of axioms and closure under (RE). The following schema shows how application of $[LE_2]$ simulates (RE):

1	SHØW: $1 : \Box\varphi \rightarrow \Box\psi$	$[3, LCOND]$
2	$1 : \Box\varphi$	*ass.*
3	SHØW: $1 : \Box\psi$	$[k+1, LRED]$
4	$1 : \neg\Box\psi$	*ass.*
5	SHØW: $1.1 : \varphi \wedge \neg\psi$	$[i+1, LE_2]$
6	$1.1 : \neg\varphi \wedge \psi$	*ass.*
7	$1.1 : \neg\varphi$	$(6, L\alpha E)$
8	$1.1 : \psi$	$(6, L\alpha E)$
9	SHØW: $1.1 : \varphi \leftrightarrow \psi$	
	$\mathcal{D}[\sigma/1.\sigma]$	
i	$1.1 : \varphi$	$(8, 9)$
$i+1$	$\perp$	$(7, i, L\perp I)$
$i+2$	$1.1 : \varphi$	$(5, L\alpha E)$
$i+3$	$1.1 : \neg\psi$	$(5, L\alpha E)$
$i+4$	SHØW: $1.1 : \varphi \leftrightarrow \psi$	
	$\mathcal{D}[\sigma/1.\sigma]$	
k	$1.1 : \psi$	$(i+2, i+4)$
$k+1$	$\perp$	$(i+3, k, L\perp I)$

Here we must use the assumed proof (with renamed labels) of $\varphi \leftrightarrow \psi$ twice to obtain a result. The proof of $\Box\psi \rightarrow \Box\varphi$ is analogical, hence we got a proof of $\Box\varphi \leftrightarrow \Box\psi$ on the basis of an earlier proof of $\varphi \leftrightarrow \psi$ which shows that our formalization is complete. Thus we obtain:

Theorem 8.5 (Adequacy of LND-L) *For all basic congruent (homogenous multimodal) logics* **L** *(including those closed under (RG)) it holds:* $\models_L \varphi$ *iff* $\vdash_{LND-L} \varphi$

8.7 MRND Systems with Labels

Fitting's approach may be extended to clausal ND in at least two ways. The simplest way of combining RND with labels is to admit that in all contexts we deal with labelled formulae; in particular, clauses are built from formulae with labels. Such an application of labels will be called local and suitable systems, called LLRND (locally labelled RND), will be considered first. But it is also possible to add labels to whole clauses.[12] Such labels have global character since they qualify the set of formulae, and the system of this sort will be called GLRND (globally labelled RND). For keeping presentation in reasonable bounds we restrict our attention to propositional basic normal and the the simplest temporal logics, although this approach may be extended further. We invite the reader to experimentation.

8.7.1 Local Labelling

One should note that in this subsection Γ, Δ denote clauses built from labelled formulae and X, Y denote the sets of so defined clauses. With this proviso we may redefine all RND inference rules for classical connectives in a straightforward way:

(LLW)	$\Gamma \ / \ \Gamma,\ \sigma : \varphi$
$(LLRes)$	$\Gamma,\ \sigma : \varphi \ ; \ \Gamma,\ \sigma : -\varphi \ / \ \Gamma$
$(LL\neg\neg)$	$\Gamma,\ \sigma : \neg\neg\varphi \ // \ \Gamma,\ \sigma : \varphi$
$(LL\alpha)$	$\Gamma,\ \sigma : \alpha \ // \ \Gamma,\ \sigma : \alpha_1 \ ; \ \Gamma,\ \sigma : \alpha_2$
$(LL\beta)$	$\Gamma,\ \sigma : \beta \ // \ \Gamma,\ \sigma : \beta_1, \sigma : \beta_2$

[12] A similar solution was applied in Mints [191] by addition of Fitting's labels to whole sequents in his version of (labelled) hipersequent calculi.

The rule $[SUB]$ may be rewritten without any change, if we remember that each φ_i in the schema is meant as a labelled element of Γ and if $\varphi_i = \sigma : \psi$, then by $-\varphi_i$ we mean $\sigma : -\psi$.

Now, if we replace a rule (*Exp-A*) from Section 7.4 with (*LExp-A*) $\Gamma, \sigma : \varphi \ / \ \Gamma, \sigma : \psi$, or (*Res-A*) with (*LRes-A*) $\Gamma, \sigma : \varphi \ ; \ \Delta, \sigma : -\psi \ / \ \Gamma, \Delta$, we may keep the same φ and ψ as in the table from Section 7.4.2, but labels are nothing more than an adornment. Moreover, without $[CNEC]$ or $[CPOS]$ (redefined for labelled formulae) we lose completeness. But we can do better – due to labels we can get rid of strict subderivations and additional proof construction rules, except $[SUB]$. We need just one general rule for π-formulae:

$(LL\pi^i)$ $\Gamma, \sigma : \pi^i \ / \ \Gamma, \sigma.k : \pi$, where $\sigma.k$ is a new i-label in a derivation

Here, the introduction of a new label by π-formula clearly corresponds to the creation of a strict subderivation by $[CPOS]$ in MRND. The rule itself is an obvious clausal generalization of $(L\pi E)$ from LND. To get a formalization of **K** and their extensions we use (locally) labelled counterparts of expansion or resolution rules stated for MRND. Schemata of both rules (*LLExp-A*) and (*LLRes-A*) are the same as in nonlabelled case, namely:

$$\Gamma, \varphi \ / \ \Gamma, \psi \quad \text{or} \quad \Gamma, \varphi \ ; \ \Delta, -\psi \ / \ \Gamma, \Delta$$

but now, both φ and ψ are labelled formulae, and not only formulae themselves are different (defined suitably for each axiom A) but also the label of φ may differ from the label of ψ. For logics based on axioms like D, DC, T, TC and $4C$ we may keep φ and ψ as in the table from Section 7.4.2 with the same label σ added to both formulae. For the rest, including K, we display suitable candidates in the table below, but this time, in the first column, we have sometimes two axioms as a counterpart of a rule:

Axiom	φ	ψ	Side condition
K	$\sigma : \nu^i$	$\sigma.k : \nu$	$\sigma.k$ any i-label
4	$\sigma : \nu^i$	$\sigma.k : \nu^i$	$\sigma.k$ any i-label
5	$\sigma : \nu^i$	$\tau : \nu$	$\mid \sigma \mid >1$ and $\mid \tau \mid >1$
B	$\sigma.k : \nu^i$	$\sigma : \nu$	$\sigma.k$ any i-label
$B+4$	$\sigma.k : \nu^i$	$\sigma : \nu^i$	$\sigma.k$ any i-label
$B\text{-}Te$	$\sigma.k : \nu^i$	$\sigma : \nu$	$\sigma.k$ is j-label and $i \neq j \in \{F, P\}$
$B\text{-}Te+4$	$\sigma.k : \nu^i$	$\sigma : \nu^i$	$\sigma.k$ is j-label and $i \neq j \in \{F, P\}$

In the table we have listed the substitutes we must put in the places of φ and ψ in our (*LLExp-A*) or (*LLRes-A*) rules, remembering that in the latter we use $-\psi$. (*LLExp-K*) as an expansion- or (*LLRes-K*) as a resolution rule, is our basic ν-rule which, together with ($LL\pi^i$), yields a complete formalization of LLRND-**K**. In order to obtain extensions we must add either (*LLExp-A*) or (*LLRes-A*) for suitable axiom A. But this time we must remember that for every logic axiomatized by B and 4 we must add also a rule for $B + 4$, whereas for every temporal logic with 4 we must add also respective rule for $B\text{-}Te + 4$. In this way we cover the same logics which were formalized by MRND in Chapter 7.

Once again one may easily verify that (*LLExp-A*) and (*LLRes-A*) are mutually interderivable in LLRND; the proof goes exactly as in the nonlabelled case, hence we simply state the result as a:

Lemma 8.4 (Equivalence of expansion and resolution rules)

1. *(LLExp-A) is derivable in LLRND+(LLRes-A)*
2. *(LLRes-A) is derivable in LLRND+(LLExp-A)*

A definition of a derivation for a clause Σ in LLRND is a straightforward generalization of a definition provided in Chapter 4; we omit the details. Soundness and completeness of the resulting LLRND-systems follow from adequacy results stated in Sections 8.4 and 8.5. In particular, for soundness we apply an interpretation of labels as points in a model as in the Definition 8.4, and under this interpretation we assume that a clause is satisfied in a model if at least one element of a clause is satisfied in the value of its label. One can easily check in this way the (LAB)normality of all the rules, but note that respective (*LLExp*-5) is sound only in the monomodal case!

Completeness follows from the fact that instances of (*LLExp-A*) are clausal generalizations of the rules from Section 8.5, and that RND is a generalization of ND. Certainly, proofs of suitable axioms are very easy to obtain and the reader is encouraged to do it. But one may doubt if LLRND machinery is sufficient for simulation of (*RG*) since there are no rules for an introduction of modal constants. But it is extremely easy to obtain LLRND proof of ν^i if we have a proof of ν. The latter must have also some indirect proof $\mathcal{D}$ with $1 : -\nu$ as an assumption and $\perp$ as the last line. It is sufficient to rename all labels σ in $\mathcal{D}$ into $1.\sigma$, add $1 : -\nu^i$ at the beginning (as an assumption) and change the justification of $1 : -\nu$ (or rather $1.1 : -\nu$ after renaming of labels in $\mathcal{D}$), since it is now obtained by the application of ($LL\pi^i$) to $1 : -\nu^i$, and we are done. Thus obtained $\mathcal{D}'$ is a proof (by [SUB]) of ν^i. It works, in particular, for temporal logics with two (*RG*) rules.

Finally, note that in case of transitive symmetric logics (like **KB4** or **S5**) and transitive temporal logics we need for completeness also additional rules for $B+4$ and $B\text{-}Te+4$ respectively. Here is an example of a proof in LLRND-**Kt**, showing how these rules work.

1	SHØW: $1:\neg G(p\rightarrow Hq), 1: Fp\rightarrow q$	$[10, SUB]$
2	$1: G(p\rightarrow Hq)$	*ass.*
3	$1:\neg(Fp\rightarrow q)$	*ass.*
4	$1: Fp$	$(3, LL\alpha)$
5	$1:\neg q$	$(3, LL\alpha)$
6	$1.1[F]: p$	$(4, LL\pi^i)$
7	$1.1[F]: p\rightarrow Hq$	$(2, LLExp\text{-}K)$
8	$1.1[F]:\neg p,\ 1.1[F]: Hq$	$(7, LL\beta)$
9	$1.1[F]:\neg p$	$(5, 8, LLRes\text{-}B\text{-}Te)$
10	$\bot$	$(6, 9, LLRes)$

8.7.2 Global Labelling

In this subsection we make only a small change; we consider clauses built of ordinary (nonlabelled) formulae with labels attached to the whole clause instead. Hence labels have a global character and the system of this sort will be called GLRND. This small change however forces us to make a lot of further changes in the description of rules. Also, it seems that in consequence there are serious differences in the strategy of making derivations in LLRND and GLRND.

In GLRND $[SUB]$ is also the only proof construction rule but now we assume that in the schema, some fixed label σ is preceding Γ, Δ and each assumption $-\varphi_i$; X consists of labelled clauses as well, but their labels may be different.

The inference rules for GLRND-**K** are the following:

(GLW)	$\sigma:\Gamma\ /\ \sigma:\Gamma,\varphi$
$(GLRes)$	$\sigma:\Gamma,\varphi\ ;\ \sigma:\Gamma,-\varphi\ /\ \sigma:\Gamma$
$(GL\alpha)$	$\sigma:\Gamma,\alpha\ //\ \sigma:\Gamma,\alpha_1\ ;\ \sigma:\Gamma,\alpha_2$
$(GL\beta)$	$\sigma:\Gamma,\beta\ //\ \sigma:\Gamma,\beta_1,\beta_2$
$(GL\neg\neg)$	$\sigma:\Gamma,\neg\neg\varphi\ //\ \sigma:\Gamma,\varphi$
$(GL\pi^i)$	$\sigma:\pi_1^i,...,\pi_n^i\ /\ \sigma.k:\pi_1,...,\pi_n$, where $\sigma.k$ is a new i-label in a derivation
$(GLExp\text{-}K)$	$\sigma:\nu_1^i,...,\nu_n^i\ /\ \sigma.k:\nu_1,...,\nu_n$, where $\sigma.k$ is any i-label in a derivation

For logics axiomatized by schemata $D, DC, T, TC, 4C$ we can provide the pair of rules:

($GLExp$-A) $\sigma : \Gamma, \varphi \;/\; \sigma : \Gamma, \psi$ or
($GLRes$-A) $\sigma : \Gamma, \varphi \;;\; \sigma : \Delta, -\psi \;/\; \sigma : \Gamma, \Delta$

with φ and ψ defined exactly as in the table from Section 7.4.2. Also the proof of interderivability of these rules is exactly the same. For other logics the situation is a bit more complicated, so at first we simply display suitable GL-expansion rules with the variant rule for D which is a generalization of Fitting's rule (LD') discussed in Remark 8.4:

($GLExp$-D)	$\sigma : \nu_1^i, ..., \nu_n^i \;/\; \sigma.k : \nu_1, ..., \nu_n$, where $\sigma.k$ is any i-label in a derivation, possibly new
($GLExp$-4)	$\sigma : \nu_1^i, ..., \nu_n^i \;/\; \sigma.k : \nu_1^i, ..., \nu_n^i$, where $\sigma.k$ is any i-label in a derivation
($GLExp$-5)	$\sigma : \nu_1^i, ..., \nu_n^i \;/\; \tau : \nu_1, ..., \nu_n$, where $\mid \sigma \mid >1$ and $\mid \tau \mid >1$
($GLExp$-B)	$\sigma.k : \nu_1^i, ..., \nu_n^i \;/\; \sigma : \nu_1, ..., \nu_n$, where $\sigma.k$ is i-label in a derivation
($GLExp$-$B4$)	$\sigma.k : \nu_1^i, ..., \nu_n^i \;/\; \sigma : \nu_1^i, ..., \nu_n^i$, where $\sigma.k$ is i-label in a derivation
($GLExp$-B-Te)	$\sigma.k : \nu_1^i, ..., \nu_n^i \;/\; \sigma : \nu_1, ..., \nu_n$, where $\sigma.k$ is j-label in a derivation and $i \neq j \in \{F, P\}$
($GLExp$-$B4$-Te)	$\sigma.k : \nu_1^i, ..., \nu_n^i \;/\; \sigma : \nu_1^i, ..., \nu_n^i$, where $\sigma.k$ is j-label in a derivation and $i \neq j \in \{F, P\}$

We can prove (LAB)normality of all the rules displayed above by means of the same technique of label interpretation as that applied in Section 8.4, provided we treat clauses as disjunctions. This is enough for demonstrating soundness of GLRND. Completeness follows by analogical argument as that used for LLRND in the preceding subsection.

The system works in a similar vein as a destructive resolution system of Fitting [94] but labels make GLRND more extensive. Also the format of RND offers more freedom in proof construction. In particular, since [SEP] is an admissible rule of RND (see Chapter 4) and also of GLRND, we do not need any special rule for breaking down clauses in order to separate a modal part needed for the application of modal rules.

In order to get a resolution-like characterization we need a pair of rules for each ($GLExp$-A) considered above:

$(GLRes'\text{-}A)$ $\sigma : \Gamma, \varphi \; ; \; \tau : -\psi \; / \; \sigma : \Gamma$ and
$(GLRes''\text{-}A)$ $\sigma : \varphi \; ; \; \tau : \Gamma, -\psi \; / \; \tau : \Gamma$

where labels σ and τ are not necessarily different. Label of a clause containing φ and $-\psi$, as well as these formulae, are defined exactly as in the table from Section 8.7.1.

In case of GLRND we can also prove the interderivability of (*GLExp-A*) with a pair of (*GLRes-A*) rules but this time a proof is slightly more involved.

Lemma 8.5 (Equivalence of expansion and resolution rules)

1. *(GLExp-A) is derivable in GLRND+(GLRes′-A)+(GLRes″-A)*
2. *(GLRes′-A) and (GLRes″-A) are derivable in GLRND+(GLExp-A)*

Proof We will demonstrate the second case. The following schema shows derivability of $(GLRes'\text{-}A)$ by (*GLExp-A*) with $\sigma : \Gamma = \{\sigma : \chi_1, \ldots, \chi_k\}$ (if $k = 0$, a schema considerably simplifies). The application of a particular case of the latter rule is correct for every instance of a parameter A, which may be checked by comparison of each case of φ and ψ in the table from Section 8.7.1 for any A. Derivability of $(GLRes''\text{-}A)$ is similar. ■

1	$\sigma : \Gamma, \varphi$	premise
2	$\tau : -\psi$	premise
3	SHØW: $\sigma : \Gamma$	$[k+k+5, SUB]$
4	$\sigma : -\chi_1$	*ass.*
	$\vdots$	
$k+3$	$\sigma : -\chi_k$	*ass.*
	$\vdots$	
$k+k+3$	$\sigma : \varphi$	$(1, 4-k+3, GLRes)$
$k+k+4$	$\tau : \psi$	$(k+k+3, GLExp\text{-}A)$
$k+k+5$	$\bot$	$(2, k+k+4, GLRes)$

For the sake of illustration we display an example of a proof in GLRDN-**K** where, for simplicity, we use [*SEP*] which, as we remarked above, is an admissible proof construction rule. Let φ be a shorthand for a thesis $(\Box p \to \Box q) \vee (\Box r \to \Box s) \to \neg(\Box(p \wedge r) \wedge \Diamond\neg(q \vee s))$

1	SHØW: $1 : \varphi$	$[23, SUB]$
2	$1 : \neg\varphi$	*ass.*
3	$1 : (\Box p \rightarrow \Box q) \vee (\Box r \rightarrow \Box s)$	$(2, GL\alpha)$
4	$1 : \Box(p \wedge r) \wedge \Diamond\neg(q \vee s)$	$(2, GL\alpha)$
5	$1 : \Box(p \wedge r)$	$(4, GL\alpha)$
6	$1 : \Diamond\neg(q \vee s)$	$(4, GL\alpha)$
7	$1 : \Box p \rightarrow \Box q, \Box r \rightarrow \Box s$	$(3, GL\beta)$
8	$1 : \neg\Box p, \Box q, \Box r \rightarrow \Box s$	$(7, GL\beta)$
9	$1 : \neg\Box p, \Box q, \neg\Box r, \Box s$	$(8, GL\beta)$
10	SHØW: $1 : \neg\Box p, \neg\Box r$	$[17, SEP]$
11	$1 : \Box q, \Box s$	*ass.*
12	$1.1 : \neg(q \vee s)$	$(6, GL\pi^i)$
13	$1.1 : q, s$	$(11, GLExp\text{-}K)$
14	$1.1 : \neg q$	$(12, GL\alpha)$
15	$1.1 : \neg s$	$(12, GL\alpha)$
16	$1.1 : s$	$(13, 14, GLRes)$
17	$\bot$	$(15, 16, GLRes)$
18	$1.2 : \neg p, \neg r$	$(10, GL\pi^i)$
19	$1.2 : p \wedge r$	$(5, GLExp\text{-}K)$
20	$1.2 : p$	$(19, GL\alpha)$
21	$1.2 : r$	$(19, GL\alpha)$
22	$1.2 : \neg r$	$(18, 20, GLRes)$
23	$\bot$	$(21, 22, GLRes)$

Thanks to the introduction of $[SEP]$ we could immediately break up the clause from line 9, which contains both π- and ν-formulae, on two clauses (in lines 10 and 11) containing only modal formulae of one sort. This enabled an application of $(GLExp\text{-}K)$ (line 13), and then $(GL\pi^i)$ (line 18). The reader may try to prove this thesis only by means of $[SUB]$.

Both variants of LRND are not analytic but may be transformed to analytic form, based only on elimination rules, without losing completeness (in particular, both versions of weakening $((LLW)$ and $(GLW))$ are eliminable). In Chapter 10 we will show completeness of an analytic version of LND. It is obvious that both proposed versions of LRND may simulate LND, hence they may be restricted in a similar way. Details of suitable refinements are left to the reader.

Chapter 9

Logics of Linear Frames

Logics of linear frames, called here for short linear logics,[1] form a particularly interesting and important class, especially in temporal interpretation. But we devote a separate Chapter for their treatment not because of their importance but rather because of special problems generated by their formalization in the setting of labelled systems. We will be dealing with basic linear logics in monomodal version (**K4.3, KD4.3, S4.3**) and in bimodal temporal version (**Kt4.3**, **Kt4D.3** e.t.c.).

Section 9.1 is a survey of several formalizations of linear logics introduced so far. We are particularly interested in the comparison of different strategies of proof construction and linear model search embodied in the rules of several systems. It is remarkable that most of the proposed solutions is based on branching rules. Moreover, in some approaches the number of branches generated by suitable rules is not fixed but exponentially dependent on the number of π-formulae. Because of this feature most of the solutions cannot be simulated in ND, particularly in Jaśkowski's format.

The rest of the Chapter is devoted to the presentation of LND system for linear logics. In contrast to many other systems it is based on nonbranching rules. First of all, such a solution is more general because it may be applied not only in ND but also in TS or SC, provided some form of cut is admitted. Moreover, we will show that the application of nonbranching rules is more convenient from the standpoint of complexity. In many cases it may lead to construction of proofs exponentially shorter than corresponding proofs created by branching rules. In Section 9.2, for the sake of illustration of the

[1] It shouldn't cause any misunderstanding, since in this book we do not deal with Girard's linear logic.

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*, Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_9,

basic strategy, we introduce the simplest form of such a system for **S4.3**. It is extended to more complex version for **Kt4.3** in the next section.

Section 9.4 contains completeness proof for analytic version of our system. It must be proved directly since there are no analytic TS's which may be simulated by analytic LND, as was the case of logics considered in the preceding Chapter. The proof is based on relative maximalization of each label with respect to some predefined finite set of formulae. Although it is based on constructive proof-search procedure and saves subformula property, it is not very practical; in the next Chapter we introduce more efficient procedures. Finally, we describe some extensions and modifications of this system in Section 9.5. In particular, we describe two versions of RND for linear logics, with local and global labels. We also briefly describe how to extend our approach to obtain a similar formalization of other logics, like **S4F** and **S4R**. It is due to the fact that these logics are determined by frames defined by syntactically similar conditions (i.e. universal implications).

9.1 Deductive Systems for Logics of Linear Frames

9.1.1 Survey of Systems

Before we present our LND system for logics of linear frames we describe, and make a comparison of other proposals. They are extensions of systems introduced in earlier chapters, so we only discuss the rules which serve to formalization of linearity.

The existing systems for linear modal and temporal logics belong essentially to two kinds of proof systems: tableau systems and sequent systems of different sorts. The only one resolution system dealing with linearity, known to me, is due to Fischer, Dixon and Peim [88], but this formalization is based on the "Next" operator. In what follows we do not consider systems that formalize logics of linear frames but in different languages based on such operators like "Until", "Since" or "Next", or characterized by different semantics. It means that, for example, TS's of Wolper [284], Lichtenstein and Pnuelli [177], Goubault-Larrecq and Schmitt [118], resolution system of Fischer, Dixon and Peim [88], and strongly labelled ND system of Bolotov, Basukoski, Grigoriev and Shangin [47], despite their unquestioned value, will not be discussed.

Below we present a chronological list of solutions for linear logics in standard languages:

1. 1971 – Rescher and Urquhart's TS in [231]
2. 1973 – Zeman's standard SC in [288]
3. 1991 – Shimura's standard SC in [252]
4. 1991 – Catach's automated theorem prover TABLEAUX from [64]
5. 1992 – Wansing's display calculus; the first version, finally in [280]
6. 1992 – Goré's TS in [116]
7. 1994 – Indrzejczak's standard ND in [140]
8. 1994 – Kashima's nested TS in [161]
9. 2000 – Indrzejczak's multisequent calculus in [146]
10. 2000 – Marx', Mikulas' and Reynolds' nonstandard TS in [183]
11. 2002 – Indrzejczak's labelled KE in [149]
12. 2002 – Castellini's and Smaill's strongly labelled SC in [63]
13. 2003 – Baldoni's strongly labelled TS in [18]
14. 2005 – Negri's strongly labelled SC in [194]

As we will see, their simulation in ND is problematic in many cases, mainly because of branching character of proposed rules, but it will be instructive to recall them all before we present our solution. Some systems for linear logics apply rules, where the number of branches is not constant but depends on some factor. In fact, these are chronologically the first formalizations of such logics and we describe them first. Systems that were invented later usually apply rules with fixed number of branches. Usually it is possible thanks to more complex character of their deductive machinery.

In order to facilitate a comparison of different approaches we will use Kashima's format of tableau for uniform presentation of rules, so the reader should recall this formalization first (cf. Section 7.5.3). In some cases one may also consult the original formulation given in previous chapters.

Multi-Branching Rules

In standard SC and TS's for modal logics the same approach was applied in [288, 252, 116]. We have already described suitable rules in Section 7.1.4. and discussed some disadvantages of this solution. Here we rather focus on the strategy of (linear) model searching involved in these rules. Basically, the systems of Zeman, of Shimura and of Goré, realize the strategy of generating at the same time all linear models admissible at the current stage. Thus, in case of **S4.3**, we have a rule which in Kashima's format looks like this[2]:

$$(\neg\Box E^3_K) \quad \frac{X[\neg\Box\Delta]}{X[\{\neg\Box\Delta_1, \neg\varphi_1\}] \mid \ldots \mid X[\{\neg\Box\Delta_n, \neg\varphi_n\}]}$$

where: $\Delta = \{\varphi_1, \ldots, \varphi_n\}, \Delta_i = \Delta - \{\varphi_i\}$

It means that if we have at some stage of model construction n unused π-formulae at some state w (= the set of formulae in a premise), we must generate n branches, one for each π-formula. We also rewrite in each branch the remaining $n-1$ π-formulae as unused. So, every π-formula introduces an immediate $\mathcal{R}$-successor of w, where the activation of remaining π-formulae is postponed. This way, every branch which does not close, leads to construction of linear model. This approach was used only to formalization of monomodal linear logics and it is not evident how it may be extended to temporal logics. This usually requires some nonstandard system as a basis; the only exception is Indrzejczak's ND from [140] for temporal logics, presented in Section 7.1.5.

The remaining approaches to linear logics are connected with some nonstandard systems. The earliest one is due to Rescher and Urquhart [231] and applied in the system of boxed tableaux, briefly described in Section 7.5.2. They also provide a specific π-rule for linearity, but in contrast to Zeman/Shimura/Goré approach, their rule applies to only one π-formula at a time. Anyway, it is also a branching rule, and the number of branches is not constant but depends on the level of embedding of the current box. Roughly speaking, it corresponds to the number of different points in attempted linear model which are "after" (in case of π^F) or "before" (in case of π^P) the current state; so, in case there are none, it is an ordinary, non-branching $(\pi^i E)$. We will not specify the general rule but, for the sake of illustration, display in Kashima's format a rule of π^F-elimination for weak connectedness in a special situation. Let the model described in the premise contain two

[2]Note that we do not apply superscripts with "{" in monomodal case.

points w_1 and w_2, $\mathcal{R}$-accessible from the point w, where the π^F-formula we are dealing with is satisfied. The rule looks like this:

$$(\pi^F E^3_K) \quad \frac{X[F\varphi \ \{^F \ \Gamma \ \{^F \ \Delta \ \}\}]}{S_1 \mid S_2 \mid S_3 \mid S_4 \mid S_5}$$

where: $S_1 = X[\{^F \ \varphi \ \{^F \ \Gamma \ \{^F \ \Delta \ \}\}\}]$;
$S_2 = X[\{^F \ \varphi, \Gamma \ \{^F \ \Delta \ \}\}]$;
$S_3 = X[\{^F \ \Gamma \ \{^F \ \varphi \ \{^F \ \Delta \ \}\}\}]$;
$S_4 = X[\{^F \ \Gamma \ \{^F \ \varphi, \Delta \ \}\}]$;
$S_5 = X[\{^F \ \Gamma \ \{^F \ \Delta \ \{^F \ \varphi \ \}\}\}]$.

Thus, S_1 corresponds to the situation, where a new state w_0 was introduced such that $\mathcal{R}ww_0$ and $\mathcal{R}w_0w_1$, $S_2 - w_0 = w_1$, $S_3 - \mathcal{R}w_1w_0$ and $\mathcal{R}w_0w_2$, e.t.c.

Rescher and Urquhart do not present rules for monomodal linear logics and for strong connectedness in bimodal context, but their approach may be easily modified. For example, a suitable rule for **S4.3** with "three-points-after" model represented in the premise, looks like this:

$$(\Diamond E^3_K) \quad \frac{X[\Diamond\varphi \ \{\Gamma\{\Delta\{\Sigma\}\}\}]}{S_1 \mid S_2 \mid S_3 \mid S_4}$$

where: $S_1 = X[\{ \ \varphi \ \{ \ \Gamma \ \{ \ \Delta \ \{ \ \Sigma \ \}\}\}\}]$;
$S_2 = X[\{ \ \Gamma \ \{ \ \varphi \ \{ \ \Delta \ \{ \ \Sigma \ \}\}\}\}]$;
$S_3 = X[\{ \ \Gamma \ \{ \ \Delta \ \{ \ \varphi \ \{\Sigma \ \}\}\}\}]$;
$S_4 = X[\{ \ \Gamma \ \{ \ \Delta \ \{ \ \Sigma \ \{ \ \varphi \ \}\}\}\}]$

It is worth saying that in Rescher/Urquhart's system one may separate F-linearity from P-linearity. For example, if we take an ordinary elimination rule for π^F, and the linear variant for π^P, we will obtain a TS adequate for the logic of tree-like models (linear past, but several possible future-paths).

Rules with Fixed Number of Branches

Strongly labelled TS system of Catach [64], implemented as TABLEAUX, contains a rule for weak connectedness (which may be easily changed to cover connectedness). It is also a branching π-rule but, in contrast to both approaches already described, it generates a constant number of branches. In Hintikka style TS but with strong labels and relational formulae included (cf. Section 8.2.2) it may be stated:

$$(\Diamond E^L)\quad \frac{\Gamma, xRy, x : \Diamond\varphi}{\Gamma, xRy, xRz, zRy, z : \varphi \mid \Gamma, xRy, y : \varphi \mid \Gamma, xRy, y : \Diamond\varphi}$$

where Γ contains both labelled and relational formulae.

In Kashima's style one may display it as follows:

$$(\Diamond E^L_K)\quad \frac{X[\Diamond\varphi\ \{\Gamma\ldots\}]}{X[\{\ \varphi\ \{\ \Gamma\ldots\}] \mid X[\{\ \Gamma,\ \varphi\ \ldots\}] \mid X[\{\ \Gamma,\ \Diamond\varphi\ \ldots\}]}$$

where: "..." means that some other sets of formulae in braces may appear to the right of Γ.

One may note a strong similarity with the solution applied in Rescher's system. Again, we consider only one π-formula at a time, and in case at least one state w_1 in attempted model is $\mathcal{R}$-accessible to a state w where $\Diamond\varphi$ holds, we apply the rule. If there are no such points, we apply a standard non-branching $(\pi^i E)$. The difference between Catach's and Rescher's approach is that in the former we explicitly display only cases $\mathcal{R}w_0w_1$ (leftmost branch) and $w_0 = w_1$ (middle branch). In the rightmost branch $\Diamond\varphi$ is simply propagated to w_1 for later use, whereas in Rescher's approach all possible cases are generated immediately.

In Catach's approach the fixed number of branches is due to the fact that at each stage only two different worlds w_0 and w_1 need to be confronted, exactly as in the condition for weak connectedness. Even more direct realization of this condition is involved in several nonstandard SC's of Wansing, Indrzejczak and Negri. All of them are nonstandard but differ in many ways, yet suitable rules are based on the same idea.

Wansing [280] obtains the formalization of linear temporal logics on the ground of display calculus with the help of two structural rules, for linear future and past respectively:

$$(DLF)\quad \frac{X \Rightarrow Y \mid \bullet X \Rightarrow Y \mid \star \bullet \star X \Rightarrow Y}{\bullet \star \bullet \star X \Rightarrow Y}$$

$$(DLP)\quad \frac{X \Rightarrow Y \mid \bullet X \Rightarrow Y \mid \star \bullet \star X \Rightarrow Y}{\star \bullet \star \bullet X \Rightarrow Y}$$

An addition of these rules to display calculus for **Kt4** yields a formalization of **Kt4.3**. One may use only one of them obtaining a system with only past- or future-linearity. It is also easy to get an effect of connectedness by deleting the leftmost premise. We do not attempt to explain the meaning of these rules since it presupposes some presentation of display calculus in general – for this one should consult e.g. [280]. Anyway, Wansing's rules may be simulated in Kashima's system; the counterpart of (DLF) looks like that:

$$(DLF_K) \quad \frac{X[\{^F\Gamma\}\{^F\Delta\}]}{X[\{^F\Gamma, \Delta\}] \mid X[\{^F\Gamma\{^F\Delta\}\}] \mid X[\{^F\Delta\{^F\Gamma\}\}]}$$

In multiple sequent calculus (cf. Section 8.2.1) linearity is obtained in a similar way via structural rule:

$$(MLF) \quad \frac{\Gamma \ <0,0> \Rightarrow \Delta \mid \Gamma \ <0,1> \Rightarrow \Delta \mid \Gamma \ <1,0> \Rightarrow \Delta}{\Gamma \ <1,1> \Rightarrow \Delta}$$

If we recall the interpretation of indexed sequents given in Section 8.2.1 it would be evident that indices of three premises correspond to the three disjuncts in the succedent of the condition for weak connectedness. So despite totally different apparatus it represents exactly the same approach which is involved in Wansing's system. But there is one difference; the machinery of weakly labelled multisequents is not as expressive as that of display logic, in consequence only future-linearity is representable in MSC.

Recently, Negri [194] has proposed a strongly labelled SC for monomodal logics, where frame conditions falling under the schema of universal or geometric implication (cf. Section 1.1.5 for definitions) are covered by the rules of uniform character. Since (weak) connectedness is an example of a universal implication it is also dealt with in this frame. Roughly speaking, the general schema of SC rule is:

$$(R\text{-}UI) \quad \frac{\psi_1, \varphi_1, ..., \varphi_k, \Gamma \Rightarrow \Delta \mid, ..., \mid \psi_n, \varphi_1, ..., \varphi_k, \Gamma \Rightarrow \Delta}{\varphi_1, ..., \varphi_k, , \Gamma \Rightarrow \Delta}$$

for each universal implication of the form $\forall x_1 ... x_i(\varphi_1 \wedge ... \wedge \varphi_k \rightarrow \psi_1 \vee ... \vee \psi_n)$. The particular instance for weak connectedness is the following:

$$\frac{y = z, xRy, xRz, \Gamma \Rightarrow \Delta \mid yRz, xRy, xRz, \Gamma \Rightarrow \Delta \mid zRy, xRy, xRz, \Gamma \Rightarrow \Delta}{xRy, xRz, \Gamma \Rightarrow \Delta}$$

Once again, it is obvious that premises correspond to disjuncts of the succedent, whereas in the conclusion, the antecedent of weak connectedness condition is stated.

Also Castellini and Smaill [63] provided a general method for introducing suitable rules in their strongly labelled SC. Applying their procedure we obtain the following rule for weak connectedness:

$$(CSL) \frac{\Gamma \Rightarrow \Delta, xRy \mid \Gamma \Rightarrow \Delta, xRz \mid y = z, \Gamma \Rightarrow \Delta \mid yRz, \Gamma \Rightarrow \Delta \mid zRy, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

Note that this rule, although of different shape, is also modeled directly on suitable frame condition which may be equivalently stated:

$$\forall xyz(\neg \mathcal{R}xy \vee \neg \mathcal{R}xz \vee \mathcal{R}yz \vee \mathcal{R}zy \vee y = z) \tag{9.1}$$

Such a formula corresponds directly to a branching rule with 5 premises.

Although we have constantly used Kashima's format for a uniform representation of different approaches to formalization of linearity, we haven't yet discussed his own solution from [161]. His rules for linearity have also a constant number of branches. In order to obtain a formalization for **Kt4.3** Kashima introduces the following rule:

$$(TrCo)\quad \frac{X[G\Gamma, H\Delta, \Theta]\ G\Lambda, H\Pi, \Sigma}{X[\Pi, H\Pi]\ \Gamma, G\Gamma \ \mid\ X[\Lambda, G\Lambda]\ \Delta, H\Delta \ \mid\ X[\Sigma]\ \Theta}$$

Clearly, it works for a variant with (*turn*) (cf. a Section 7.5.3), since $G\Lambda, H\Pi, \Sigma$ in the premise are not in braces. Because (*turn*) is correct only in temporal logics (or monomodal symmetric) this solution is applicable only for temporal linear logics. But the situation is not the same as in ND system from Section 7.1.5; we may in this formalization define a suitable rule also for monomodal linear logic. For this we must admit rules, where analogical transfer of sets of formulae is allowed on arbitrary level of nesting in a K-sequent. Such a rule for **S4.3** may have the following form:

$$(S4.3_K)\quad \frac{X[\{\Gamma, \Box\Delta\}, ..., \{\Theta, \Box\Lambda\}]}{X[\{\Gamma, \Box\Delta, \Box\Lambda\}, ..., \{\Theta, \Box\Lambda\}] \mid X[\{\Gamma, \Box\Delta\}, ..., \{\Theta, \Box\Lambda, \Box\Delta\}]}$$

Nonbranching Rules

We know of four formalizations of linear logics, where suitable rules are not branching: two ND systems of Indrzejczak, standard [140] and labelled one [151], and two TS's: one due to Marx, Mikulas and Reynolds [183], and the second due to Baldoni [18]. Except labelled ND, remaining ones were already introduced, so we only briefly recall the basics. Labelled ND will be presented in a greater detail in the rest of the Chapter.

A standard ND system for temporal logics, where the effect of temporal axioms LF and LP is simulated by reiteration rule, was already presented in Section 7.1.5. Here we only remark that it is rather not well suited for proof search in an analytic way, although actual proof construction is quite easy in many cases.

Particularly simple schemata of rules covering LF and LP are obtained in Baldoni's strongly labelled TS by his general recipe for generating rules from $G^{a,b,c,d}$ axioms. Since both temporal axioms are instances of axioms for grammar logics of the form $[b]\varphi \rightarrow [c]\varphi$, suitable rules have particularly simple form:

$$(BLF)\ \ xR_{F;P}y\ /\ xR_{P\cup\varepsilon\cup F}y \quad \text{and} \quad (BLP)\ \ xR_{P;F}y\ /\ xR_{P\cup\varepsilon\cup F}y$$

Clearly, this is possible due to the presence of complex modalities and later leads to branching when we apply the rule for breaking unions of relations. For instance, when applying a suitable rule to $xR_{P\cup\varepsilon\cup F}y$ we get three branches for xR_Py, $xR_\varepsilon y$ and xR_Fy.

An interesting solution is examined in weakly labelled TS of Marx, Mikulas and Reynolds, where tableaux are triples of sets of formulae. We have already presented this system together with the characteristic π-rules for linearity. So for the time being we recall this solution in the Kashima's format:

$$(3FE_K)\ \ \frac{X[\ \Gamma, G\Delta, F\varphi\ \{^F \Lambda, H\Sigma, \neg F\varphi, \neg\varphi\}]}{X[\ \Gamma, G\Delta, F\varphi\ \{^F\ G\Delta, \Delta, \varphi, H\Sigma, \Sigma\ \{^F\ \Lambda, H\Sigma, \neg F\varphi, \neg\varphi\}\}]}$$

$$(3PE_K)\ \ \frac{X[\{^P\ \Gamma, G\Delta, \neg P\varphi, \neg\varphi\ \}\ \Lambda, H\Sigma, P\varphi]}{X[\{^P\{^P\ \Gamma, G\Delta, \neg P\varphi, \neg\varphi\ \}\ G\Delta, \Delta, \varphi, H\Sigma, \Sigma\ \}\ \Lambda, H\Sigma, P\varphi]}$$

We will discuss the intended meaning of these rules in the next subsection.

9.1.2 A Comparison of System's Properties and Strategies of Linearization

Before we go on with our preferred solution of linearity representation we summarize some features and compare strategies involved in other approaches. It helps to explain the rationale behind the rules presented in the next sections and to show possible advantages of our approach.

Modularity

In many systems, including standard ones, linearity is dealt with via π-rules. It is rather not in harmony with the usual methodology of tableau systems where, there is only one basic rule for π-formulae and all strengthenings are obtained through addition (or modification) of rules for ν-formulae. The consequence of getting linearity by π-rules is that in some systems we have no modularity. For example, in standard Zeman/Shimura/Goré approach we have not a system for, say, **S4** with some special rule(s) for linearity but a system where a complex rule provides transitivity, connectedness (and sometimes even more) in one shoot. Failed proof-search in such a system immediately generates linear models which may be unpleasant in automated deduction. For example, we cannot use a procedure defined for, e.g. **S4**, and add some new elements, but we must define a specific procedure from scratch. Also, if we find a proof of a thesis by a procedure defined for such a system we may be not able to check if it is a thesis of **S4.3** or of some sublogic. The same remark applies to Rescher and Urquhart's system.

In the systems of Catach, or of Marx, Mikulas, Reynolds, a partial separation of frame properties is possible but at the cost of having two π-elimination rules in one system. In both cases during a proof search we attempt to build linear models from the beginning.

Only systems of more complex character, like e.g. display calculus of Wansing [281], strongly labelled system of Negri [194], or multiple-sequent calculus of Indrzejczak [146], have special structural rules for linearity. So they provide not only modularity of the system but also satisfy Došen's condition that every extension of the basic system should be obtained by rules not exhibiting logical constants.

In Kashima's calculus linearity is dealt with by complex rules of rather mixed character. They are not structural and only partially modular; F-linearity cannot be separated from P-linearity for original rules. But we have shown that this limitation may be overcome by providing a variant rule for monomodal linearity.

Branching

It is striking that most of the rules devised to deal with linearity introduce some kind of branching into the proof search. Moreover, often the number of possible branches is not fixed but depends on some factor. This is the case of standard tableau systems of Zeman, Goré and the sequent system of Shimura, as well as semantically based TS of Rescher and Urquhart. There are systems having rules with a fixed number of branches but this is possible due to the rather complex character of the whole system. This group includes: nested sequent system of Kashima, display calculus of Wansing and some labelled systems like: strongly labelled TS of Catach and Baldoni, strongly labelled sequent systems of Castellini and Smaill, and that of Negri, and weakly labelled multiple sequent calculus of Indrzejczak.

Basically, there are three strategies involved in the application of branching rules. In systems, where the number of branches generated by suitable rule is not fixed, it generally depends on the number of unused π-formulae currently present on extended branch. But this dependence is realized in different ways; we must distinguish between the strategy of Zeman, Shimura and Goré, and the strategy of Rescher and Urquhart. The former is a strategy of systematic decreasing of the number of branches, whereas the latter is a strategy of systematic increasing of their number. However, in both approaches we realize the strategy of generating at the same time all linear models admissible at the current stage. So, from this point of view we may describe them as global strategies.

In the strategy of Zeman, Shimura and Goré, in case of **S4.3**, if we have at some stage n unused π-formulae we generate n new branches, one for each π-formula. We also rewrite in each branch the remaining $n-1$ π-formulae as unused, hence, potentially, any branch produces $n-1$ new branches and so on, until we get $n!$ branches, where each one represents a possible linear sequence of n worlds generated by these n π-formulae. Obviously, the number of branches may increase considerably, as any of these original π-formulae may, by saturation, add k new π-formulae. The case of Goré's and Shimura's rule for weak connectedness ia even more complex, since from n π-formulae in the premise it produces 2^n-1 branches. Generally, in this approach we generate immediately all linear models admissible at some stage, by using all π-formulae at once.

The tableau system of Rescher and Urquhart follows slightly different variant of this strategy: we apply a π-rule for only one formula at a time but we immediately put a new state in every possible place in a sequence of already existing states. For example, let the branch represent a model with n different states in a sequence following a state t where some unused π^F holds. Then, if we deal with this π^F, we must create $n+1$ new branches for each open branch containing t with π^F. Every new branch corresponds to a new possible chain in the following way. Before the application of the rule we had a chain $\langle ..., t, t_1, ..., t_n\rangle$, and after we have $n+1$ new sequences: $\langle ..., t, t_0, t_1, ..., t_n\rangle$, $\langle ..., t, t_1, t_0, ..., t_n\rangle$, ..., $\langle ..., t, t_1, ..., t_n, t_0\rangle$, where t_0 is a new state generated by the activation of π^F-formula. So, in this strategy the number of new branches does not depend on the number of unused π-formulae but on the number of those that were already used to create new states in attempted models of open branches. Anyway, both Zeman/Goré/Shimura and Rescher/Urquhart's strategy eventually gives the same growth of branches, corresponding to all permutations of states; the difference is that in Rescher, Urquhart's strategy we produce $n!$ in the reverse order.

In fact, the original Rescher, Urquhart's rule is even more complicated, because it is devised for temporal, irreflexive systems, hence the number of new branches is $2n + 1$[3] forced by weak connectedness but in case of **S4.3**, due to strong connectedness, we can simplify the matters. Again, the overall computational effect of either Rescher and Urquhart, or of Goré's strategy is the same.

So, both strategies described above are global in the sense of engagement of all π-formulae (or states in a model) during the application of discussed rules. The remaining solutions are based on the local strategy: a fixed number of branches generated by suitable rules is due to the fact that at each stage only two different states in a model need to be confronted, exactly as in the condition for (weak) connectedness. In details there are also some differences. Catach's rule, as we already remarked, follows quite closely the strategy of Rescher and Urquhart, only the work is partitioned. While Rescher's rule deals with all states in a sequence $\langle ..., t, t_1, ..., t_n \rangle$ at once (see above), Catach's rule produces only two alternative models with new state t_0 (between t and t_1, and with $t_0 = t_1$), whereas in the third branch we only push π^i from t to t_1. Anyway, Catach's rule while working locally still obtains a global effect of building only linear models on all open branches, as we noted above.

Only the rules of strongly labelled SC's, as well as those of Wansing, Kashima or Indrzejczak are local indeed, because they enable a full separation of the stage of building a model (an introduction of new states by π-rules) and the linearization stage (by suitable rules).

Although the rules with fixed number of generated branches may look better, it does not seem that in these systems we generally obtain proof trees with smaller number of branches (we think here of the number of branches implied by the number of π-formulae creating new states). In the worst case we have the same result; $n!$ branches from n π-formulae in case of strong connectedness. But in practice they seem to work better; one may often construct much simpler proofs for many theses than in Goré or Rescher/Urquhart system.

Finally, let us take a look at nonbranching tableau system of Marx, Mikulas and Reynolds. In fact, this system follows almost the same model-seeking strategy as that of Rescher and Urquhart. It also deals with only

[3]An additional effect of identifying a new state with every one already present.

one π-formula at a time but instead of generating all possible sequences of states, it rather eliminates those that are impossible, putting the new state in a suitable place (no immediate contradiction). So this rule is based on the following statement of weak connectedness:

$$\forall xyz(\mathcal{R}xy \land \mathcal{R}xz \land \neg\mathcal{R}yz \land y \neq z \to \mathcal{R}zy) \tag{9.2}$$

But it works due to the prior use of an analytic cut in already existing states. The procedure of proof search defined for completeness proof leads not only to downward saturation but also to relative maximalization of sets of formulae corresponding to states of a model. This way, a nonbranching rule for linearity presupposes an earlier introduction of many branches by cut, and in the worst case does not offer a better performance.

Needless to say that from the computational standpoint nonbranching rules are more welcome. We will illustrate this point in the next section. Finally, it should be noted that also transfer of these rules to ND system may encounter some difficulties. Some of the branching rules may be simulated in Gentzen's format ND with strong labels. For example, Negri's rule may be added to Basin/Matthews/Vigano's system in the form:

$$\frac{\Gamma, xRy, xRz \quad \begin{matrix}[yRz] \\ \vdots \\ \varphi\end{matrix} \quad \begin{matrix}[y = z] \\ \vdots \\ \varphi\end{matrix} \quad \begin{matrix}[zRy] \\ \vdots \\ \varphi\end{matrix}}{\varphi}$$

But we are interested in systems in Jaśkowski's format, where such kind of an extension is not natural in realization. So we should rather follow the strategy of Marx/Mikulas/Reynolds.

Linearity and Labels

The last remark concerns labels and linearity; in particular it is interesting to see how different strategies for obtaining linearity interact with the application of labels.

As for the strongly labelled systems the relationship is obvious. Existing systems show that linearity may be realized directly either by the representation of a condition of (weak) connectedness (Negri's rule) or its variant (9.1) ((CSL) of Castellini and Smaill). In fact, the form (9.2) may be also easily expressed with the help of the rule:

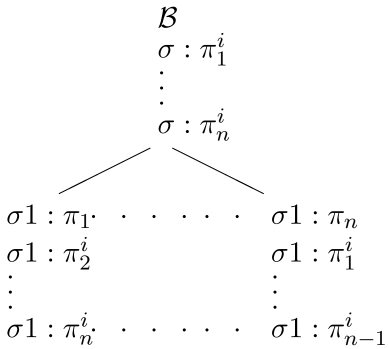
$$(SLL) \quad \frac{zRy, xRy, xRz, \Gamma \Rightarrow \Delta, yRz, y = z}{xRy, xRz, \Gamma \Rightarrow \Delta, yRz, y = z}$$

It is also possible to define suitable rules which realize in the strongly labelled setting all other previously discussed approaches. But it does make sense only if we need a tool for comparison of how they work, because we have already noticed that modular and structural rules with fixed branching factor are theoretically and practically superior. Note also, that if we want to introduce suitable rules to strongly labelled tableau or ND systems, in many cases negated r-formulae must be used, which is not always admitted in the basic formalization. It is already the case of (CSL) and (SLL), in contrast to Negri's rule which may be expressed in ND with no use of negation.

We have also presented weakly labelled systems for logics of linear frames (i.e. MSC), but except [149, 151], which will be introduced in the next section, there is no labelled system in the sense of Fitting (medium labelling) that would offer any treatment. Neither Massacci, nor Goré proposed any rules for linear logics. It may cause the impression that linearity is apparently not easily realizable with the help of some simple labelled-rules, and virtually needs some devices for rewriting labels. Recall that such a label is not only a name of a state but also encodes its localization in the attempted model. So in order to put, say $\sigma.i$ and $\sigma.j$ in a sequence, we must either rewrite one of them, or introduce some additional metalevel information concerning their accessibility.

In fact, it would be necessary if we followed the solutions of Rescher and Urquhart or Marx, Mikulas, Reynolds (where, in fact, labels are rewritten when π-rule is applied), or try to simulate the rules of Wansing or Indrzejczak. In all these cases, if more than one new world is being created, suitable rules simply list all the possibilities for ordering states. A direct realization of such a strategy in a system with medium labelling would consist in rewriting some labels already present on the branch in order to show that the new label denotes the world that goes before or between them in a sequence. But not all approaches need so strong adjustments which, in fact, change medium labelling into a kind of strong labelling.

For example, the strategy represented in Zeman/Goré/Shimura systems is in fact not difficult to express in a labelled system with no need for rewriting labels. For simplicity, we illustrate such an extension for **S4.3**. We may define the following labelled rule:



Obviously, by using such a rule we are linked to a tableau system because in ND (particularly in Jaśkowski's format) we have no natural means to express such branching. Although we can obtain also labelled counterparts of Goré's rule for **K4.3** it is not obvious how to find such rules for linear temporal logics with two interplaying modalities. As we have already noticed we also lose modularity. On the other hand, the spirit of Fitting's system is saved, where the extension of a label corresponds exactly to the accessibility relation. There is also no need for cut; we could prove completeness of such a system by Fitting's method, where each set of formulae with the same label must be downward saturated only. By the way, we could also simplify the shape of labels in such a system; similarly as in case of simplified TS for **S5** we may use just natural numbers as labels with accessibility relation expressed by $<$ or $\leq$ on them.

In our LND system described below we have followed a slightly different route, where the effect of linearity is obtained with the help of nonbranching ν-rules. They are based on the form (9.2) of condition of (weak) connectedness, similarly as π-rules of Marx, Mikulas and Reynolds. Consequently, our approach is close to the strategy realized in their tableau system. Instead of considering all possibilities for locating a new point, we simply check if the set of formulae present in some accessible point does not preclude the possibility of transfer for ν-formulae. Contrary to the Marx, Mikulas, Reynolds' realization of this strategy, in our system this test is not connected with the creation of a new point (that is with π-rule) but is being done independently, on already existing states which are not yet ordered by label shape. Since our system realizes the same strategy as [183], we also need cut, but – as we will show – an analytic form is sufficient. In our ND system such applications of cut are of course covered by $[RED]$. Proposed rules may be attached also to labelled tableau systems but in this case some explicit form of (analytic) cut is also necessary. Also the form of our rules is more natural in ND setting, because they are many-premise rules. In fact, the first version of the system for **S4.3** from [149] was based on labelled KE-system with analytic cut. The extension to some linear temporal logics was proposed in [151], this time with ND as a basis of formalization.

9.2 LND-System for S4.3

Because the idea of proposed formalization and rationale behind the rules are easier to expose on simple example, we start with a case study of the system for **S4.3**.

9.2.1 Characteristic Rule and Its Correctness

In order to obtain LND system for **S4.3**, we need to add only one rule to LND system for **S4**:

($\nu 3$) $\sigma : \nu_1^i, \tau : -\nu_1, \tau : \nu_2^i \;/\; \sigma : \nu_2^i$

It is a kind of conditional ν-rule, because the transfer of ν_2^i is possible if some conditions stated in the first two premises are satisfied. Intuitively, these premises exclude the situation that the value of τ is accessible from σ which, by strong connectedness, leads to recognition that the opposite holds and justifies the operation performed on the third premise. In what follows such rules for simplicity will be called 3-rules.

Some examples may help to understand what kind of deductive steps may be obtained by ($\nu 3$). Our rule allows us to make the following inferences:

from 1.1.2:$\Box p$, 1.4:$\neg p$ and 1.4:$\neg\Diamond q$ we infer 1.1.2:$\neg\Diamond q$

(where: $\sigma : \nu_1^i$:= 1.1.2:$\Box p$, $\tau : -\nu_1$:= 1.4:$\neg p$, $\tau : \nu_2^i$:= 1.4:$\neg\Diamond q$, and $\sigma : \nu_2^i$:= 1.1.2:$\neg\Diamond q$)

from 1.2:$\neg\Diamond p$, 1.1.5.2:p and 1.1.5.2:$\Box q$ we infer 1.2:$\Box q$

(where: $\sigma : \nu_1^i$:= 1.2:$\neg\Diamond p$, $\tau : -\nu_1$:= 1.1.5.2:p, $\tau : \nu_2^i$:= 1.1.5.2:$\Box q$, and $\sigma : \nu_2^i$:= 1.2:$\Box q$)

Let us note that the construction and the length of considered labels have no influence on the correctness of performed inferences. σ and τ in the schema of ($\nu 3$) are completely arbitrary. This feature, along with the multiplication of premises, makes 3-rules somewhat different than well known tableau rules. Ordinary rules for labelled systems have rather local character, they extend the label or jump to another one, but it is an operation on the chosen formula – we do not need to explore the whole branch in order to perform a rule. Already our rules for monotonic and congruent logics slightly departed from this schema because of having two premises. Our rules for linearity are even more nonstandard. They allow us to compare any pair of different labels and make a jump after some preliminary resolution step on formulae with these labels. Thus, there are two operations involved, made on two different formulae and two different labels at the same time. It makes the structure of labels inessential for these rules, and it is one more reason for connecting them rather with LND system than with tableaux; in case of most ND inference rules, the premises are searched for in the whole derivation before we can add conclusion(s). Note, that in the system of [149] for **S4.3** this arbitrariness of involved labels was tempered. The characteristic rule has side condition:

$$(\nu 3')\ \sigma : \nu_1^i,\ \tau : -\nu_1,\ \tau : \nu_2^i \ /\ \sigma : \nu_2^i, \text{ where } \mid \sigma \mid >1 \text{ and } \mid \tau \mid >1$$

This side condition was responsible for LAB(normality) with respect to strongly connected models. But the rule may be relaxed with no harm since **S4.3** is also adequate with respect to the class of frames with (nonstrict) linear order (cf. [112]). So the proof of soundness requires only demonstration of LAB(normality) of the new rule with respect to models satisfying the condition of dichotomy. For the sake of illustration we will prove axiom 3 and a thesis 3′ which is sometimes used in alternative axiomatizations of **S4.3** (cf. [60]).

1	SHØW: $1 : \Box(\Box p \to q) \lor \Box(\Box q \to p)$	$[6, LCOND]$
2	$1 : \neg\Box(\Box p \to q)$	*ass.*
3	$1.1 : \neg(\Box p \to q)$	$(2, L\pi E)$
4	$1.1 : \Box p$	$(3, L\alpha E)$
5	$1.1 : \neg q$	$(3, L\alpha E)$
6	SHØW: $1 : \Box(\Box q \to p)$	$[8, LNEC]$
7	$1.2 : \top$	*ass.*
8	SHØW: $1.2 : \Box q \to p$	$[11, LCOND]$
9	$1.2 : \Box q$	*ass.*
10	$1.2 : \Box p$	$(9, 5, 4, \nu 3)$
11	$1.2 : p$	$(10, LT)$

1	SHØW: $1 : \Diamond p \wedge \Diamond q \rightarrow \Diamond(\Diamond p \wedge q) \vee \Diamond(p \wedge \Diamond q)$	$[5, LCOND]$
2	$1 : \Diamond p \wedge \Diamond q$	$ass.$
3	$1 : \Diamond p$	$(2, L\alpha E)$
4	$1 : \Diamond q$	$(2, L\alpha E)$
5	SHØW: $1 : \Diamond(\Diamond p \wedge q) \vee \Diamond(p \wedge \Diamond q)$	$[17, LRED]$
6	$1 : \neg(\Diamond(\Diamond p \wedge q) \vee \Diamond(p \wedge \Diamond q))$	$ass.$
7	$1 : \neg\Diamond(\Diamond p \wedge q)$	$(6, L\alpha E)$
8	$1 : \neg\Diamond(p \wedge \Diamond q)$	$(6, L\alpha E)$
9	$1.1 : p$	$(3, L\pi E)$
10	$1.1 : \neg(p \wedge \Diamond q)$	$(8, L\nu E)$
11	$1.1 : \neg\Diamond q$	$(9, 10, L\beta E)$
12	$1.2 : q$	$(4, L\pi E)$
13	$1.2 : \neg(\Diamond p \wedge q)$	$(7, L\nu E)$
14	$1.2 : \neg\Diamond p$	$(12, 13, L\beta E)$
15	$1.1 : \neg\Diamond p$	$(11, 12, 14, \nu 3)$
16	$1.1 : \neg p$	$(15, LT)$
17	$\bot$	$(9, 16, L\bot I)$

We will show that $(\nu 3)$ is (LAB)normal in models satisfying dichotomy.

Lemma 9.1 $(\nu 3)$ *is (LAB)normal in* **S4.3**

PROOF Let $\langle \mathcal{W}, \mathcal{R}, V \rangle$ be any **S4.3**-model, where $U(\mathcal{D})$ is satisfied under interpretation $\Im$: $LAB(U(\mathcal{D})) \longrightarrow \mathcal{W}$ such that $\Im(\sigma)\mathcal{R}\Im(\tau)$, if $\sigma{=}\tau$ or τ is an extension of σ. Assume that premises of $(\nu 3)$ are satisfied under $\Im$, so $\Im(\sigma) \vDash \nu_1^i$, $\Im(\tau) \vDash -\nu_1$ and $\Im(\tau) \vDash \nu_2^i$; also $\Im(\tau) \vDash \Box_i \nu_2^i$ holds by transitivity. By assumption $\Im(\sigma)\mathcal{R}\Im(\tau)$ or $\Im(\tau)\mathcal{R}\Im(\sigma)$. If the first holds, then $\Im(\tau) \vDash \nu_1$ and we have a contradiction. So the second holds, hence $\Im(\sigma) \vDash \nu_2^i$, and the conclusion of $(\nu 3)$ is satisfied either. ■

It is enough to prove soundness of our system, hence this lemma, the completeness of both LND-**S4** and of axiomatic formalization of **S4.3**, and the first of the proofs above, guarantee together that our LND-**S4.3** is adequate. So we simply state:

Theorem 9.1 (Adequacy of LND-S4.3) $\vDash_{S4.3} \varphi$ *iff* $\vdash_{LND-S4.3} \varphi$

In fact, we can prove completeness of an analytic version of this system as well, but first we introduce the variants for other linear logics. For the time being we make some comparison with other systems to show that using nonbranching rules sometimes may save a lot of labor.

9.2.2 Efficiency

The following example can make it clear that in practice our rules often allow us to produce much shorter proofs than the branching ones. Let φ denote the following thesis of **S4.3**: $\Diamond(\Box p \wedge \neg q) \wedge \Diamond(\Box q \wedge \neg r) \to \Box(\Box r \to s) \vee \Box(\Box s \to p)$

1	SHØW: $1 : \varphi$	$[25, LRED]$
2	$1 : \neg\varphi$	$ass.$
3	$1 : \Diamond(\Box p \wedge \neg q) \wedge \Diamond(\Box q \wedge \neg r)$	$(2, L\alpha E)$
4	$1 : \neg(\Box(\Box r \to s) \vee \Box(\Box s \to p))$	$(2, L\alpha E)$
5	$1 : \Diamond(\Box p \wedge \neg q)$	$(3, L\alpha E)$
6	$1 : \Diamond(\Box q \wedge \neg r)$	$(3, L\alpha E)$
7	$1 : \neg\Box(\Box r \to s)$	$(4, L\alpha E)$
8	$1 : \neg\Box(\Box s \to p)$	$(4, L\alpha E)$
9	$1.1 : \Box p \wedge \neg q$	$(5, L\pi E)$
10	$1.1 : \Box p$	$(9, L\alpha E)$
11	$1.1 : \neg q$	$(9, L\alpha E)$
12	$1.2 : \Box q \wedge \neg r$	$(6, L\pi E)$
13	$1.2 : \Box q$	$(12, L\alpha E)$
14	$1.2 : \neg r$	$(12, L\alpha E)$
15	$1.3 : \neg(\Box r \to s)$	$(7, L\pi E)$
16	$1.3 : \Box r$	$(15, L\alpha E)$
17	$1.3 : \neg s$	$(15, L\alpha E)$
18	$1.4 : \neg(\Box s \to p)$	$(8, L\pi E)$
19	$1.4 : \Box s$	$(18, L\alpha E)$
20	$1.4 : \neg p$	$(18, L\alpha E)$
21	$1.3 : \Box q$	$(16, 14, 13, \nu 3)$
22	$1.3 : \Box p$	$(21, 11, 10, \nu 3)$
23	$1.3 : \Box s$	$(22, 20, 19, \nu 3)$
24	$1.3 : s$	$(23, LT)$
25	$\bot$	$(17, 24, L\bot I)$

For easier comparison with tableau proofs we have applied only elimination rules and indirect proof. Note that there is no use of $[LRED]$ in this example, except the first. In terms of trees it means that there is only one branch. The reader is asked to check how the proof of this thesis proceeds in Goré's system – we can assure you that no clever strategy can save us from creating at least 20 branches. The same applies to Rescher and Urquhart's system and to others aforementioned, including Marx, Mikulas, Reynolds' system where, regardless of the nonbranching rule for linearity, we have to use cut quite often.

In the example, after preliminary steps we have 4 π-formulae but one can easily define similar examples with arbitrary n π-formulae, where for each $i < n$ $\pi_i = \Box p_i \wedge \neg p_{i+1}$ and $\pi_n = \Box p_n \wedge \neg p_1$. The addition of more π-formulae only slightly (i.e. linearly) increases the length of a proof in our system, but leads to exponential growth of the number of branches in proof-trees from other formalizations. For example, if $n = 5$, then our proof will be only 5 lines longer but the smallest tree in Goré's system counts 70 branches. Taking **K4.3** as a point of reference (see the rules in the next section) leads to even more disastrous effects.

This is not the only class of theses for which we can provide similar analysis; the careful reader may try to prove in LND-**S4.3** and in Goré's system a thesis $\Diamond\Box p \wedge \Diamond\Box(p \to q) \to \Diamond\Box q$ (expected results: one application of $[LRED]$ = two branches in our system versus 12 branches in Goré's system, provided we use β-rules only when strictly necessary).

The source of the problem lies in the very construction of the branching rules. We briefly describe it, taking as an example the rule of Goré for **S4.3**. Contrary to ordinary π-rules for non-linear modal logics this one makes the calculus confluent; we never cancel any other π-formulae – they are simply pushed forward. It is theoretically satisfying but may be annoying in practice. If, say in **K**, we test π-formulae one by one, we can stop if we find the first subtree leading to $\bot$ on each branch, and delete other π-formulae by weakening. Thus we may obtain a short proof, even if it was preceded by a tedious search. But in case of **S4.3** we are processing all current π-formulae at once. There is no need for backtracking, but also no chance for shortcuts. Our nonbranching rules may be seen just as a kind of formalized shortcuts in proof construction.

So, from the practical point of view, we can construct shorter proofs by our rules quite often, the question is, can we find them quickly? In the next Chapter we will show, after the introduction of the improved proof-search procedure, that in the worst case our system behaves at least not worse than other systems we have discussed.

It seems also that nonbranching rules of this sort are more general than branching rules because they may be used not only in ND but also with other types of labelled systems. In fact, even some unlabelled systems may simulate such rules. For example, we may obtain Kashima's format TS for **S4.3** with the help of the following rule.

$$(K\text{-}3)\quad \frac{X[\{\Gamma, \Box\varphi\}, \{\neg\varphi, \Delta\}]}{X[\{\{\Gamma, \Box\varphi\}\neg\varphi, \Delta\}]}$$

This rule works in Kashima's system exactly as ($\nu 3$) on the ground of labelled ND or TS. But one should remember that any system, where linearity is realized with the help of ($\nu 3$) or some of its counterpart, needs (analytic) cut for completeness. It may be shown on simple example. Assume that in a derivation we have formulae: 1.1:$\Box p$, 1.2:$\Box q$ and no other U-formula with these labels. If we want to order these labels we must first use some cut (i.e. $[LRED]$ in ND) e.g. on formulae 1.2 : p and 1.2:$\neg p$. The latter gives wanted premise for the application of ($\nu 3$) to 1.2:$\Box q$; we infer 1.1:$\Box q$ and after application of (LT) we have that 1.1: is accessible from 1.2:. The former formula yields immediately the opposite connection between the labels.

9.3 LND for Linear Temporal Logics

LND-system for **S4.3** provided a convenient explanation of the general strategy of linearization. We have also illustrated some profits we may earn thanks to nonbranching rules. What is very important, this solution may be generalized also for other linear logics, including temporal ones, which is in contrast to some other approaches which work only for monomodal logics.

9.3.1 Formalization of Kt4.3

LND-system for **Kt4.3** demands an addition of four 3-rules to LND-**Kt4**:

$$\begin{array}{ll}
(3FF) & \sigma : \nu_1^F,\ \sigma : \nu_1,\ \tau : -\nu_1,\ \tau : \nu_2^F\ /\ \sigma : \nu_2,\ \sigma : \nu_2^F \\
(3PP) & \sigma : \nu_1^P,\ \sigma : \nu_1,\ \tau : -\nu_1,\ \tau : \nu_2^P\ /\ \sigma : \nu_2,\ \sigma : \nu_2^P \\
(3FP) & \sigma : \nu_1^F,\ \sigma : \nu_1,\ \tau : -\nu_1,\ \sigma : \nu_2^P\ /\ \tau : \nu_2,\ \tau : \nu_2^P \\
(3PF) & \sigma : \nu_1^P,\ \sigma : \nu_1,\ \tau : -\nu_1,\ \sigma : \nu_2^F\ /\ \tau : \nu_2,\ \tau : \nu_2^F
\end{array}$$

($3FF$) is a direct generalization of ($\nu 3$); it has one more premise (the second) because we must additionally exclude the possibility that σ and τ denote the same point. Two conclusions are the consequence of the lack of reflexivity. ($3PP$) is a mirror image of ($3FF$), whereas the presence of ($3FP$) and ($3PF$) is a result of symmetry of past and future. ($3FP$) has three conditional premises the same as ($3FF$); ($3PF$) – the same as ($3PP$), only the move of ν-formulae from the last premise has an opposite direction.

Perhaps the application of these 3-rules should be illustrated again, since they are the most complicated inference rules in our system. For example:

from $1.2.1[F] : \neg Fp,\ 1.2.1[F] : \neg p,\ 1.1[P] : p$ and $1.1[P] : Gq$ we may infer by $(3FF)$ both $1.2.1[F] : q$ and $1.2.1[F] : Gq$;

from $1.2.1[F] : \neg Fp,\ 1.2.1[F] : \neg p,\ 1.1[P] : p$ and $1.2.1[F] : \neg Pq$ we may infer by $(3FP)$ both $1.1[P] : \neg q$ and $1.1[P] : \neg Pq$. The shape of labels (their length and the fact that σ is an F-label and τ is a P-label) is incidental as well as the shape of ν-formulae occurring in our examples.

We illustrate our system in action with two derivations: open and closed.

1	SHOW: $1 : G(Gp \to q) \vee G(Gq \to p)$	
2	$1 : \neg(G(Gp \to q) \vee G(Gq \to p))$	*ass.*
3	$1 : \neg G(Gp \to q)$	$(2, L\alpha E)$
4	$1 : \neg G(Gq \to p)$	$(2, L\alpha E)$
5	$1.1[F] : \neg(Gp \to q)$	$(3, L\pi E)$
6	$1.2[F] : \neg(Gq \to p)$	$(4, L\pi E)$
7	$1.1[F] : Gp$	$(5, L\alpha E)$
8	$1.1[F] : \neg q$	$(5, L\alpha E)$
9	$1.2[F] : Gq$	$(6, L\alpha E)$
10	$1.2[F] : \neg p$	$(6, L\alpha E)$
11	SHØW: $1.1[F] : \neg p$	$[15, LRED]$
12	$1.1[F] : p$	*ass.*
13	$1.1[F] : q$	$(7, 12, 10, 9, 3FF)$
14	$1.1[F] : Gq$	$(7, 12, 10, 9, 3FF)$
15	$\bot$	$(8, 13, L\bot I)$
16	SHØW: $1.2[F] : \neg q$	$[20, LRED]$
17	$1.2[F] : q$	*ass.*
18	$1.2[F] : p$	$(9, 17, 8, 7, 3FF)$
19	$1.2[F] : Gp$	$(9, 17, 8, 7, 3FF)$
20	$\bot$	$(10, 18, L\bot I)$

In this example (which was already proven but in **S4.3**) there is no chance to close the overall argument. Two subsidiary derivations are completed which shows that neither the point denoted by 1.1 is accessible to the point 1.2, nor is the opposite. But it is still consistent that 1.1=1.2 since in both points p and q are evaluated in the same way, as false.

1	SHØW: $1 : PFp \to Fp \vee p \vee Pp$	$[3, LCOND]$
2	$1 : PFp$	*ass.*
3	SHØW: $1 : Fp \vee p \vee Pp$	$[12, LRED]$
4	$1 : \neg(Fp \vee p \vee Pp)$	*ass.*
5	$1 : \neg Fp$	$(4, L\alpha E)$
6	$1 : \neg p$	$(4, L\alpha E)$
7	$1 : \neg Pp$	$(4, L\alpha E)$
8	$1.1[P] : Fp$	$(2, L\pi E)$
9	$1.1.1[F] : p$	$(8, L\pi E)$
10	$1.1.1[F] : \neg p$	$(5, 6, 9, 7, 3FP)$
11	$1.1.1[F] : \neg Pp$	$(5, 6, 9, 7, 3FP)$
12	$\bot$	$(9, 10, L\bot I)$

Proving LAB(normality) of 3-rules with respect to trichotomic models is similar as in the case of $(\nu 3)$ with one small difference; we must use a notion of interpretation $\Im$ specified for temporal logics in the Definition 8.7 from Section 8.5.3. Their correctness is sufficient for completing a soundness proof. As for completeness it is enough to note that the thesis we have proved is, on the basis of **Kt4**, equivalent to L (cf. [60]), so it holds:

Theorem 9.2 (Adequacy of LND-Kt4.3) $\models_{Kt4.3} \varphi$ *iff* $\vdash_{LND\text{-}Kt4.3} \varphi$

9.3.2 Other Linear Logics

The system may be easily modified in several ways to capture other linear logics as well. We will sketch some possibilities.

We can add more rules to provide formalizations of stronger linear logics. For example, an addition of labelled versions of (DF) and/or (DP) yield three possible extensions of **Kt4.3** with future- or past-seriality, or both, whereas an addition of (LT) yields a temporal counterpart of **S4.3**. In the latter we may obviously simplify our 3-rules:

$(3FF')$ $\sigma : \nu_1^F,\ \tau : -\nu_1,\ \tau : \nu_2^F\ /\ \sigma : \nu_2^F$
$(3PP')$ $\sigma : \nu_1^P,\ \tau : -\nu_1,\ \tau : \nu_2^P\ /\ \sigma : \nu_2^P$
$(3FP')$ $\sigma : \nu_1^F,\ \tau : -\nu_1,\ \sigma : \nu_2^P\ /\ \tau : \nu_2^P$
$(3PF')$ $\sigma : \nu_1^P,\ \tau : -\nu_1,\ \sigma : \nu_2^F\ /\ \tau : \nu_2^F$

In these rules we get rid of one conclusion, since it is derivable by (LT); also one of the premises in 3-rules is superfluous, since in the presence of reflexivity we must capture only strong connectedness.

On the other hand, we may provide formalizations of monomodal **K4.3** and **KD4.3** by taking only $(3FF)$. All these modifications are straightforward. It seems that density can be also easily provided syntactically by the addition of inference rules modeled on suitable axioms. But for this system in analytic version we are not able to provide a completeness proof. On the other hand, it is not clear in what way we can obtain a formalization (analytic or not) of discrete linear order.

However, our system suffers from one serious restriction: it is not possible to obtain a formalization for bimodal logic where only one modality is linear. It is not enough to get rid of one pair of 3-rules and keep the other because they are interrelated. Moreover, in tree-models instead of linearity we necessarily have a weaker condition of right (or left) weak connectedness. Neither $(3FF)$ nor $(3PF)$ is (LAB)normal in models satisfying this condition and similarly for $(3PP)$ and $(3FP)$ in case of left weak connectedness. Moreover, in bimodal case we cannot obtain counterparts of our 3-rules with side condition (analogous to $(\nu 3')$) which are LAB(normal) in weakly connected frames, because in one label not the one accessibility relation is involved; we have discussed this point in Remark 8.5.

There is no difficulty in extending our LND-systems for linear logics to first-order classical or free version; one may just add labelled rules stated in Section 8.5.4. These extensions are complete, since Garson's [104] proof works for all logics which are axiomatized with the help of axioms corresponding to universal implications, and our ND formalizations of several versions of **QML** are strong enough to simulate his systems (cf. Chapter 2 in this respect).

9.4 Analytic Version of LND for Linear Logics

LND system for linear logics depicted above is not analytic, but fortunately, after introduction of suitable restrictions it is still complete. For simplicity, we consider here a system LAND1 – a labelled version of AND1, as described in Chapter 4 (cf. also the next Chapter). It is an LND system restricted only to elimination rules, including rules for elimination of modal/temporal constants, and to $[LRED]$ as the only proof construction rule. Moreover, we restrict the applications of $[LRED]$ to analytic ones; only such $\sigma : \psi$ may appear as S-line which is a (negated) subformula of the first S-formula. It is evident that LAND1 is a labelled version of AND1 from Chapter 4. Thus, for example, LAND1-**Kt4.3** is LAND1-**Kt4** with additional four 3-rules. Unfortunately, we cannot show completeness by simulation of suitable tableau systems, because there is nothing to be simulated.

Moreover, there are serious differences between our system and labelled tableaux that force us to resign, in the completeness proof, from the beautiful technique of Fitting, where the machinery of labels plays the main part and cut is dispensable. In Fitting's system there is a natural relation between labels (namely that of being an extension of) that corresponds to the accessibility relation of the attempted model. This correspondence is violated by our device of dividing labels into two groups; we have already underlined that extension of a label is not any more a sign of moving forward (cf. Remark 8.5). Anyway, this is something of minor importance and we can keeep control over the structure of attempted model if the rules have single step character. Hence, if we get rid of our four 3-rules we could quite easily prove completeness for **Kt4** or **Kt** by Fitting's method without any need for the cut. It will be done in the next Chapter.

However, the rules for linearity do not fit the picture of tableaux with medium labelling in an exact way, as we have already mentioned in Section 9.2.1. Massacci's single step rules have strictly local character, they extend the label or jump to another one, but always it is a direct neighbour (successor or predecessor of a given label). Fitting's original rules for transitive logics admit in fact long jumps but it is always bounded by the structure of a label. Moreover, in both cases, an application of a rule is an operation on the chosen formula – we do not need to explore the whole branch, and in this sense they have a local character. On the contrary, our rules for linearity are different in that they have a global character. To perform a rule we must compare any pair of different labels and make a jump after preliminary resolution step on formulae with these labels. Hence, 3-rules have two new features: the structure of the labels involved in 3-rules is inessential for application of these rules, and the premises are searched for in the whole derivation before we can add conclusion(s).[4]

At least two features of our completeness proof are the consequences of this peculiarity of 3-rules. First, they cause that the structure of labels is not in any obvious correspondence with the final accessibility relation of a falsifying model. For this reason we will define the accessibility relation for models not by reference to the label's structure but in a standard way (as in Henkin-style constructions) – it is even necessary in monomodal linear system like **S4.3** (see [149]). Second, the use of cut (in LND represented by applications of [$LRED$]) is necessary if we want to linearize the states

[4]The latter feature was in fact present also in our labelled rules for weak logics introduced in Section 8.6.

of a model; we have already illustrated this point in Section 9.2. But, as we shall see, all required applications of cut satisfy subformula property.

Below we will present a completeness proof for LAND1-**Kt4.3** which is adapted (with slight modifications) from Indrzejczak [151]. Yet another completeness proof for LAND1-**S4.3**[5] may be found in [149] but it works only for this particular logic, whereas a proof stated below is a more general construction.

In what follows we will be talking about derivations for arbitrary but fixed nonprovable φ. Also, for simplicity, we will identify sets of l-formulae uniformly labelled, with their labels. Thus we will often write $\varphi \in \sigma$ if $\sigma : \varphi$ belongs to a derivation. We will consider only derivations where every label is a subset of $\overline{\{\neg\varphi\}}$.

Definition 9.1 (Relative maximality)

1. σ is *consistent* iff no formula and its complement belong to σ;
2. σ is *saturated in* $\overline{\{\neg\varphi\}}$ iff for any $\psi \in \overline{\{\neg\varphi\}}$ either $\psi \in \sigma$ or $-\psi \in \sigma$;
3. σ is *maximal in* $\overline{\{\neg\varphi\}}$ iff it is consistent and saturated.

Note, that the simple consequence of a saturation is that for any α and β from $\overline{\{\neg\varphi\}}$:

(a) $\alpha \in \sigma$ iff $\alpha_1 \in \sigma$ and $\alpha_2 \in \sigma$ and
(b) $\beta \in \sigma$ iff $\beta_1 \in \sigma$ or $\beta_2 \in \sigma$.

σ is called downward saturated, if only weaker versions (left-to-right implications) of (a) and (b) are satisfied

We need a suitable counterpart of Lindenbaum lemma.

Lemma 9.2 (Maximalization) *Let $\mathcal{D}$ be an open derivation for φ; if $\sigma \subseteq U(\mathcal{D})$ is consistent, then there is an open derivation $\mathcal{D}'$ for φ such that $\sigma \subseteq U(\mathcal{D}')$ and σ is maximal in $\overline{\{\neg\varphi\}}$*

Proof Assume that σ is not maximal, otherwise the lemma holds trivially. First we apply all static rules to formulae in σ that were not used before. After this stage σ is not necessarily saturated because there may be some

[5] Actually, it is performed for labelled KE-system with ($\nu 3'$) not for LAND1 but that is not the point.

β-formulae not used so far (there were no minor premises for application of β-rule). If we still have such unused β-formulae, we choose the first one, introduce lacking minor premise and its complement as an S-line and indirect assumption respectively, and return to the application of static rules. It follows from consistency of σ that either new subderivation remains open or – in case of its closure – U($\mathcal{D}$) is extended by one of β_i in the outer subderivation. We continue saturation and, if necessary, repeat the above step with lacking minor premises and their complements for other unused β-formulae. Since σ is finite we must obtain an extension of $\mathcal{D}$ with σ downward saturated but not necessarily maximal in $\overline{\{\neg\varphi\}}$. If there are some formulae in $\overline{\{\neg\varphi\}}$ such that neither they nor their complements belong to σ, we deal with them similarly as with lacking components of β-formulae, i.e. we introduce a formula and its complement as an S-formula and assumption, and continue with static rules. Note that it is admissible in LAND1 because it is a derivation for φ, so all considered applications of [$LRED$] are analytic. Again, by consistency of σ either new subderivation remains open or outer derivation is consistently extended. Because $\overline{\{\neg\varphi\}}$ is finite, then by repeating this procedure we eventually produce an open extension of $\mathcal{D}$ such that $\sigma \subseteq$U($\mathcal{D}$) and is maximal in $\overline{\{\neg\varphi\}}$. ■

Let us define an accessibility relation on maximal labels:

Definition 9.2 (Accessibility and fulfillment)

Let σ and τ be maximal in $\overline{\{\neg\varphi\}}$:

1. τ is *accessible from* σ (or $\sigma \rhd \tau$) iff

   (a) if $\nu^F \in \sigma$, then $\nu^F \in \tau$ and $\nu \in \tau$, and

   (b) if $\nu^P \in \tau$, then $\nu^P \in \sigma$ and $\nu \in \sigma$.

2. (a) if $\pi^F \in \sigma$, then π^F is *fulfilled* iff there is some τ such that $\sigma \rhd \tau$ and $\pi \in \tau$.

   (b) if $\pi^P \in \sigma$, then π^P is *fulfilled* iff there is some τ such that $\tau \rhd \sigma$ and $\pi \in \tau$.

It is a characteristic feature of this proof that while building a derivation we build also a graph of accessibility of labels taking care of its linearity. We start with one-element sequence $\langle 1 \rangle$ and with every application of π-rule we modify it, either adding a new σ to the beginning, or to the end, or inserting this label between other elements of a sequence. Formally, we introduce the notion of a chain of labels.

Definition 9.3 (Chain)

1. Let **C** be a denumerable collection of labelled sets, such that each $\sigma \in$**C** is maximal in $\overline{\{\neg\varphi\}}$: **C** is *a chain in* $\overline{\{\neg\varphi\}}$ iff for any different labels τ and θ in **C**, θ is accessible from τ or τ is accessible from θ.

2. A chain **C** is fulfilled iff each π-formula from every label of **C** is fulfilled.

Note! "different labels" means labels of different shape, not necessarily labelling different sets of formulae. Just the contrary, since our proof admits infinite chains and $\overline{\{\neg\varphi\}}$ is always finite it may happen that the same set of formulae may reappear infinitely often but everytime with a different label.

Lemma 9.3 (Model Existence) *Let* **C** *be a fulfilled chain and let* $\mathfrak{M}_C = \langle \mathcal{T}, <, V \rangle$ *where:* $\mathcal{T} = LAB(\mathbf{C})$, $< \; = \; \rhd$, $V(p) = \{\sigma\colon p \in \sigma\}$ *then:*
$\mathfrak{M}_C$ *is a* **Kt4.3**-*model such that for any* $\sigma \in$**C** *and any* ψ *in* $\overline{\{\neg\varphi\}}$*:*

$\sigma \vDash \psi$ *iff* $\psi \in \sigma$*,*

The proof is standard, by induction on the length of φ; transitivity and linearity of $<$ is secured by the construction of **C**.

The last lemma suggests the way to achieve completeness; we must build an open derivation $\mathcal{D}$ such that U($\mathcal{D}$) will be eventually a chain. We will call a *full-derivation* for φ, any derivation which satisfies the stipulation that all labels in U($\mathcal{D}$) must be maximal in $\overline{\{\neg\varphi\}}$, before we add the new one (i.e. before we apply π-rule and transfer rules). By Lemma 9.2. we know that every consistent label may be maximalised. Every time all labels are maximal, we pick up the first unfulfilled π^F (or π^P) in U($\mathcal{D}$), apply $(L\pi E)$ and then apply $(L\nu E)$ and $(L4)$ to all ν-formulae in the parent of fresh σ. By "the first unfulfilled π^F (or π^P)" we mean the first such a formula from the top of a derivation – this way every such a formula will be eventually fulfilled. It means also that before we apply π-rule to some, e.g. $\sigma : \pi^F$, we first check if there is no τ such that $\sigma \rhd \tau$ and $\pi \in \tau$ or $\pi^F \in \tau$. In the former case our π is already fulfilled, in the latter case it is postponed to later stage. Similarly for every π^P. Then we return to maximalization.

Now we prove the key lemma:

Lemma 9.4 (Chain Extension) *Let U($\mathcal{D}$) in a full derivation $\mathcal{D}$ for φ be an unfulfilled n-element chain and let $\mathcal{D}'$ be the result of fulfilment of some π^F (or π^P), then U($\mathcal{D}'$) is an $n+1$-element chain.*

PROOF Assume we have finished the k-th stage of our construction and U($\mathcal{D}$) yields n-element chain $\mathbf{C}$=$\sigma_1, ...\sigma_n$. Pick up the first unfulfilled π-formula, for definiteness let our first unfulfilled π-formula be some $\sigma_i : \pi^F$, we prove our claim by induction on the number of successors of σ_i in $\mathbf{C}$.

Basis: If σ_i is the last item in the chain ($i = n$), we simply apply (πE) introducing a new son of σ_i. Next, by ν-rules and saturation we build a suitable maximal set. After that we apply ($LB\nu^P$) and ($LB4\nu^P$) to every ν^P in the new label to secure that it is accessible from σ_i and also from all its predecessors. This ends the proof for the basis.

To show that our claim holds if σ_i has $k < n$ successors, assume that it is satisfied for any $i < k$ successors. Consider all successors of σ_i; since our π^F is not yet fulfilled, by maximality $-\pi$ belongs to each one. Again by maximality either π^F or its complement is in the immediate successor of σ_i. If the first case holds, then we are done by induction hypothesis; similarly for further successors. So assume that in all successors of σ_i we have $-\pi^F$ (and $-\pi$). So we apply (πE) introducing a new label, being an F-child of σ_i. By an argument such as was used for a basis, we show that it must be accessible from σ_i and from all predecessors of σ_i. On the other hand, by ($3FF$) and ($3FP$) all successors of σ_i must be accessible to this new label, since in each case we have required assumptions for the application of these rules, namely π itself in the fresh successor of σ_i and $-\pi$ with $-\pi^F$ which is a suitable ν-formula, since complement of any π-formula is ν-formula and vice versa. So, as a result, we obtain $n+1$-element chain **C'** where the new label is inserted between σ_i and the immediate successor of σ_i from $\mathbf{C}$.

A similar argument applies to any unfulfilled π^P, but this time we make an induction on the number of predecessors and refer to ($3PP$) and ($3PF$). ■

Now we are in a position to prove a completeness of LAND1-**Kt4.3**.

Theorem 9.3 (Completeness) *If $\models_{Kt4.3} \varphi$, then LAND1-***Kt4.3** $\vdash \varphi$.

PROOF As usual we prove the contrapositive. By assumption no derivation for φ closes, no full derivation in particular. It means that notwithstanding how often we must close some subderivation and start again the process of maximalization we eventually must find an open one. First we extend $1.\Gamma$ until we get a set maximal in $\overline{\{\neg\varphi\}}$ and if there are no π-formulae we can stop, otherwise we select the first unfulfilled π-formula, apply a suitable rule and create a new (F- or P-) label 1.1. Alternately applying ν-rules and maximalization we obtain two-element chain. If it is not fulfilled, then we follow the procedure for constructing full derivations, building a chain in $\overline{\{\neg\varphi\}}$. Now, this process either terminates or not. In the first case we have n-element fulfilled chain, in the second our procedure guarantees that every π-formula must be eventually used, so we get a fulfilled chain in the omega-step. Either way, by Lemma 9.3. we can extract a falsifying model for φ from the chain. ■

The proof we have provided is constructive and general enough to be applied to **KtD4.3** and **KtT4.3**. For monomodal logics **K4.3**, **KD4.3** and **S4.3** it works too, with omission of details concerning the second modality. Although it involves some procedure for proof search it is not very practical. The fact that we may run infinite search is not the main problem; we may easily provide some mechanism for loop-control and obtain a decision procedure – we will return to these matters in the next Chapter. The problem is with maximalization of every label; it makes this procedure extremely inefficient. Every time before we apply $(L\pi E)$ to enlarge a model, we must first perform a lot of superfluous inferences. From the standpoint of quick proof they are unnecessary and they extremely increase the complexity of a derivation. Moreover, in the context of modal logics, an introduction of these additional formulae may lead to the creation of additional states in a model – we will discuss this problem in the next Chapter. In fact, an application of 3-rules is in our procedure rather an additional way of obtaining a closure of subderivation, not a way to enrich a label with new ν-formulae. It is a result of preliminary maximalization of every label in $\overline{\{\neg\varphi\}}$; either respective labels contain conclusions from applications of 3-rules (so it does not make sense to apply them) or they contain their complements which yield $\bot$ and closure of current subderivation. In the next Chapter we will introduce a more efficient procedure based on downward saturation only.

9.5 Extensions and Limitations

Our characteristic 3-rules may be easily adapted to labelled RND setting in both versions: with local and global labels. Suitable clausal rules for LLRND are displayed below:

$(LL3)$	$\Gamma, \sigma : \nu_1^i \ ; \Delta, \tau : -\nu_1 \ ; \Theta, \tau : \nu_2^i \ / \ \Gamma, \Delta, \Theta, \sigma : \nu_2^i$
$(LL3')$	$\Gamma, \sigma : \nu_1^i \ ; \Sigma, \sigma : \nu_1 \ ; \Delta, \tau : -\nu_1 \ ; \Theta, \tau : \nu_2^i \ /$ $\Gamma, \Sigma, \Delta, \Theta, \sigma : \nu_2$
$(LL3'.4)$	$\Gamma, \sigma : \nu_1^i \ ; \Sigma, \sigma : \nu_1 \ ; \Delta, \tau : -\nu_1 \ ; \Theta, \tau : \nu_2^i \ /$ $\Gamma, \Sigma, \Delta, \Theta, \sigma : \nu_2^i$
$(LL3\text{-}Te)$	$\Gamma, \sigma : \nu_1^i \ ; \Delta, \tau : -\nu_1 \ ; \Theta, \sigma : \nu_2^j \ /$ $\Gamma, \Delta, \Theta, \tau : \nu_2^j, \quad \text{where } i \neq j \in \{F, P\}$
$(LL3'\text{-}Te)$	$\Gamma, \sigma : \nu_1^i \ ; \Sigma, \sigma : \nu_1 \ ; \Delta, \tau : -\nu_1 \ ; \Theta, \sigma : \nu_2^j \ /$ $\Gamma, \Sigma, \Delta, \Theta, \tau : \nu_2, \quad \text{where } i \neq j \in \{F, P\}$
$(LL3'\text{-}Te4)$	$\Gamma, \sigma : \nu_1^i \ ; \Sigma, \sigma : \nu_1 \ ; \Delta, \tau : -\nu_1 \ ; \Theta, \sigma : \nu_2^j \ /$ $\Gamma, \Sigma, \Delta, \Theta, \tau : \nu_2^j, \quad \text{where } i \neq j \in \{F, P\}.$

$(LL3)$ added to LLRND-**S4** yields LLRND-**S4.3**, $(LL3')$ and $(LL3'.4)$ added to LLRND-**K4** yield LLRND-**K4.3**. In case of temporal logics **Kt4.3** and **KtT4.3** we need additionally symmetric variants: $(LL3\text{-}Te)$ for the latter and the remaining two for the former. A demonstration of soundness is left to the reader; completeness follows from the fact that 3-rules stated for LND are just special cases of clausal 3-rules.

In GLRND we need the following 3-rules:

$(GL3)$	$\sigma : \Gamma, \nu_1^i \ ; \tau : -\nu_1 \ ; \tau : \nu_2^i, \ldots, \nu_k^i \ / \ \sigma : \Gamma, \nu_2^i, \ldots, \nu_k^i$
$(GL3')$	$\sigma : \Gamma, \nu_1^i \ ; \sigma : \Delta, \nu_1 \ ; \tau : -\nu_1 \ ; \tau : \nu_2^i, \ldots, \nu_k^i \ /$ $\sigma : \Gamma, \Delta, \nu_2, \ldots, \nu_k$
$(GL3'.4)$	$\sigma : \Gamma, \nu_1^i \ ; \sigma : \Delta, \nu_1 \ ; \tau : -\nu_1 \ ; \tau : \nu_2^i, \ldots, \nu_k^i \ /$ $\sigma : \Gamma, \Delta, \nu_2^i, \ldots, \nu_k^i$
$(GL3\text{-}Te)$	$\sigma : \nu_1^i \ ; \tau : \Gamma, -\nu_1 \ ; \sigma : \nu_2^j, \ldots, \nu_k^j \ /$ $\tau : \Gamma, \nu_2^j, \ldots, \nu_k^j, \quad \text{where } i \neq j \in \{F, P\}.$
$(GL3'\text{-}Te)$	$\sigma : \nu_1^i \ ; \sigma : \nu_1 \ ; \tau : \Gamma, -\nu_1 \ ; \sigma : \nu_2^j, \ldots, \nu_k^j \ /$ $\tau : \Gamma, \nu_2, \ldots, \nu_k, \quad \text{where } i \neq j \in \{F, P\}.$
$(GL3'\text{-}Te4)$	$\sigma : \nu_1^i \ ; \sigma : \nu_1 \ ; \tau : \Gamma, -\nu_1 \ ; \sigma : \nu_2^j, \ldots, \nu_k^j \ /$ $\tau : \Gamma, \nu_2^j, \ldots, \nu_k^j, \quad \text{where } i \neq j \in \{F, P\}.$

They are in one-to-one correspondence to 3-rules stated for LLRND so exactly the same logics may be formalized with their help on the basis of GLRND. We leave an adequacy proof to the reader as well.

The following example of a proof of the thesis $\varphi = \Diamond p \wedge \Diamond q \rightarrow \Diamond(p \wedge q) \vee \Diamond(\Diamond p \wedge q) \vee \Diamond(p \wedge \Diamond q)$ in LLRND-**K4.3** provides an illustration of application of clausal 3-rules:

1	SHØW: $1 : \varphi$	$[20, SUB]$
2	$1 : \neg\varphi$	*ass.*
3	$1 : \Diamond p \wedge \Diamond q$	$(2, LL\alpha)$
4	$1 : \neg(\Diamond(p \wedge q) \vee \Diamond(\Diamond p \wedge q) \vee \Diamond(p \wedge \Diamond q))$	$(2, LL\alpha)$
5	$1 : \Diamond p$	$(3, LL\alpha)$
6	$1 : \Diamond q$	$(3, LL\alpha)$
7	$1 : \neg\Diamond(p \wedge q)$	$(4, LL\alpha)$
8	$1 : \neg\Diamond(\Diamond p \wedge q)$	$(4, LL\alpha)$
9	$1 : \neg\Diamond(p \wedge \Diamond q)$	$(4, LL\alpha)$
10	$1.1 : p$	$(5, LL\pi E)$
11	$1.2 : q$	$(6, LL\pi E)$
12	$1.1 : \neg(p \wedge q)$	$(7, LLExp\text{-}K)$
13	$1.1 : \neg(p \wedge \Diamond q)$	$(9, LLExp\text{-}K)$
14	$1.1 : \neg p,\ 1.1 : \neg q$	$(12, LL\beta)$
15	$1.1 : \neg p,\ 1.1 : \neg\Diamond q$	$(13, LL\beta)$
16	$1.2 : \neg(\Diamond p \wedge q)$	$(8, LLExp\text{-}K)$
17	$1.2 : \neg\Diamond p,\ 1.2 : \neg q$	$(16, LL\beta)$
18	$1.1 : \neg p,\ 1.2 : \neg q$	$(15, 14, 11, 17, LL3')$
19	$1.1 : \neg p$	$(11, 18, Res)$
20	$\bot$	$(10, 19, Res)$

One should note that our approach to linear logic may be extended to other modal logics which are semantically characterized by universal implications (cf. Section 1.1.5.) We will illustrate the point with two examples: **S4F** and **S4R**. Suitable semantic conditions may be stated as universal implications:

$F \ \ \forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \to \mathcal{R}yx \vee \mathcal{R}zy)$

$R \ \ \forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \to x = z \vee \mathcal{R}yz)$

But for our needs the following equivalents are more accurate:

$F' \ \ \forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \wedge \neg\mathcal{R}yx \to \mathcal{R}zy)$

$F'' \ \ \forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \wedge \neg\mathcal{R}zy \to \mathcal{R}yx)$

$R' \ \ \forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \wedge x \neq z \to \mathcal{R}yz)$

$R'' \ \ \forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \wedge \neg\mathcal{R}yz \to x = z)$

These conditions may be expressed by the following rules:

(F') $\sigma\tau : \nu_1^i,\ \sigma : -\nu_1,\ \sigma\theta : \nu_2^i,\ /\ \sigma\tau : \nu_2^i$
(F'') $\sigma\theta : \nu_1^i,\ \sigma\tau : -\nu_1,\ \sigma\tau : \nu_2^i,\ /\ \sigma : \nu_2^i$
(R') $\sigma : \varphi,\ \sigma\theta : -\varphi,\ \sigma\tau : \nu^i,\ /\ \sigma\theta : \nu^i$
$(1R'')$ $\sigma\tau : \nu^i,\ \sigma\theta : -\nu,\ \sigma\theta : \varphi\ /\ \sigma : \varphi$ or
$(2R'')$ $\sigma\tau : \nu^i,\ \sigma\theta : -\nu,\ \sigma : \varphi\ /\ \sigma\theta : \varphi$

To obtain LND-**S4F** we must add to LND-**S4** either (F') or (F''); for LND-**S4R** we need one of (R'), $(1R'')$ or $(2R'')$. Completeness of nonanalytic version follows from the fact that respective axioms are easily provable with the help of one application of any of the stated rules – we leave it to the reader.

To establish (LAB)normality of these rules one should note that in all cases σ corresponds to x, $\sigma\tau$ to y and $\sigma\theta$ to z. Informally, in both rules corresponding to F the first two premises express that $\neg\mathcal{R}yx$ or $\neg\mathcal{R}zy$, whereas in the rules corresponding to R, that $x \neq z$ or $\neg\mathcal{R}yz$. Accessibility of y and z from x is just encoded in the structure of respective labels. The rest is just a result of rewriting of ν-formulae (in cases where the succedent of respective condition is a relational atom) or any formulae (in case of identity) from one label to another. Soundness follows immediately from (LAB)normality of these rules.

We hope that both systems may be restricted to an analytic version, similarly as LND systems for linear logics, but for the time being we are unable to offer a constructive completeness proof.[6] However, it is certain that in case of analytic formalization of **S4R** we should use either (R') or both $(1R'')$ and $(2R'')$. In the latter case symmetry of identity causes that we need two rules to move all the formulae from one label to the other and reversely.

Finally, note that we can obtain in this way only a formalization of monomodal versions of respective logics, since in both cases we must have resigned from single-step character of rules.

The last remark reffers to subtle questions of applicability of Fitting's-like labels to multimodal logics in general (cf. Section 8.3 where some systems of this sort are mentioned). Baldoni [18] compares medium labelling with his approach based on strong labelling and shows that the latter is superior in many respects. It is true that with the help of medium labelling we cannot provide such a general and extensive approach as is possible by means of strong labelling. For example, we were able to show that some

[6] Analytic labelled system for these logics may be found in Leszczyńska [175] but it is based on the rules of different character.

specific logics semantically characterized by universal implications are formalizable in this way but we cannot provide a general schema of formalization for all such logics (as e.g. in [194]), even for monomodal ones. Medium labelling seems to be strongly sensitive with respect to specific properties of accesibility relation, whereas strong labelling is rather independent of them and based on the general architecture of any frame.

In multimodal setting limitations of medium labelling are even more severe. Baldoni gives an example showing that some multimodal interactive logics are not formalizable because it may appear that we have a label on a branch but there is no way to obtain all its sublabels. Consider a multimodal logic with the axiom $[a][b]\varphi \to [c]\varphi$. In Baldoni's TS it is formalised by a rule (BR): xR_cy / xR_az; zR_by with z being a new label on a branch. We may easily obtain a tableau proof of $[a]p \wedge \langle c\rangle q \to \langle a\rangle p$ in his system in the following way:

1	$1 : \neg([a]p \wedge \langle c\rangle q \to \langle a\rangle p)$	
2	$1 : [a]p \wedge \langle c\rangle q$	$(1, L\alpha)$
3	$1 : \neg\langle a\rangle p$	$(1, L\alpha)$
4	$1 : [a]p$	$(2, L\alpha)$
5	$1 : \langle c\rangle q$	$(2, L\alpha)$
6	$1R_c2$	$(5, L\pi E)$
7	$2 : q$	$(5, L\pi E)$
8	$1R_a3$	$(6, BR)$
9	$3R_b2$	$(6, BR)$
10	$3 : \neg p$	$(3, 8, L\nu E)$
11	$3 : p$	$(4, 8, L\nu E)$
11	$\bot$	$(10, 11)$

In the system based on medium labelling we rather stop with $1.1[c] : q$ in line 6. One may think about some rule like, e.g. $\sigma : \langle c\rangle\varphi$ / $\sigma : \langle a\rangle\langle b\rangle\varphi$ to overcome a problem. An application of such a rule yields $1 : \langle a\rangle\langle b\rangle q$ from line 5 and then, by application of $(L\pi E)$, we got $1.1[a] : \langle b\rangle q$ which enables application of $(L\nu E)$ to lines 3 and 4 and closure of a branch. But such a solution is rather artificial and nonanalytic.

Still we may generalize the apparatus of Fitting's labels to other multimodal interactive logics except temporal ones. For instance, in Chapter 5 we have given as an example epistemic logics formalized by means of inclusion axioms of the form $\Box_i\varphi \to \Box_j\varphi$. In models they correspond to the condition $\mathcal{R}_j \subseteq \mathcal{R}_i$. In order to obtain LND for such logics it is sufficient to strengthen (νE) (and possibly other transfer rules like (4)) for ν^i, admitting for conclusion $\sigma.k : \nu$ not only with respect to every i- but also with respect to every j-label $\sigma.k$.

Such a rule is simple, natural and provide a solution to the second of Baldoni's example showing weakness of medium labelling. Consider the logic with two inclusion axioms: $[a]\varphi \rightarrow [c]\varphi$ and $[b]\varphi \rightarrow [c]\varphi$. We can easily provide a proof of $[a]p \wedge \langle c\rangle q \rightarrow \langle b\rangle p$ in TS enriched with additional ν-rules.

1	$1 : \neg([a]p \wedge \langle c\rangle q \rightarrow \langle b\rangle p)$	
2	$1 : [a]p \wedge \langle c\rangle q$	$(1, L\alpha)$
3	$1 : \neg\langle b\rangle p$	$(1, L\alpha)$
4	$1 : [a]p$	$(2, L\alpha)$
5	$1 : \langle c\rangle q$	$(2, L\alpha)$
6	$1.1[c] : q$	$(5, L\pi E)$
7	$1.1[c] : \neg p$	$(3, L\nu E)$
8	$1.1[c] : p$	$(4, L\nu E)$
9	$\perp$	$(7, 8)$

Lines 7 and 8 are deduced by ν-rule applied just to this label which is admissible by inclusion of respective accessibility relations.

But to be honest it must be said that our solution deals with just a specific example and not with general difficulty pointed out by Baldoni, namely that in medium labelling one has no natural means to identify different labels. The problem is that different $\mathcal{R}$-paths may lead in a model to the same world. In medium labelling these are represented by different labels, whereas in strong labelling we introduce a unique label for each state, leaving details concerning the structure of a model to be represented independently by relational formulae. These limitations of medium labelling are serious if one is interested in really uniform and extensive treatment of big classes of modal logics. To overcome this problem we will introduce hybrid logics in the last two chapters. But if one needs easy going systems for the most important modal logics it is still arguable that Fitting's approach is better. Hence, before we go on to the most general approach we focus on the problem of analytic versions of LND.

Chapter 10

Analytic Labelled ND and Proof Search

LND system from Chapter 8 allows us to construct simple derivations but is not analytic. We have mentioned in Section 8.5 that one may obtain complete, universal and analytic version similarly as in Chapter 4; by stepwise simulation of every tableau with the help of only elimination rules and analytic applications of cut (i.e. [$LRED$]). This way we have obtained in the last Chapter, a labelled version of AND1 called LAND1. Because labelled TS's of Fitting, Massacci or Goré are known to be complete (several proofs may be found e.g. in [93], [185, 186] or [117]) we have an indirect completeness proof for LAND1 for many modal logics, including all basic normal ones. Since mentioned authors do not provide labelled TS's for e.g. temporal logics in general, and for linear (monomodal and temporal) logics, we have dealt with the problem in the last Chapter. We have noted however, that offered solutions, although theoretically satisfying, are in practice rather complex.

At present, on the basis of the results from Chapter 4, we develop more efficient proof search procedures which yield completeness and decidability of LAND1 for many logics. This will be done in stages because stronger logics need more complex proof search algorithms. In Section 10.1 we introduce some preliminary notions and two basic procedures which will be used as modules in the subsequent algorithms. Section 10.2. provides the simplest solution for logics **K, D, T** and some discussion of optimization techniques. In Section 10.3 we define a procedure for some transitive logics, whereas in the next one we deal with symmetric and Euclidean logics. In this context also problems connected with loop control are discussed. Finally, in

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*, Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_10,

Section 10.5, we introduce algorithms for linear logics. All completeness proofs based on our procedures require only downward saturation of sets of formulae with the same label, in contrast to the proof from the preceding Chapter based on less efficient technique of relative maximalization.

We focus only on the problem of proof search for normal basic logics and linear (temporal) logics. Suitable algorithms for regular and monotonic basic logics may be easily obtained on the basis of provided solutions. The problem for congruent logics seems to be harder to solve. We leave open also the question of direct proof search procedures for first-order modal logics and an exact definition of labelled counterpart of AND2.

10.1 Analytic LND

In what follows we will be using as a basis a system LAND1, i.e. LND restricted only to elimination rules and to (analytic) [$LRED$] as the only proof construction rule. It is evident that such a system is a labelled version of AND1 from Chapter 4, hence the name. We have already dealt with such a system for linear logics in the preceding Chapter.

It should be noted however that the problem of analyticity in case of labeled systems is not as obvious as in case of nonlabelled ones. Applications of admissible rules do not introduce other formulae than elements of $\overline{\{\neg\varphi\}}$ (where φ is a formula we are attempting to prove) but $(L\pi E)$ systematically introduces new labels into a derivation. So subformula property holds but there is no substructure property with respect to some predefined and finite set of labels. We will see that in case of logics that are not transitive for any $\mathcal{D}$ the set of LAB($U(\mathcal{D})$) is always finite. In case of transitive logics this property does not hold but it may be obtained with the help of additional techniques. In fact, for constructive completeness proof of our systems the termination of respective procedures is not necessary. It is sufficient if they are fair, in the sense that for every action which must be performed there is a stage where it will be eventually performed. It is the case of constructive completeness proofs for first-order logics, where undecidability excludes termination. But we consider only decidable logics so it will be a good thing to obtain termination as well.

We base our completeness and decidability results on strategies introduced by Fitting [93, 92] and refined for single step tableaux by Massacci [185, 186]. An excellent presentation is provided by Goré [117]. We follow it quite closely, in particular for simplicity we limit ourconsiderations to monomodal basic logics. Briefly, in order to prove completeness of analytic formalization of suitable logic **L** one must do the following:

1. Define a fair proof-search procedure which provides either a proof or a tableau/derivation with open but completed set of formulae; in case we want a decision procedure it must be also terminating.

2. Define suitable labelled version of Hintikka set and prove model-existence lemma for it; in particular, it must be shown that constructed model is really a model for **L** (matching lemma).

3. We must show that a (completed) set of labelled formulae obtained by the application of proof-search procedure is really a Hintikka set; for that we need to know that this set is strongly generated (generation lemma).

From this completeness follows easily by contraposition. Assume that $\nvdash_L \varphi$, then, by 1. we obtain an open completed set of l-formulae containing $1 : \neg\varphi$. From 3. and generation lemma follows that this set is a Hintikka set. From 2. follows the existence of **L**-model satisfying $\neg\varphi$, hence φ is not valid in **L**.

First, we briefly recall things needed to establish 2. and 3. and then we focus on point 1. The presentation of 2. and 3. is based on [117] and [186] hence we will pay attention only to things connected with temporal logics or modifications connected with transfer of these solutions from the context of TS to ND setting. The main concern of this Chapter is a detailed presentation of questions connected with defining fair and terminating proof search procedures for ND systems. We have shown in Chapter 4 that the structure of ND derivations is connected with some subtleties already in **CPL**; modal logics provide additional factors that make hard a direct transfer of procedures defined for TS.

10.1.1 Labelled Hintikka Sets

In order to provide a constructive proof of completeness of LAND1-**L** we must generalize the notions of l-saturation, downward saturation and Hintikka set, introduced in Chapter 4 for classical logic. For simplification we assume, as in the completeness proof from the previous Chapter, that σ, τ, θ will be used not only for denoting labels but also for denoting sets of formulae labelled by suitable item. A context makes it clear which usage is being adressed.

Definition 10.1 (L-saturated sets)

Let X be a strongly generated set of labelled formulae and $\sigma \subseteq FOR(X)$:

A. σ is *Ll-saturated* in **L** if the following hold:

1. if $\neg\neg\varphi \in \sigma$, then $\varphi \in \sigma$,

2. if $\alpha \in \sigma$, then $\alpha_1 \in \sigma$ and $\alpha_2 \in \sigma$,

3. if $\beta \in \sigma$ and $-\beta_i \in \sigma$, then $\beta_j \in \sigma$; for $i \neq j \in \{1, 2\}$.

B. σ is *L-downward saturated* in **L**, if it is Ll-saturated in **L**, but instead of condition 3. it satisfies:

3'. if $\beta \in \sigma$, then $\beta_1 \in \sigma$ or $\beta_2 \in \sigma$

C. X is L-Hintikka set in **L** iff for every $\sigma \subseteq FOR(X)$:

1. σ is *L-downward saturated* in **L**,

2. σ is **L**-consistent, i.e. if some formula belongs to σ, then its complement does not.

3. if $\sigma : \nu^i \in X$, then $\tau : \nu \in X$, for every τ such that $\sigma \triangleright \tau$,

4. if $\sigma : \pi^i \in X$, then $\tau : \pi \in X$, for some τ such that $\sigma \triangleright \tau$.

For temporal logics we have the following conditions instead of conditions *3.* and *4.*:

3. if $\sigma : \nu^F \in X$, then $\tau : \nu \in X$, for every τ such that $\sigma \triangleright \tau$,

4. if $\sigma : \pi^F \in X$, then $\tau : \pi \in X$, for some τ such that $\sigma \triangleright \tau$,

5. if $\sigma : \nu^P \in X$, then $\tau : \nu \in X$, for every τ such that $\tau \triangleright \sigma$,

6. if $\sigma : \pi^P \in X$, then $\tau : \pi \in X$, for some τ such that $\tau \triangleright \sigma$.

Following Goré [117] we will use a symbol $\triangleright$ for binary relation on the set of strongly generated labels. A definition depends on the properties of relation of accessibility in suitable class of frames and is specified for every logic in the table.

Logic	$\sigma \triangleright \tau$
K	$\tau = \sigma.k$
D	$\tau = \sigma.k$ or ($\tau = \sigma$ and σ is final)
T	$\tau = \sigma.k$ or $\tau = \sigma$
K4	$\tau = \sigma.\theta$ and $\mid \theta \mid \geq 1$
KB	$\tau = \sigma.k$ or $\sigma = \tau.k$
K5	$\tau = \sigma.k$ or ($\mid \tau \mid \geq 2$ and $\mid \sigma \mid \geq 2$)
KD4	($\tau = \sigma.\theta$ and $\mid \theta \mid \geq 1$) or ($\tau = \sigma$ and σ is final)
KDB	($\tau = \sigma.k$ or $\sigma = \tau.k$) or $\sigma = \tau = 1$ is the only label
KD5	like for **K5** or $\sigma = \tau = 1$ is the only label
K45	($\tau = \sigma.\theta$ and $\mid \theta \mid \geq 1$) or ($\mid \tau \mid \geq 2$ and $\mid \sigma \mid \geq 2$)
K4B	$\triangleright$ is universal relation and there are at least two different labels
KD45	like for **K45** or $\sigma = \tau = 1$ is the only label
B	$\tau = \sigma.k$ or $\sigma = \tau.k$ or $\tau = \sigma$
S4	($\tau = \sigma.\theta$ and $\mid \theta \mid \geq 1$) or $\tau = \sigma$
S5	$\triangleright$ is universal
Kt	$\tau = \sigma.k$ and $\sigma.k$ is an F-label or $\sigma = \tau.k$ and $\tau.k$ is a P-label
Kt4	$\triangleright$ is a transitive closure of $\triangleright$ for **Kt4**

In case of serial logics a final label is any label with no children. We have added also a definition of $\triangleright$ for **Kt** and **Kt4**. For the latter we could not use a characterisation like for **K4** since the extension of label σ is not necessarily in relation $\triangleright$ to σ. For example, let 1.2 be a P-label and 1.2.1 an F-label, then $1 \triangleright 1.2.1$ does not hold.

Now we are in a position to define a special kind of frame:

Definition 10.2 (Hintikka frame)

Let X be L-Hintikka set for **L**, then Hintikka frame for **L** is an ordered pair $\mathfrak{F}_H = \langle \mathcal{W}, \{\mathcal{R}\}\rangle$, where: (a) $\mathcal{W} = LAB(X)$, (b) $\mathcal{R} = \triangleright$.

For all basic normal logics and for respective temporal logics the following holds:

Lemma 10.1 (Matching) *$\mathfrak{F}_H$ for* **L** *belongs to $\mathcal{F}$ characterising* **L**.

Proof requires systematic checking that $\triangleright$ satisfies conditions for $\mathcal{R}$. One may find in Goré [117] and Massacci [186] a demonstration of some cases. For temporal logics it is straightforward, in particular, transitivity follows by definition.

For completeness proof the key result is provided by model-existence lemma which is an analogon of lemma 4.1.

Lemma 10.2 (Satisfiability of L-Hintikka sets) *Let X be an L-Hintikka set for* **L**, *and* $\mathfrak{M}_H$ — *an* **L**-*model on* $\mathfrak{F}_H$ *such that* $V(p) = \{\sigma \subseteq FOR(X) : p \in \sigma\}$, *then:* $\sigma : \psi \in X$ *implies* $\sigma \vDash \psi$,

We leave a standard proof by induction on the length of a formula to the reader.[1]

By Lemma 10.2 for completeness it is sufficient to define a proof search procedure such that in case of a formula nonprovable in **L**, U($\mathcal{D}$) appears to be an L-Hintikka set for **L**. We will introduce four such algorithms since the presence (or lack) of transitivity or symmetry or euclideaness, requires significant modifications. Independently of the kind of logic we put some constraints on our procedures. Sets of formulae obtained by them must be completed in the sense that all possible rules were applied to all formulae. On the other hand, we do not want to have more than one occurence of each formula and there are possible situations where we did not applied a rule to some formula but a conclusion is already present. A formula to which no rule applies, since every consequence of every rule is already in the set, is classified as *used formula*. Note that the definition of used formula admits a situation where a consequence of some rule application may be already present in the completed set even if we haven't applied this rule to the premise. In particular, we are not forced to an application of π-rule and enlarging of a model when π is already present in some accessible label. Moreover, we add a constraint that such label-generating rules may be applied only once to the same premise. We note one more result:

Lemma 10.3 (Generation) *U($\mathcal{D}$) at every stage of any derivation is a strongly generated set.*

It is easy to check that the set of labels of any derivation with the relation $\prec$, where $\sigma \prec \tau$ iff τ is a child of σ, is finitely generated tree; it follows from the property of being strongly generated. What is perhaps not so obvious is the fact that during the process of closing a subderivation we do not loose this property. But it follows from the fact that children of some labels are always present either on the same level of derivation where parents occur or

[1] Note that in case ψ is a negated formula $\neg\varphi$ we must consider all cases which φ may obtain.

in some nested subderivations. Hence, it is impossible that after a closure of some subderivation we obtain a situation to the effect that some, e.g. σ and $\sigma.k.n$ is present in the current U($\mathcal{D}$) but $\sigma.k$ is not. In particular, U($\mathcal{D}$) of every open and completed derivation is a strongly generated set.

Assume for the moment that we have a suitable procedure which for non-theses of **L** produces open and completed derivations. The following holds:

Lemma 10.4 (Completion) *U($\mathcal{D}$) of open and completed derivation for* **L** *is an L-Hintikka set for* **L**.

Proof We will show the case of **Kt4**. Conditions 1. and 2. from the definition of L-Hintikka set hold because all classical rules were used and the set is open. Clauses 4. and 6. are satisfied because π-formulae were used. For example, if $\pi^P \in \sigma$, then, because it must have been used, either $\pi \in \sigma.i$ for some P-label by application of $(L\pi^P E)$ or this rule was not applied because π was already present in some $\triangleright$-accessible label. Hence there is some $\tau \triangleright \sigma$, not necessarily $\sigma.i$, which satisfies the succedent of clause 6.

For clause 3., assume that $\nu^F \in \sigma$ and $\sigma \triangleright \tau$. Either τ is a direct neighbour of σ (i.e. either τ is a son (i.e. F-label) of σ or σ is a daughter (i.e. P-label) of τ) or there is a chain of intermediate labels creating $\triangleright$-path from σ to τ. In the first case $\nu \in \tau$ either by $(L\nu^F E)$ or by $(LB\nu^F E)$ since ν^F is used. In the second case we proceed by induction on the length of this chain of labels; generation lemma guarantees that all elements of this chain are present in U($\mathcal{D}$). For the basis suppose we have $\sigma \triangleright \theta \triangleright \tau$ and θ is a direct neighbour of both labels. There are three cases: θ a son of σ and τ a son of θ, or θ a daugther of τ and σ a daughter of θ, or σ a daughter of θ and τ a son of θ. The clause is satisfied either by the application of $(L4)$ and $(L\nu^F E)$, or by $(LB4\nu^F)$ and $(LB\nu^F)$, or by $(LB4\nu^F)$ and $(L\nu^F)$. For induction step we assume that a clause is satisfied for every $\triangleright$-path of length n and consider a path of length $n+1$ leading from σ to τ. Let θ be a direct neighbour of σ in this $\triangleright$-path, then by $(L4)$ or $(BL4\nu^F)$ considered ν^F is transferred to θ. $\triangleright$-path from θ to τ is of length n hence it is satisfied by induction hypothesis. Clause 5. is dealt with by the analogical argument. ■

Now, the only element necessary to obtain a completeness proof is a fair proof-search procedure for LAND1. The rest of the Chapter is concerned with this question.

10.1.2 Basic Procedures

Algorithms of proof search in labelled TS for normal basic logics may be found in Fitting, Goré and Massacci (cf. [93, 117, 185]. Fitting's algorithm, modified by Goré, is an example of a breath-first procedure, so – as we explained in Chapter 4 – is not suitable for direct adaptation to ND systems. Massacci applies a depth-first procedure with some strategy of preference for the choice of a formula which is handled first at some stage. Massacci at first applies $(L\pi E)$ to all π-formulae on some branch and then considers other formulae. Although his procedure may be adapted to ND we propose a different solution which may be just added to the procedure from Chapter 4 which was proved to be fair and terminating. It seems that this approach is better if we want to focus on specific problems connected with several classes of logics. Moreover, these algorithms may be generalized further to cover also linear logics, whereas Massacci's procedure seems to be extendible only to the case of **S4.3**.

Although all procedures we introduce below are based on the solution from Chapter 4 it is worth noting that different approaches are possible when we extend it to modal logics. In the setting of labelled systems one may distinguish at least two principal strategies. On syntactical level the difference between them concerns the order of application of rules, but it is better to understand their sense in semantical terms. Briefly, either we prefer a saturation of a world and postpone introduction of new states, or we build a domain of a model first and work out the details of particular worlds later. Clearly, the alternative we have sketched is not exhaustive. One may find algorithms for labelled TS's, where formulae on the branch are dealt with one by one which means that in the attempted model the process of creation of new worlds and their saturation is not separated, e.g. the procedure of Fitting in [93] or of Goré in [117].

The first approach may be called a strategy of "saturation before enlargement (of a model)". In this approach we always first perform a saturation of some σ before we apply π-rule and start a saturation of the next state in the attempted model. Algorithms of this sort for several versions of nonlabelled TS's may be found e.g. in [169, 116]. Let us note that for standard TS in Hintikka format it is the only possible strategy because π-rules correspond to a jump from one state to the new one with a transfer of all admissible formulae. There is no possibility of returning to the "old" state. That's why we must first saturate a set of formulae in some state before we leave it forever.

Clearly, in this approach we encounter branching earlier than in classical case because applications of β-rules usually go before applications of nonbranching modal rules. From the standpoint of complexity of obtained proof-trees it is not very satisfying. In case of a derivation in LND it corresponds to the situation of opening new subderivations as long as we saturate suitable σ, before we apply π-rule and introduce a new label. The number of these subderivations is usually smaller in ND than the number of branching in standard TS because KE (and ND) is exponentially better in this respect (cf. remarks in Chapters 3 and 4) but still we may be forced to excessive branching. We have already discussed disadvantages of uncontrolled branching or increasing the depth of a derivation in classical logic. Moreover, in modal logics – as we will see – it may lead to additional troubles, as was observed by Horrocks [133].

The second strategy works by enlarging a model before a saturation. In such algorithms we first apply all nonbranching rules that increase a length of the current subderivation and after that we start a new subderivation (or introduce branching in TS) if it is required. It means that we may often apply π- and ν-rules before we obtain downward saturation of any world in a model. In semantic terms it corresponds to the addition of new worlds and free walking from one world to another in attempted model. Such a strategy may be realized in labelled systems or in other nonstandard TS or SC which are flexible enough to allow unrestricted jumping from state to state in a model. An example of such a procedure for labelled tableaux may be found in Massacci [186], although he does not require that β-rules must be applied as a last resort. It is also characteristic for his algorithm that $(L\pi E)$ is applied to all π-formulae with the same label in one step, whereas in other procedures only one π-formula is taken under consideration.

It seems that this strategy is a better choice in case of depth-first algorithms. One may find numerous examples where it leads to simpler derivations, especially for theses with many nested modal functors. In this case one may faster obtain $\mathcal{R}$-path to just this state σ, where contradiction arises without necessity of saturation of all $\mathcal{R}$-ancestors of σ. It is not difficult to find examples where we may easily construct a short proof of depth 1 when proceeding in this way, while in consequence of application of the former strategy (i.e. saturation first) we obtain a very long and complex proof of high depth with numerous repetitions of the same sequences of inferences in different subproofs (or branches).

But we have already remarked that possibility of obtaining short proofs does not mean that the space of proof-search is smaller – it may be just the opposite. A procedure may admit many different proofs, including also short ones, but it may be difficult to find them. Usually searching for short proofs is connected with the proper choice of a formulae we deal with in crucial stages, and this is nondeterministic. In fact, there are cases where the freedom of walking from world to world may result in exponentially bounded space of search for problems which are known to be $PSPACE$-complete.[2] In these cases the first strategy is better behaved than the second.

There is one more point connected with peculiarities of ND realization which causes that, despite the flexibility offered by labels, we have chosen the first strategy for later development. On the ground of LND, a demonstration of correctness of algorithms based on the second strategy is much more complicated. Moreover – as we will show – in case of transitive logics an implementation of such algorithms may lead to the loss of completeness. So our constructive proofs will be based on the strategy “saturation before enlargement”, but in case of logics which are neither transitive nor symmetric we will sketch also alternative solution in remarks concerning optimization.

All solutions are based on the procedure SAT(σ) which is a modal modification of SAT(U($\mathcal{D}$)):

SAT(σ) PROCEDURE

Input: a set $\sigma \subseteq$FOR(U($\mathcal{D}$)) of some derivation
Output: L-downward saturated σ

1. **Until** σ is not Ll-saturated, **do**
   apply all static rules to every U-formula.
2. **If** σ is not L-downward saturated, **then**
   choose the first unused β-formula from σ, start a new subderivation (write down SHOW: $\sigma : \beta_i \oplus \sigma : -\beta_i$), and **goto** 1.
   else stop.

Termination and fairness is proved exactly as for SAT(U($\mathcal{D}$)) in Chapter 4. Note that in step 1. the application of all static rules is required which means that if the logic is serial or reflexive, then except rules for α- and β-formulae we must apply also suitable ν-rules ((LD) or (LT)).

Additionally, we introduce as a separate module a procedure for enlarging a model EXT(U($\mathcal{D}$)). It holds for all logics considered in this section and is defined accordingly:

[2] More detailed account of complexity problems of several techniques may be found e.g. in Massacci [186].

EXT(U($\mathcal{D}$)) PROCEDURE

Input: a set U($\mathcal{D}$) with chosen unused π-formula and LAB(U($\mathcal{D}$)).
Output: LAB(U($\mathcal{D}$)) enlarged with one element.

1. Apply $(L\pi E)$ to a formula introducing new label τ to LAB(U($\mathcal{D}$)).
2. Apply ν-rules to all ν^i-formulae in the parent of τ.
3. **Declare** $\sigma := \tau$; stop.

Clearly, in step 2. we apply all ν-rules belonging to LND for respective logic which involve transition from the parent (label) to its fresh child. In practice, in logics without transitivity, it is just $(L\nu E)$ and additionally in Euclidean logics $(L5.4)$ (or $(L5)$ in **K5** and **KD5**), in transitive logics it is also $(L4)$. The remaining ν-rules are either static (and their application is governed by SAT(σ)), or they involve reverse transition, from child-label to its parent. The latter rules will be treated separately in Section 10.4.

10.2 Logics K, D, T

The algorithm defined below yields decision procedure for **K, D, T**.

ALGORITHM 2

Input: A formula φ (a candidate for a thesis of **L**) and the set LAB(U($\mathcal{D}$))={1}.
Output: A proof of φ or a finite open derivation with U($\mathcal{D}$) being L-Hintikka set for **L**, containing $1 : -\varphi$ and finite set LAB(U($\mathcal{D}$)).

0. **Start:** Write down SHOW:1:$\varphi \oplus 1 : -\varphi$ as the beginning of a derivation; declare $\sigma := 1$, LAB(U($\mathcal{D}$)) $= \{1\}$.
1. **Call** SAT(σ) to the current label.
2. **If** σ is inconsistent, **then** apply $(L \perp I)$ and close current subderivation by $[LRED]$;
   - 2.1. **If** the degree of closed subderivation = 1, **then** stop: $\vdash \varphi$ **else** declare $\sigma := \tau$, where τ is a label of S-formula opening closed subderivation, and **goto** step 1.
3. **If** there is an unused π-formula in U($\mathcal{D}$), **then**:
   - 3.1. choose the first one in U($\mathcal{D}$),
   - 3.2. **Call** EXT(U($\mathcal{D}$)),
   - 3.3. **Goto** step 1.

   else stop: φ has no proof.

Let us comment on a few features of algorithm 2.

A. In step 3.1. we mean the first π-formula in a derivation (going from the top), not the first in currently saturated set σ. Otherwise some π-formulae might be never used.

B. Note that conditional subinstruction for step 2. leads not only to the closure of current subderivation (in case of inconsistency) and returning to step 1. (in case it has degree higher than 1), but requires a new declaration of σ which must be saturated. It is necessary because some part of saturated set with the same label is now boxed and we should start its saturation again to guarantee that in case of open derivation, its U($\mathcal{D}$) is indeed an L-Hintikka set. Of course, it may appear that previous S-formula of this boxed subproof is enough to provide a saturated set; in this case the procedure SAT(σ) immediately stops.

C. According to the definition of "used" modal formula, before we apply a rule to some $\sigma : \pi^i$, we must check if there is no $\triangleright$-accessible τ, where π holds. If there is some, then $\sigma : \pi^i$ is immediately signed as used; otherwise we first apply $(L\pi E)$ to it. But here the architecture of ND may cause some problems. A situation of π-formulae is quite similar to a situation of β-formulae (cf. the proof of termination of Algorithm 1 in Chapter 4). Usually, the conclusion of an application of $(L\pi E)$ is not in the same subderivation where the premise but in some nested subderivation of higher degree. It is a consequence of our algorithm which first saturates current σ making a derivation of depth $n \geq 1$, and then takes some unused π-formula from the top of a derivation. Because nested subderivation containing a conclusion of this $(L\pi E)$ application may be later closed, then the premise may become again unused. To take care of this situation we should admit that π-formula is eventually used if the conclusion is in the same subderivation where it is located. In this situation it may be marked by **U** as used, similarly as nonmodal formulae. But if a conclusion is put in a nested subderivation, then a premise should be treated as used as long as this subderivation is open. We may mark this conditional status of being used with **CU**, and delete it everytime a subderivation containing a conclusion is boxed. Such π-formula is retrieved and may be used again. Note however, that the number of calls for EXT(U($\mathcal{D}$)) to one formula $\sigma : \pi^i$ must be finite, since in one subderivation it is applied once and the number of nested subderivations (or nodes in corresponding tree – $\mathcal{T}(\mathcal{D})$) is finite. A formal proof may be provided similarly as for β-formulae in Lemma 4.4.

D. A situation of every ν-formula is analogous but the number of applications of ν-rules to each one must be multiplied by the number of different π-formulae belonging to the same set σ. In practice, we must apply similar principle concerning marking as (conditionally) used with additional condition that they are marked when all π-formulae (from the same σ) were processed and delete marks if at least one such π-formula is retrieved. A bit different case applies if in suitable σ there is no π-formula but there are ν-formulae. In case of **T** these formulae will be used by application of SAT(σ) because (LT) is a static rule. In **K** we must mark them as used because in attempted model they will be satisfied trivially since σ is a "dead end" in a model (no accessible states).

We show the following:

Lemma 10.5 (Termination of algorithm 2) *For any φ, algorithm 2 produces a finite derivation.*

PROOF Note that the performance of every subprocedure is terminating. An addition of modalities has no impact on branching factor or on the length of branches of $\mathcal{T}(\mathcal{D})$ which is a tree representation of our derivation (cf. Chapter 4). Hence both Lemmas 4.3 and 4.4 hold, and by König lemma we obtain a finite tree. The only thing we must check is whether every subderivation (a node of $\mathcal{T}(\mathcal{D})$) is finite in the presence of modalities.

First, $\overline{\{\neg\varphi\}}$ is finite and every σ is its subset; moreover, consistent σ must contain not more than $1/2n$ elements, where n is the cardinality of $\overline{\{\neg\varphi\}}$. So infinite derivation is possible only if we could generate infinite $LAB(U(\mathcal{D}))$ by repeated application of ($L\pi E$). Fitting [93] gives proofs of two facts concerning $LAB(U(\mathcal{D}))$ (cf. also [117] and [185]):

1. For every k, the number of labels σ of this length (i.e. $\mid \sigma \mid= k$) is finite.

2. There is n such that for every σ, $n \geq\mid \sigma \mid$.

Since, by definition of algorithm, repeated application of a rule to some π-formula in the same subderivation is excluded (cf. commentary C. above) then, because $\overline{\{\neg\varphi\}}$ and $LAB(U(\mathcal{D}))$ are finite, every node of $\mathcal{T}(\mathcal{D})$ (i.e. every separate subderivation) is finite. ■

Lemma 10.6 (Fairness of algorithm 2) *For every φ, algorithm 2 yields (in finite time) either a proof or a falsifying model in normal logic* $\mathbf{L} \in \{\mathbf{K}, \mathbf{D}, \mathbf{T}\}$.

PROOF It follows from the preceding lemma that if φ has no proof, then we must obtain a finite derivation, where every formula is marked by **U** or **CU**. But it means that U($\mathcal{D}$) is completed and so it is an L-Hintikka set for **L** containing 1:$\neg\varphi$. We leave a standard proof to the reader. ■

A direct consequence of the above lemma is

Theorem 10.1 (Completeness of LAND1) *If* $\models_L \varphi$, *then* $\vdash_{LAND1-L} \varphi$, *for normal logic* $\mathbf{L} \in \{\mathbf{K}, \mathbf{D}, \mathbf{T}\}$.

Before we go on to other logics it is worth doing some remarks concerning possible optimization of proof search in LAND1.

10.2.1 Optimization

Remark 10.1 (Problems with cut) We have noticed that procedures applying strategy of "saturation before enlargement" usually lead to very redundant and inefficient derivations. In case of modal logics it is not only a problem of excessive branching as such. It is a problem of cut applied on modal formulae. Horrocks [133] pointed out that even analytic cut applied on modal formulae may lead to disastrous effects. Such a cut always gives some π-formula in one branch which leads to the introduction of a new state in a model. Negative impact of this phenomenon may be controlled to some extent. One possible solution is to use modal formulae for cut only as a last resort. In Algorithm 2 we are forced to cut (i.e. to introduce subderivations) only on unused β-formulae. We may add some preference strategy on the set of these formulae:

A. First choose β-formulae having both direct subformulae nonmodal.

B. Next choose from these β-formulae, where one β_i is not modal, and the other (β_j) is ν-formula; introduce a new subderivation on the basis of nonmodal β_i and its complement.

C. Next choose from these β-formulae, where one β_i is not modal, and the other (β_j) is π-formula; introduce a new subderivation on the basis of nonmodal β_i and its complement but additionally select $-\beta_i$ as S-formula and β_i as an assumption.

D. In case where we must open a new subderivation on the basis of modal formulae (the only unused β-formulae consist of two modal components), select π^i as an S-formula and its complement (which must be ν-formula) as an assumption.

In this way we postpone a process of introduction of new states to attempted model, and if the construction of a derivation ends up earlier we may avoid much work. Note that a detailed specification in points C i D, which formula must be used as S-formula and which as an assumption is also a consequence of the idea of adding new labels only eventually.

An application of this strategy not always saves us from excessive enlargement of LAB(U($\mathcal{D}$)). It is possible to use more radical solution however. Recall that we may add to ND an admissible rule [β] being a counterpart of β-rule from tableaux (cf. Chapter 4). In case D above, if at least one of unused β-formulae has ν-formulae as both components we may introduce a new subderivation by [β] instead of [$LRED$]. It means that we will use β_1 and β_2 – both ν-formulae as an S-formula and as an assumption. Note that if we consider such improvement then the order of point C and D should be changed, because introduction of a subderivation on the basis of β-formula containing π-formula finally may lead to the creation of a new label. ♣

Remark 10.2 (Serial Logics) Excessive enlargement of a model may take place especially in case of serial logics. (LD) is a static rule so it leads to increasing of the number of π-formulae in every label quite early. It is easy to observe that every π-formula deduced by (LD), after application of ($L\pi E$) to it and then a series of application of ($L\nu E$) to ν-formulae, produces a set identical with any other set created by means of other π-formula deduced first by (LD) (they have only different labels). For example: in σ we have $\Box\varphi$ and $\Box\psi$ but no π-formula. After application of (LD) we obtain $\Diamond\varphi$ and $\Diamond\psi$ leading in consequence to the application of procedure EXT(U($\mathcal{D}$)) to both, and to the creation of two new labels. But each one, after running of EXT(U($\mathcal{D}$)), consists of φ and ψ.

Possible way of avoiding such a situation is to replace (LD) with (LD'), as in Fitting's system. However, such a solution forces us to substantial modifications of Algorithm 2. Instead, we may keep (LD) and the algorithm but with the addition of some control mechanism. In case of application of ($L\pi E$) to the first of π-formulae in some σ, we should mark as used all π-formulae obtained in σ by the application of (LD). ♣

Remark 10.3 (Alternative Strategy) Let us consider an algorithm which realizes a strategy of "model enlargement before saturation" and which in many cases provides shorter proofs. It is a consequence of late introduction of new subderivations which saves us from repetitions of sequences of inferences in different places of the same proof. In order to find a proof as soon as possible we decided also to put consistency test of U($\mathcal{D}$) rather early in the hierarchy of stages of a procedure. It is common in tableau procedures to put such a test at the end of construction of a branch because it is memory expensive. It is connected with the fact that if we apply simple algorithm with no preference strategies we do not need to check all the branch – just take the next formula and apply suitable rule, whereas consistency test requires the search for complementary formulae through all the branch (cf. [95]). In case of ND we are in many cases forced to preliminary checking at least the part of U($\mathcal{D}$) (searching for the second premise for β-rules, introducing subderivations), so frequent applications of consistency test is not the worst but may speed up a proof by earlier detection of inconsistency. Clearly, in this case the test must not be restricted to literals but search for inconsistency in the whole U($\mathcal{D}$).

A demonstration of fairness and termination of this algorithm needs a little harder work, than for algorithm 2, so we leave it because of the lack of space, and state only a description in pseudo-code.

ALGORITHM 2′

Input and output as for algorithm 2.

0. **Start:** Write down SHOW:1:$\varphi \oplus 1 : -\varphi$ and
   declare $\sigma := 1$, LAB(U($\mathcal{D}$)) $= \{1\}$.
1. **Until** U($\mathcal{D}$) is not Ll-saturated, **do**
   apply static rules to every U-formula.
2. **If** U($\mathcal{D}$) is inconsistent, **then**
   apply $(L \perp I)$ and close the current subderivation by $[LRED]$;
   2.1. **If** the degree of boxed subderivation $= 1$, **then** stop: $\vdash \varphi$
   else goto step 1.
3. **If** there are unused π-formulae in U($\mathcal{D}$), **then**
   select the first one in U($\mathcal{D}$), apply suitable rule and goto step 1.
4. **If** there are unused ν-formulae in U($\mathcal{D}$), **then**
   select the first one in U($\mathcal{D}$), apply suitable rule and goto step 1.
5. **If** U($\mathcal{D}$) is not L-downward saturated, **then**
   select the first unused β-formula in U($\mathcal{D}$), start a new subderivation (writing down SHOW:$\sigma : \beta_i \oplus \sigma : -\beta_i$) and goto step 1.
   else stop: φ has no proof.

10.3 Transitive Logics and Loop-Control

Transitive logics generate additional problems discussed in many places (e.g. [93, 117]). So we omit the details and show some solution for **K4, KD4, S4**.

An argument which was stated in favor of termination of algorithm 2 fails for these logics because the fact 2 used in the proof of Lemma 10.4. does not hold. It is not true that for each derivation there is some n which bounds the length of labels. The key problem is with a rule ($L4$), where we rewrite ν-formula to a new label. ($L4$) does not satisfy subformula property and potentially leads to generation of infinite derivations. Simple example illustrates the point:

1 SHOW:	1:$\Box\Diamond p \to \neg\Diamond\Box p$	
2	1:$\neg(\Box\Diamond p \to \neg\Diamond\Box p)$	*ass.*
3	1:$\Box\Diamond p$	(2, $L\alpha E$)
4	1:$\Diamond\Box p$	(2, $L\alpha E$)
5	1.1:$\Box p$	(4, $L\pi E$)
6	1.1:$\Diamond p$	(3, $L\nu E$)
7	1.1:$\Box\Diamond p$	(3, $L4$)
8	1.1.1:p	(6, $L\pi E$)
9	1.1.1:$\Diamond p$	(7, $L\nu E$)
10	1.1.1:$\Box\Diamond p$	(7, $L4$)
11	1.1.1.1:p	(9, $L\pi E$)
12	1.1.1.1:$\Diamond p$	(10, $L\nu E$)
13	1.1.1.1:$\Box\Diamond p$	(10, $L4$)
e.t.c.		

As long as we are interested in completeness proof and do not need a decision procedure it is not a problem. Goré in [117] describes an algorithm which is not terminating but guarantees that every π-formula will be used at some stage and that every ν-formula will be fulfilled in every accessible label. Our Algorithm 2 is rather different from that of Goré but has the same property – an addition of ($L4$) has no impact on its fairness. But it is interesting to modify a procedure in such a way that decision procedure may be extracted. We may borrow some solution from techniques provided for automated theorem proving (one may consult e.g. Horrocks [134]).

Let us note that the number of consistent subsets of $\overline{\{\neg\varphi\}}$ is finite. ($L4$) does not satisfy subformula property in the strict sense, but it does not add new elements to $\overline{\{\neg\varphi\}}$. Hence on infinite branch we must encounter repeatedly the same sets of formulae but with different labels. The situation in attempted model is the following: we have n-element $\triangleright$-path leading from σ_1 to σ_n, where $\sigma_n \subseteq \sigma_1$. It makes possible an identification of σ_n with σ_1, which yields a cluster containing all labels from σ_1 to σ_{n-1}. It requires an addition of some loop control mechanism to Algorithm 2; let us call it LOOP(σ). The test consists of checking whether for tested σ there is some $\triangleright$-ancestor θ such that $\sigma \subseteq \theta$. If it holds we say that θ blocks σ. In this case we delete σ from LAB(U($\mathcal{D}$)), because in the attempted model we identify it with the blocking label. Modified algorithm looks as follows:

ALGORITHM 3

Input and output as for algorithm 2.

0. **Start:** Write down SHOW:1:$\varphi \oplus 1 : -\varphi$ as the beginning of a derivation; declare $\sigma := 1$, LAB(U($\mathcal{D}$)) $= \{1\}$.
1. **Call** SAT(σ) to the current label.
2. **If** σ is inconsistent, **then**
   apply $(L \perp I)$ and close current subderivation by $[LRED]$;
   2.1. **If** the degree of closed subderivation = 1, **then** stop: $\vdash \varphi$
   else declare $\sigma := \tau$ and goto step 1
   (where τ is a label of S-formula opening closed subderivation).
3. **If** there is unused π-formula in U($\mathcal{D}$), **then**:
   3.1. choose the first one in U($\mathcal{D}$),
   3.2. **Call** LOOP(τ), where τ is a label of the chosen π-formula,
   3.3. **If** τ is blocked by θ, **then** delete τ from LAB(U($\mathcal{D}$))
   and goto step 3.
   else call EXT(U($\mathcal{D}$)) and goto step 1.
   else stop: φ has no proof.

Note the localization of LOOP(σ). We leave a discussion of other possible solutions but one remark is in order. Deletion of τ from LAB(U($\mathcal{D}$)) is not an elimination of suitable part of a derivation but it means that all π-formulae from τ are marked as used. Introduction of loop control has the effect that termination result for Algorithm 2 holds also for the present version. It is easy to check that obtained consistent U($\mathcal{D}$) without formulae belonging to labels deleted in step 3.3, is a Hintikka set for suitable transitive logic. So the results from the preceding subsection hold for Algorithm 3 and we have:

Lemma 10.7 (Termination and fairness of algorithm 3) *For every φ, an algorithm 3 yields in finite time either a proof or a falsifying model in normal logic* $\mathbf{L} \in \{\mathbf{K4}, \mathbf{KD4}, \mathbf{S4}\}$.

Theorem 10.2 (Completeness) *If* $\models_L \varphi$, *then* $\vdash_{LAND1-L} \varphi$, *for normal logic* $\mathbf{L} \in \{\mathbf{K4}, \mathbf{KD4}, \mathbf{S4}\}$.

Remarks concerning optimization from the preceding subsection still apply except Remark 10.3. An algorithm stated there is not possible to adapt for transitive logics because it is incomplete. Here is a simple example: a formula $\Diamond\Box p \wedge \Box\Diamond q \rightarrow \Diamond((r \rightarrow p) \wedge (r \rightarrow q))$ is a thesis of **K4** but the following derivation realized by algorithm $2'$ does not stop.

1 SHOW:	$1{:}\Diamond\Box p \wedge \Box\Diamond q \rightarrow \Diamond((r \rightarrow p) \wedge (r \rightarrow q))$	
2	$1{:}\neg(\Diamond\Box p \wedge \Box\Diamond q \rightarrow \Diamond((r \rightarrow p) \wedge (r \rightarrow q)))$	*ass.*
3	$1{:}\Diamond\Box p \wedge \Box\Diamond q$	(2, $L\alpha E$)
4	$1{:}\neg\Diamond((r \rightarrow p) \wedge (r \rightarrow q))$	(2, $L\alpha E$)
5	$1{:}\Diamond\Box p$	(3, $L\alpha E$)
6	$1{:}\Box\Diamond q$	(3, $L\alpha E$)
7	$1.1{:}\Box p$	(5, $L\pi E$)
8	$1.1{:}\neg((r \rightarrow p) \wedge (r \rightarrow q))$	(4, $L\nu E$)
9	$1.1{:}\neg\Diamond((r \rightarrow p) \wedge (r \rightarrow q))$	(4, $L4$)
10	$1.1{:}\Diamond q$	(6, $L\nu E$)
11	$1.1{:}\Box\Diamond q$	(6, $L4$)
12	$1.1.1{:}q$	(10, $L\pi E$)
13	$1.1.1{:}p$	(7, $L\nu E$)
14	$1.1.1{:}\Box p$	(7, $L4$)
15	$1.1.1{:}\neg((r \rightarrow p) \wedge (r \rightarrow q))$	(9, $L\nu E$)
16	$1.1.1{:}\neg\Diamond((r \rightarrow p) \wedge (r \rightarrow q))$	(9, $L4$)
17	$1.1.1{:}\Diamond q$	(11, $L\nu E$)
18	$1.1.1{:}\Box\Diamond q$	(11, $L4$)
19	$1.1.1.1{:}q$	(17, $L\pi E$)
e.t.c.		

One may easily check that the next six lines would be a repetition of lines 13–18 but with label 1.1.1 replaced by label 1.1.1.1, and – generally – every next seven lines would be a repetition of lines 12–18 but with a new label. It is forced by the strategy of enlargement before saturation which, in the absence of minor premise, forbids the use of β-formula from line 15. If we resign from the algorithm, we may already in line 16 introduce $1.1.1{:}r \rightarrow p$ as S-formula and $1.1.1{:}\neg(r \rightarrow p)$ as an assumption. One application of

$(L\alpha E)$ to this assumption leads to contradiction with line 13 and to closure of this subproof. Then we easily obtain a contradiction in outer derivation ($(L\beta E)$ on line 15 and the new usable-line 16, then $(L\alpha E)$ on the result of this inference), which yields a closure of the main subderivation and a proof. The above example shows that in case of transitive logics we must first saturate old labels before we start the new ones.

The other possible solution is an addition of some test for cycles. LOOP (σ), as stated above is not sufficient because it presupposes saturation. A test proposed by Massacci in [186] (similarly as that of Horrocks [134]) does not require saturation but compared sets of formulae must be identical (requirement of subsumption is too weak if we get rid of saturation) and moreover, if one of them is enlarged by a new formula in due course, then blocking is broken. Such loop control test based on dynamic blocking will be applied later for symmetric and Euclidean logics, because it is necessary in this class of logics. In case of transitive logics we prefer to apply simpler solution described above but one may try to modify alternative algorithm $2'$ by addition of loop test based on dynamic blocking.

10.4 Symmetric and Euclidean Logics

10.4.1 No Transitivity

Logics characterized by symmetric or Euclidean frames generate specific problems for automated proof search. Because transitivity introduces additional complications we first focus on nontransitive logics. An algorithm presented below is defined for logics **KB, DB, B, K5, KD5** and bimodal temporal **Kt**.

First we should note that, in all logics discussed so far, transfer rules allow modal formulae only to be pushed forward, from a label to its extension. The shape of these rules forbids the possibility of moving backward, to states already left. Algorithm 2 (and 3) is well suited to this kind of ν-rules but it fails for symmetric or Euclidean logics because reverse transfer of ν-formulae may be often impossible. Let us illustrate the point. One may easily check that derivation for **KB**-thesis $\Box(p \to \Box q) \to (\Diamond p \to q)$, performed by Algorithm 2, yields (by SAT(σ) on $\sigma = 1$) $\mathrm{U}(\mathcal{D}) = \{1 : \Box(p \to \Box q), 1 : \Diamond p, 1 : \neg q\}$. Instruction 3 with EXT(U$(\mathcal{D})$) will add to U$(\mathcal{D})$ formulae $1.1 : p$ and $1.1 : p \to \Box q$. By SAT(σ) on $\sigma = 1.1$ we additionally get $1.1 : \Box q$. One application of (LB) to this formula would give $1 : q$ and a contradiction, but Algorithm 2 is not able to perform this step. It is impossible because (LB) is a ν-rule and application of ν-rules is connected with preliminary application of $(L\pi E)$ in EXT(U($\mathcal{D}$)), but our U($\mathcal{D}$) does not contain any unused π-formula. Hence, for such logics a combination of applications of ν-rules with $(L\pi E)$ in EXT(U($\mathcal{D}$)) is not enough; we need an independent instruction in algorithm.

But this is still insufficient. An application of (LB) certainly requires the old label to be saturated and tested for consistency once again. Moreover, not only parent-label may change but also some other of her children. Let us consider an example. In U($\mathcal{D}$) we have labelled sets $\sigma, \sigma.1, \sigma.2$ and $\sigma.3$ – all of them saturated. $\sigma.3$ is the current label and contains $\Box\Box\varphi$. Independent instruction for ν-formulae we have postulated above, run an application of (LB) and yields $\sigma : \Box\varphi$. Now σ is a new current label being saturated again. Assume that σ is still consistent; it must contain at least one additional ν-formula (namely $\Box\varphi$) which might be not fulfilled in children of σ, and in the parent of σ, if $\sigma \neq 1$. Even if our additional instruction is formulated sufficiently general and we apply to $\Box\varphi$ not only (LB) but also $(L\nu E)$, then four (or three) labels should be changed in one step: the parent of σ, $\sigma.1$, $\sigma.2$ and $\sigma.3$ (unless some of them already contains φ). Such labels will be called *neighbours* of σ (or i-neighbours in case of temporal logics), and defined as follows:

Definition 10.3 (Neighbours of σ) *τ is a neighbour (i-neighbour) of σ iff:*

- *for* **KB, DB, B**, *$\tau = \sigma.k$ or $\sigma = \tau.k$;*
- *for* **Kt**, *$\tau = \sigma.k$ and is i-label or $\sigma = \tau.k.$ and is j-label, where $i \neq j \in \{F, P\}$*
- *for* **K5, KD5**,

  (a) $\mid \tau \mid > 1$, if $\mid \sigma \mid > 1$,

  (b) $\tau = \sigma.k$, otherwise.

To obtain a fair algorithm we must secure an application of SAT(σ) and consistency test to each neighbour-label which was changed by the application of independent instruction for ν-rules before we may start again with instructions regulating application of rules to modal formulae. In proposed Algorithm 4 we solve this problem by introduction of a new parameter controlling construction of a derivation. A set ν(U($\mathcal{D}$)) $\subseteq$ LAB(U($\mathcal{D}$)) will contain all labels which were changed by additional application of ν-rules (except those provided by running EXT(U($\mathcal{D}$))). In the example above if after application of a ν-rule to $\sigma : \Box\varphi$ we obtain φ in $\sigma.1, \sigma.2, \sigma.3$ (and in the parent of σ, if it exists), then these three or four labels are added to ν(U($\mathcal{D}$)). This set is obviously empty at the start and should be empty at the end, otherwise some labels from this set may be still not saturated. Deletion of elements from ν(U($\mathcal{D}$)) is connected with calling SAT(σ) to them. To guarantee that every element of ν(U($\mathcal{D}$)) will be eventually taken into consideration we order them linearly by the relation $\prec_l$ defined accordingly:

Definition 10.4 (Order on labels) *$\sigma \prec_l \sigma'$ iff, $\mid \sigma \mid < \mid \sigma' \mid$ or $\sigma = \tau i \theta$ and $\sigma' = \tau j \theta'$, where $i < j, \mid \theta \mid = \mid \theta' \mid \geq 0, \mid \tau \mid \geq 1$.*

It is easy to check that $\prec_l$ linearly orders any finite subset of the set of labels which is strongly generated. Introduction of additional elements leads to the next algorithm with input and output as in the preceding ones but with additional parameter – ν(U($\mathcal{D}$)) $\subseteq$ LAB(U($\mathcal{D}$)), which is empty in input and (in case of an open derivation) in output .

ALGORITHM 4

Input and output as for algorithm 3, additionally with ν(U($\mathcal{D}$)) $= \varnothing$.

0. **Start:** Write down SHOW:1:$\varphi \oplus 1 : -\varphi$ as the beginning of a derivation; declare $\sigma := 1$, LAB(U($\mathcal{D}$)) $= \{1\}$, ν(U($\mathcal{D}$)) $= \varnothing$.
1. **Call** SAT(σ) to the current label.
2. **If** σ is inconsistent, **then**
   apply $(L \perp I)$ and close a current subderivation by $[LRED]$;
   2.1. **If** the degree of closed subderivation $= 1$, **then** stop: $\vdash \varphi$
   else declare $\sigma := \tau$, and goto step 1
   (where τ is a label of S-formula opening closed subderivation)
3. **Until** there are some unused ν-formulae in σ, **do**
   apply suitable ν-rules with respect to each neighbour of σ
   and add this neighbour to ν(U($\mathcal{D}$)).
4. **If** ν(U($\mathcal{D}$)) $\neq \varnothing$, **then**
   declare $\sigma := \tau$ and ν(U($\mathcal{D}$)) $:= \nu$(U($\mathcal{D}$)) $- \{\tau\}$,
   where τ is $\prec_l$-first label in ν(U($\mathcal{D}$)) and goto step 1.
5. **If** there are unused π-formulae in U($\mathcal{D}$), **then**:
   5.1. choose the first one in U($\mathcal{D}$),
   5.2. **call** EXT(U($\mathcal{D}$)) and goto step 1.
   else stop: φ has no proof.

New elements have no impact on termination and fairness proof for Algorithm 2. Let's note that the localization of instruction 3. and 4. guarantees that enlargement of a model with new states is not possible unless we use all ν-formulae in all accessible states and then make a saturation and consistency test in all of them. Finite number of neighbours of every label and their systematic elimination from $\nu(\mathrm{U}(\mathcal{D}))$ is sufficient for termination. So we obtain:

Lemma 10.8 (Termination and fairness of algorithm 4) *For every φ, algorithm 4 yields in finite time either a proof or a falsifying model in normal logic $\mathbf{L} \in \{\mathbf{KB}, \mathbf{KDB}, \mathbf{B}, \mathbf{K5}, \mathbf{KD5}, \mathbf{Kt}\}$.*

Theorem 10.3 (Completeness) *If $\models_L \varphi$, then $\vdash_{LAND1-L} \varphi$, for normal logic $\mathbf{L} \in \{\mathbf{KB}, \mathbf{KDB}, \mathbf{B}, \mathbf{K5}, \mathbf{KD5}, \mathbf{Kt}\}$.*

10.4.2 Transitive Symmetric or Euclidean Logics

We already know that introduction of transitivity leads, due to the form of ($L4$) (as well as ($LB4$)), to possibility of generation of infinite derivations. Algorithm 4 must be modified analogously as Algorithm 2, by addition of some loop-control device. So, for logics **KB4, DB4, K54, K54D, S5, Kt4** we obtain the following algorithm.

ALGORITHM 5

Input and output as for algorithm 4.

0. **Start:** Write down SHOW:1:$\varphi \oplus 1 : -\varphi$ as the beginning of a derivation; declare $\sigma := 1$, LAB(U($\mathcal{D}$)) = {1}, $\nu(\mathrm{U}(\mathcal{D})) = \varnothing$.
1. **Call** SAT(σ) to the current label.
2. **If** σ is inconsistent, **then**
   apply ($L \perp E$) and close current subderivation by [$LRED$];
   2.1. **If** the degree of closed subderivation = 1, **then** stop: $\vdash \varphi$
   else declare $\sigma := \tau$, and goto step 1
   (where τ is a label of S-formula opening closed subderivation)
3. **Until** there are some unused ν-formulae in σ, **do**
   apply suitable ν-rules with respect to each neighbour of σ
   and add this neighbour to $\nu(\mathrm{U}(\mathcal{D}))$.
4. **If** $\nu(\mathrm{U}(\mathcal{D})) \neq \varnothing$, **then**
   declare $\sigma := \tau$ and $\nu(\mathrm{U}(\mathcal{D})) := \nu(\mathrm{U}(\mathcal{D})) - \{\tau\}$,
   where τ is $\prec_l$-first label in $\nu(\mathrm{U}(\mathcal{D}))$ and goto step 1.

5. **If** there are unused π-formulae in U($\mathcal{D}$), **then**:
   5.1. choose the first one in U($\mathcal{D}$),
   5.2. **Call** LOOP(τ), where τ is a label of chosen π-formula,
   5.3. **If** τ is blocked by θ, **then**
   delete τ from LAB(U($\mathcal{D}$)) and goto step 5.
   else call EXT(U($\mathcal{D}$)) and goto step 1.
   else stop: φ has no proof.

Identical formulation of Conditions 5.2 and 5.3 as in Algorithm 3 does not mean that performed operations are identical. In case of transitive symmetric (and Euclidean) logics loop-control is more complicated. First, it is not sufficient that the set of formulae with label σ is a subset of some earlier set labelled by θ – they must be equal in order for σ to be blocked by θ. Second, because of the possibility of enlargement of some (blocked or blocking) set in later stages of proof construction (due to application of (LB) and similar rules) we must admit that blocking will be broken. So, in contrast to LOOP(σ) for transitive but not symmetric logics, where label is blocked once and for all, here we assume that blocking is provisory and is broken if one of the blocking set changes.

This solution is based on the technique of "dynamic blocking", devised by Horrocks [134] for implementation of tableau system for some versions of description logics. The latter may be translated for multimodal normal logics, so this technique may be also applied to them. An adaptation of dynamic blocking to our algorithm is even simpler than in Horrocks' system since the parameter ν(U($\mathcal{D}$)) may be directly applied to loop-control. We should implement a principle that blocking of σ by θ is possible only if neither label belongs to ν(U($\mathcal{D}$)); otherwise the blocking is broken. In contrast to Horrocks' system, our version does not require a distinction between direct (equality of parent-label and its child) and indirect blocking (identity of a set with some of its ancestors). It is possible because in our algorithm the process of saturation of labels and application of ν-rules is made before the selection of the next unused π-formula and LOOP-test, while in Horrocks' procedure the test is performed on the occasion of application of other rules.

Thanks to the application of dynamic blocking Algorithm 5 is not only fair for respective logics but also terminates. So we also obtain:

Lemma 10.9 (Termination and fairness of algorithm 5) *For every φ, algorithm 5 yields in finite time either a proof or a falsifying model in normal logic* $\mathbf{L} \in \{\mathbf{KB4}, \mathbf{KDB4}, \mathbf{K45}, \mathbf{KD45}, \mathbf{S5}, \mathbf{Kt4}\}$.

Theorem 10.4 (Completeness) *If* $\models_L \varphi$, *then* $\vdash_{LAND1-L} \varphi$, *for normal logic* $\mathbf{L} \in \{\mathbf{KB4}, \mathbf{KDB4}, \mathbf{K45}, \mathbf{KD45}, \mathbf{S5}, \mathbf{Kt4}\}$.

10.5 Linear Logics

In the preceding Chapter we have shown that LAND1 for basic linear logics is adequate. However, the procedure of proof search involved in the proof is impractical. First, extracted falsifying models are usually infinite, hence the proof does not provide a decision procedure although all these logics are decidable (and $coNP$-complete) which was established by Ono and Nakamura already in [200]. Second, due to necessity of maximalization of every label we are forced to make a lot of inessential inferences including many applications of $[LRED]$. As for the first problem it may be easily solved by providing some mechanism of loop control discussed in this Chapter. The second one is more serious. Despite the fact that it is a relative maximalization with respect to $\overline{\{\neg\varphi\}}$ (where φ is a formula to be proved) and that the process is finite, we obtain too much unnecessary subderivations. We have already discussed the destructive effect of using cut (i.e. $[LRED]$ in LND) in modal logics, even in analytic form (cf. Remark 10.1) hence the problem of finding procedures minimizing the number of its applications is vital. In [151] an alternative approach was sketched which will be the basis of the present development. It requires only L-downward saturation of labels and may be naturally based on the technical apparatus used so far in this Chapter.

Some refinements are needed however. In particular, accessibility of labelled sets is defined not in the manner specified in Section 10.1 but rather as in Chapter 9:

Definition 10.5 (Relation of accessibility) τ is accessible from σ iff:
(a) if $\nu^F \in \sigma$, then $\nu^F \in \tau$ i $\nu \in \tau$,
(b) if $\nu^P \in \tau$, then $\nu^P \in \sigma$ i $\nu \in \sigma$.

The difference is that now we do not require sets to be maximal but only downward saturated.

Similarly as in Chapter 9 we build a graph of accessibility of labels keeping under control the linearity of attempted model. But there are some differences with respect to the definition of a chain stated there. Now the chain is constructed not from single labels but rather from clusters of labels which is necessary to secure a finite model.

Definition 10.6 (Finite chain) Let **C** be a finite collection of clusters of labelled sets of formulae such that every σ from every cluster in **C** is L-downward saturated in $\overline{\{\neg\varphi\}}$, **C** is a chain in $\overline{\{\neg\varphi\}}$ iff for any different τ and θ: θ is accessible from τ or τ is accessible from θ.

C is fulfilled iff every π-formula in every label of every cluster in **C** is used.

Proof that every fulfilled chain yields linear model does not require any extra operations except those from Chapter 9. Transitivity is secured by definition of accessibility relation and linearity by definition of **C**. From the standpoint of intuition one may have some doubts – parallel time points are admissible in such a model. But it is innesential from the technical point of view. Let us recall that every chain of clusters may be transformed by operation of bulldozing, yielding a chain of single points but infinite; and this is just what we want to avoid (cf. [112]).

10.5.1 Finite Chains

The problem consists in providing such an algorithm of proof search which, in case of open derivation, provides U($\mathcal{D}$) being finite chain. We must first of all modify the procedure EXT(U($\mathcal{D}$)) because it additionaly selects a place in **C** for newly introduced label. Assume that σ is a label of π-formula which was chosen as the first (from the top of a derivation) unused in U($\mathcal{D}$). In case of temporal logic this π-formula may be $\sigma : \pi^F$ or $\sigma : \pi^P$. Because of that we will define it in the parallel fashion; everything connected with $\sigma : \pi^F$ will be put in the main text and everything concerning $\sigma : \pi^P$ will be put in [ ]. In case of monomodal logics (**K4.3, KD4.3, S4.3**) it is enough to leave the part of procedure contained in [ ]. τ will be used for immediate successor [ancestor] of σ in **C**. For all linear logics which are not reflexive a suitable procedure is defined as follows:

PROCEDURE EXT(U($\mathcal{D}$))'

Input: U($\mathcal{D}$) with chosen unused π-formula and **C**

Output: respective π-formula marked as used.

1.**If** σ does not belong to the last [first] cluster in **C**, **then**:
  1.1.**If** $\tau : -\pi \in$U($\mathcal{D}$), **then**:
    1.1.1.**If** $\tau : -(\pi^F) \in$U($\mathcal{D}$) [$\tau : -(\pi^P) \in$U($\mathcal{D}$)], **then**:
      Apply ($L\pi E$) and **mark** $\sigma : \pi^F$ [$\sigma : \pi^P$] as used;
      Add new $\sigma.i$ to **C** between σ and τ;

Apply $(L\nu^F E)$ $[(L\nu^P E)]$ to every ν-formula in σ wrt. $\sigma.i$;
Declare $\sigma := \sigma.i$, **stop**.
1.1.2.**Else add** "SHOW:$\tau : -(\pi^F) \oplus \tau : \pi^F$"
["SHOW:$\tau : -(\pi^P) \oplus \tau : \pi^P$"] to $\mathcal{D}$;
Mark $\sigma : \pi^F$ $[\sigma : \pi^P]$ as used;
Declare $\sigma := \tau$, **stop**.
1.2.**Else add** "SHOW:$\tau : -\pi \oplus \tau : \pi$" to $\mathcal{D}$;
Mark $\sigma : \pi^F$ $[\sigma : \pi^P]$ as used;
Declare $\sigma := \tau$, **stop**.
2.**Else apply** $(L\pi^F E)$ $[(L\pi^P E)]$ and mark $\sigma : \pi^F$ $[\sigma : \pi^P]$ as used;
Add new $\sigma.i$ to the end [beginning] of **C**;
Apply $(L\nu^F E)$ $[(L\nu^P E)]$ to every ν-formula in σ wrt. $\sigma.i$;
Declare $\sigma := \sigma.i$, **stop**.

Clearly, the above procedure has still some computational drawbacks. Since conditions for application of instruction 1.1 and 1.1.1 require checking U($\mathcal{D}$) to see if some formulae are (not) in it, in the worst case we have to search all U($\mathcal{D}$) twice. In practice it would be good to amalgamate both instructions in order to get more efficient version. In case of reflexive logics instruction 1.1.1. is simply superfluous and the following version is sufficient:

1.**If** σ does not belong to the last [first] cluster in **C**, **then**:
1.1.**If** $\tau : -(\pi^F) \in$U($\mathcal{D}$) $[\tau : -(\pi^P) \in$U($\mathcal{D}$)], **then**:
Apply $(L\pi E)$ and **mark** $\sigma : \pi^F$ $[\sigma : \pi^P]$ as used;
Add new $\sigma.i$ to **C** between σ and τ;
Apply $(L\nu^F E)$ $[(L\nu^P E)]$ to every ν-formula in σ wrt. $\sigma.i$;
Declare $\sigma := \sigma.i$, **stop**.
1.2.**Else add** "SHOW:$\tau : -\pi \oplus \tau : \pi$" to $\mathcal{D}$;
Mark $\sigma : \pi^F$ $[\sigma : \pi^P]$ as used;
Declare $\sigma := \tau$, **stop**.
2.**Else apply** $(L\pi^F E)$ $[(L\pi^P E)]$ and mark $\sigma : \pi^F$ $[\sigma : \pi^P]$ as used;
Add new $\sigma.i$ to the end [beginning] of **C**;
Apply $(L\nu^F E)$ $[(L\nu^P E)]$ to every ν-formula in σ wrt. $\sigma.i$;
Declare $\sigma := \sigma.i$, **stop**.

It is obvious that every performance of EXT(U($\mathcal{D}$))' is finite exactly as in case of the basic variant (for nonlinear logics). Higher complexity of an algorithm follows from the necessity of providing linearity of attempted model. It is guaranteed that every π-formula in U($\mathcal{D}$) must be eventually used but it may takes different forms. Suitable π-rule is appliedonly if σ

is in one of the outermost clusters or if its successor [ancestor] satisfies the antecedent of condition 1.1 and 1.1.1. In the latter case a new label $(\sigma.i)$ is inserted before [after] τ. It is easy to observe that in U($\mathcal{D}$) we obtain then $\tau : -\pi^F[\tau : -\pi^P], \tau : -\pi, \sigma.i : \pi$, hence we have a complete set of premises needed for application of $(3FF)$ $[(3PP)]$ to every ν^F-formula [ν^P-formula] in $\sigma.i$ with respect to τ, and $(3FP)$ $[(3PF)]$ to every ν^P-formula [ν^F-formula] in τ with respect to $\sigma.i$. It saves accessibility of τ from $\sigma.i$ [accessibility of $\sigma.i$ from τ] in updated **C**. In the remaining cases our π^i is used by addition of π to τ (in case of unsatisfiability of the antecedent of the second instruction, i.e. in stage 1.1. leading to 1.2), or – if it leads to inconsistency (when the antecedent of condition 1.1 is satisfied but that of 1.1.1 is not) – π^i is moved to the next [earlier] label (point 1.1.2.). Since all π-formulae are visited one by one going from the top, so it is certain that $\tau : \pi^i$ will become eventually the starting point for calling EXT(U($\mathcal{D}$)). Let us notice that **C** is enlarged only by definitive use of π^i; its moving to some successor [ancestor] by application of 1.1.2. does not make **C** bigger. Hence – by the finiteness of **C** – π^i must be eventually used. In the worst case it appears in the last [first] label in **C** and is covered by condition 1. An application of π-rule (or satisfaction of π^i by 1.2) yields satisfaction of all occurences of this π^i in all earlier [later] labels in **C**, which justifies marking them all as used. It remains to define an algorithm in such a way as to make sure that all labels in open and finite U($\mathcal{D}$) are saturated and accessible to each other.

10.5.2 Proof Search Algorithm

Now we state an Algorithm 6 for bimodal temporal logics. Input and output is as for Algorithm 3, but we additionaly must take into account **C**. Complexity of the algorithm follows from the necessity of saturation of every label before EXT(U($\mathcal{D}$)) is applied, because U($\mathcal{D}$) must be kept as a chain all the time. We will show that before application of instruction 5., which leads to elimination of successive π-formulae and possible enlargement of **C**, all labels must be downward saturated. It guarantees that if we stop an algorithm, because of the lack of unused π-formulae, we will obtain U($\mathcal{D}$) which is a fulfilled chain.

ALGORITHM 6

Input and output as for algorithm 5.

0. **Start:** Write down SHOW:1:$\varphi \oplus 1 : -\varphi$ as the beginning of a derivation; declare $\sigma := 1$, LAB(U($\mathcal{D}$)) $= \{1\}$, $\mathbf{C} = \langle\{1\}\rangle$.
1. **Call** procedure SAT(σ) to the current label.
2. **If** σ is inconsistent, **then**
   apply $(L\bot I)$ and close current subderivation by $[LRED]$;
   2.1. **If** the degree of closed derivation $= 1$, **then Stop**: $\vdash \varphi$
   Else declare $\sigma := \tau$ and goto 1.
   (where τ is a label of S-formula of closed subderivation)
3. **If** σ has immediate successor τ and unused ν^F-formulae, **then**:
   3.1. apply suitable ν-rules with respect to τ;
   3.2. declare $\sigma := \tau$, where τ is the successor of σ, and goto 1.
   Else
   4. **If** σ has immediate ancestor τ, **then**:
      4.1. **If** there are unused ν^P-formulae, **then**:
         4.1.1. apply suitable ν-rules with respect to τ;
         4.1.2. declare $\sigma := \tau$, where τ is the ancestor of σ, and goto 1.
         Else declare $\sigma := \tau$, where τ is the ancestor of σ, and goto 4.
      Else
      5. **If** in U($\mathcal{D}$) there is unsued π-formula, **then**:
         5.1. Select the first one in U($\mathcal{D}$),
         5.2. **Call** LOOP(τ), where τ is a label of selected π-formula,
         5.3. **If** τ is blocked by θ, **then**
            declare a cluster $\{\tau = \theta\}$ and goto 5.
            Else Call EXT(U($\mathcal{D}$)) and goto 1.
         Else Stop: φ has no proof

Let us note that condition 3 forces us to check for every saturated and consistent σ whether condition (a) of accessibility (cf. definition 10.6.) is satisfied with respect to its successor in **C**. If it is not satisfied then we must apply suitable rules to reach the effect. After this operation, the successor of σ becomes the current label and it is being checked. These steps are repeated until we reach the last label in **C** or stop earlier because of the lack of suitable ν^F-formulae. Then condition 4. forces us to check whether the predecessor of the current label satisfies condition (b) of accessibility. In an analogous way we walk through the chain, checking all labels and saturating them if needed, but this time we are going in the reverse direction. Let us note that this travel to the beginning is interrupted every time we make a saturation of some label, because in this case again instruction 3. is repeated which may change the direction of our walk if we find new unused ν^F-formulae in currently processed label. In any case we are granted that this travel up and down the chain must end since **C** is finite and every σ is finite too. Only after reaching the first label in **C** the instruction 5. is called for. So after every running of EXT(U($\mathcal{D}$)) we make systematic walk on all elements of **C** – from current label to the last, and back again – which makes sure that:

(a) every label in **C** is L-downward saturated,
(b) every label is accessible from all its predecessors in **C**.

Summing up, the application of EXT(U($\mathcal{D}$)) takes place only if U($\mathcal{D}$) is a chain, and the procedure stops only when all π-formulae are marked as used, hence U($\mathcal{D}$) must be a fulfilled chain.

In case of monomodal linear logics the algorithm may be simplified, since we need only the walk in one direction (to the end of a chain). Hence the part of an algorithm covering instructions starting with 4. is superfluous. Also the procedure of loop control requires only inclusion of current label in blocking label, and constructed clusters have stable character.[3]

10.5.3 Worst Case Analysis

Notice that in our algorithm [$LRED$] is applied only for missing β-premises (needed for downward saturation) and for missing premises for 3-rules (to secure linearity). Hence the use of (analytic) cut is restricted to necessary cases, instead of blowing up each label until we get a maximal set as was the case in the proof described in the preceding Chapter or in the proof based on the technique of mosaics and applied in [183]. But we cannot eliminate all the applications of cut which may lead to suspection that other approaches are better. We have shown that our rules may considerably shorten proofs but we should also consider whether in the worst case our systematic procedure does not lead to proof-search of higher complexity than in other systems proposed for logics of linear frames. Fortunatelly, as far as the branching due to the specific rules for linearity is concerned, we achieve essentially the same result.

In order to show that at worst our procedure explores the proof-search space which is of the same complexity as in other systems, it is better to rephrase its action in terms of tableaux with (analytic) cut. We restrict a

[3] Cf. respective considerations from the preceding section.

comparison of our procedure with others, to the case of Rescher/Urquhart system [231]. It is sufficient since we have established in Chapter 9 that the system of Rescher/Urquhart eksplores the same proof-search space as others. Also it is sufficient to pay attention only to these applications of cut which are connected with the use of 3-rules. We will show that satisfaction of any π-formula does not require creation of more new branches than in the case of Rescher/Urquhart system.

Suppose we have an open branch $\mathcal{B}$ with some unfulfilled $\sigma : F\varphi$ and let $\tau_1, ..., \tau_n$ be the list of successors of σ in the chain corresponding to the current open branch. According to the procedure EXT(U($\mathcal{D}$)) (step 1.2), first we make a cut with $\tau_1 : \varphi$ (sub-branch $\mathcal{B}1$) and $\tau_1 : -\varphi$ (sub-branch $\mathcal{B}2$); $\mathcal{B}1$ finishes, since $F\varphi$ is satisfied on it in τ_1. On $\mathcal{B}2$ by step 1.1.2. we make a cut with $\tau_1 : F\varphi$ (sub-branch $\mathcal{B}2.1$) and $\tau_1 : \neg F\varphi$ (sub-branch $\mathcal{B}2.2$). Now, on $\mathcal{B}2.2$ we apply $(L\pi E)$ to $\sigma : F\varphi$ thus introducing $\sigma.i : \varphi$, where $\sigma.i$ is new. $\mathcal{B}2.2$ corresponds to a chain with $\sigma.i$ inserted between σ and τ_1 since for every $\sigma.i : \nu^F$ (or $\tau_1 : \nu^P$) we have a sufficient set of premises (namely $\tau_1 : -\varphi$, $\tau_1 : \neg F\varphi$, $\sigma.i : \varphi$) to apply 3-rules and guarantee accessibility of τ_1 from σ. So $\mathcal{B}2.2$ also finishes in this respect. On $\mathcal{B}2.1$ we have moved $F\varphi$ from σ to τ_1, so we have to repeat this procedure, first doing cut on $\tau_2 : \varphi$ and $\tau_2 : -\varphi$ and then (on the latter branch) with $\tau_2 : F\varphi$ and $\tau_2 : \neg F\varphi$. One can easily check that the total number of new branches potentially created by cuts to all τ_i on $\mathcal{B}$ is $2n + 1$ – the exact number of possible places where our $F\varphi$ may be satisfied.

Hence, our procedure gives no redundant branches – in the worst case it works as the system of Rescher and Urquhart. The difference is that in our procedure we introduce $2n+1$ branches through successive binary branching due to cut, whereas in their system we immediately create $2n+1$ branches by a special rule. Needless to say that in our approach on every stage we may have an opportunity to decrease the number of further branches if required formulae are already present in suitable τ_i (i.e. if conditions 1., 1.1., 1.1.1. are satisfied).

Cosequently, the system we have proposed is not worse, and in practice it tends to behave better, than other systems. In particular, let us note that the algorithm we have stated in this section was not realised in examples of short proofs from Chapter 9. But we leave the question of optimization of proof search in our system for linear logics for another day.

Chapter 12

Proof Methods for MHL

The final Chapter covers proof theory of hybrid logics. In contrast to the rest of the book, where ND performs a priviliged position, we have tried to present almost all deductive systems constructed so far for hybrid logics and describe their most interesting features. It follows from the author's conviction that on the field of investigation on proof methods for modal logics, the application of hybrid languages instead of standard modal languages may offer a real breakthrough, so careful analysis is vital.

In the first Section we make an introductory division of existing systems on some types. In particular, we distinguish sat-calculi operating on sat-formulae only, as the most popular approach in every group of systems. In Section 12.2 we describe three variants of sequent calculi: ordinary SC due to Seligman, sat-calculus of Blackburn, and nonstandard calculus of Demri. The next Section presents tableau systems: labelled (mixed) calculus of Tzakova with their refinements due to Bolander, Braüner and Blackburn, and sat-calculus of Blackburn. Then we present ND systems – a standard one due to Indrzejczak, and sat-calculus of Braüner. Finally, we introduce two sat-calculi defined on clauses and based on resolution, namely HyLoRes due to Areces, and HRND (hybrid RND) being a generalization of RND system. Other types of systems like refutation systems, connection calculi, Davis-Putnam method or goal oriented deduction systems, although applied in nonclassical logics, and in standard modal logic in particular, were not devised for hybrid logics so far. In each case we will present the basic system and its main features. In particular, we will focus on the problem of a uniform extension of the basic system to stronger languages and logics.

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*, 398
Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_12,

12.1 Kinds of Formalizations of MHL

Before we describe and compare the existing formalizations of hybrid logics some remarks on the general features of proof systems are in order.

We have already mentioned (in Section 7.5) that for ordinary modal logic a lot of nonstandard systems were devised, particularly in the area of sequent calculi. In hybrid logics all these additional techniques are usually not needed due to the richer abilities of language. Hence proof systems for hybrid logics are usually based on the standard solutions for ordinary modal logics but, as we shall see, they often improve them in many respects.

On the other hand, all these systems may be counted as belonging to one of the nonstandard approaches in constructing proof systems, based on the use of labels (prefixes). As we have pointed out in Chapter 8, in the wide sense we may distinguish at least three groups of labelled systems. All systems presented in Chapters 8, 9, and 10 are based on the external form of representation of states in a model, whereas hybrid languages represent the internalised labelling. Thus, if we use the term labelled deduction in its most general form, then all existing proof systems for hybrid logics belong to this category.

Internalised labelled systems are the most popular solution for hybrid logics, but we may find also tableau systems with mixed labelling (nominals and external labels) – the earliest one due to Tzakova. But even in the group of internalized systems, where essentially we have a situation of application of standard proof systems to hybrid logics, it is handy to distinguish an additional group. There is a group of systems which do not use external labels but where the rules are defined only on sat-formulae or data structures (like sequents, clauses e.t.c.) built up only from sat-formulae. Systems of this sort are strongly similar to calculi with external (strong) labelling. Although such systems are naturally limited to logics in languages with sat-operators, within this group of languages they have sufficient generality. It follows from the admissibility of the rule:

$(NAME) \;\; \vdash @_i\varphi \;/\; \vdash \varphi$ if $i \notin \varphi$

So, if we want to prove a thesis φ which is not a sat-formula we must try to prove $@_i\varphi$ with $i \notin \varphi$. If we succeed, then, by $(NAME)$ it holds also for φ alone. On the other hand, if we fail, then φ is false in $V(i)$ (provided we deal with some universal system, like TS, where falsification is possible).

The reason to distinguish sat-calculi as a group of its own lies in the fact that they form the most numerous group of proof systems for hybrid logics.

Hence, finally we divide proof systems for hybrid logics on three groups:

1. Standard calculi (standard systems with additional rules for hybrid constants): Seligman's SC and ND-system of Indrzejczak.
2. Sat-calculi (systems defined on sat-formulae): Blackburn's TS, Demri's SC,[1] Braüner's ND-system, Areces' HyLoRes resolution system, HRND-system (RND for hybrid logics) of Indrzejczak.
3. Mixed calculi (systems with external labels): SC of Seligman and TS's of Tzakova and of Bolander.

12.2 Sequent Calculi

We start our presentation of proof systems with sequent calculi not only because they seem to be the most important proof systems applied in proof theory but also because of chronological reasons. In fact, the first non-axiomatic systems constructed for hybrid logics were of this type. There were several versions od SC constructed by Seligman in the early 1990s, for situation theory (see [248, 249]). These systems deal with languages without modalities, so we do not describe them in detail, focusing rather on the system from [251] which extends the earlier results and contains a formalization of strong modal hybrid logic. But this is not the only SC for hybrid logics.

In what follows, we will describe three calculi for MHL. Except Seligman's ordinary sequent calculus, we present two nonstandard ones. One of them, due to Blackburn [32], is an example of a sat-calculus, constructed rather indirectly, by the transformation of suitable tableau sat-system. Although this SC uses exclusively sat-formulae, it is rather an extension not a substantial modification of standard Gentzen approach. The second proposal, due to Demri [78], shows more serious departure from ordinary SC since it is based on sequent version of KE-system.

From the variety of other nonstandard sequent calculi applied for modal logics like, e.g. hypersequent calculus, only display calculus was used by Demri and Goré [79], to formalize tense hybrid logic. We do not present this SC because it would require a prior presentation of principles of display calculi, and there is no space for that.

[1] In fact this system is not a sat-calculus in the strict sense. There are no even sat-operators in the language! But rules are defined only on formulae of the shape $i \rightarrow \varphi$, which naturally put this calculus in this group.

12.2.1 Seligman's SC

We start with the sequent calculus of Seligman complete for the basic hybrid logic in $\mathbf{L_{H@}}$ and some stronger languages. It this case we have simply an extension of ordinary SC with additional rules. So, the first group of rules is like in the standard SC as defined in Section 3.1.1 with small differences. Seligman's rules are defined on ordinary Gentzen sequents $\Gamma \Rightarrow \Delta$, where both Γ, Δ are not sets but finite lists of formulae. In fact, his actual set of rules is slightly different from original Gentzen's set either, since there is no ordinary Gentzen's structural rules of contraction, permutation and weakening. Instead, he uses (AX) in generalized form (with side formulae) and a general rule (S) which captures the effect of contraction and permutation.

The second group of rules is dealing with nominals.

Nominal rules

$$(@I\Rightarrow)\ \frac{i, \varphi, \Gamma \Rightarrow \Delta}{i,\ @_i\varphi,\ \Gamma \Rightarrow \Delta} \qquad (\Rightarrow @I)\ \frac{i, \Gamma \Rightarrow \Delta, \varphi}{i, \Gamma \Rightarrow \Delta,\ @_i\varphi}$$

$$(@E\Rightarrow)\ \frac{i,\ @_i\varphi, \Gamma \Rightarrow \Delta}{i, \varphi,\ \Gamma \Rightarrow \Delta} \qquad (\Rightarrow @E)\ \frac{i, \Gamma \Rightarrow \Delta,\ @_i\varphi}{i, \Gamma \Rightarrow \Delta, \varphi}$$

$$(N_1)^1\ \frac{i, j, \Gamma[i] \Rightarrow \Delta[i]}{i, j, \Gamma[j] \Rightarrow \Delta[j]} \qquad (TERM)^2\ \frac{i, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \qquad (S\text{-}NAME)^3\ \frac{i, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

Side conditions:

1. where $\Gamma[i]$ means that i occur in Γ and $\Gamma[j]$ is the result of replacement of j for i in Γ
2. where all elements of Γ, Δ are sat-formulae
3. where i does not occur in Γ, Δ.

(N_1) is a kind of substitution rule; we substitute j for i at once through all the sequent.

The last group of rules is defined for modal constants.

Modal rules

$$(H\Diamond\Rightarrow)^1\ \frac{\Diamond i,\ @_i\varphi, \Gamma \Rightarrow \Delta}{\Diamond\varphi, \Gamma \Rightarrow \Delta} \qquad (H \Rightarrow\Diamond)\ \frac{\Gamma \Rightarrow \Delta, \Diamond i \quad \Gamma \Rightarrow \Delta, @_i\varphi}{\Gamma \Rightarrow \Delta, \Diamond\varphi}$$

$$(H\Box\Rightarrow)\ \frac{\Gamma \Rightarrow \Delta, \Diamond i \quad @_i\varphi, \Gamma \Rightarrow \Delta}{\Box\varphi, \Gamma \Rightarrow \Delta} \qquad (H \Rightarrow\Box)^1\ \frac{\Diamond i, \Gamma \Rightarrow \Delta, @_i\varphi}{\Gamma \Rightarrow \Delta, \Box\varphi}$$

Side condition: where i does not occur in Γ, Δ and φ.

This time we've added rules for $\Box$ that are not present in [251], but are easy to obtain.

Proofs in the system are defined in standard way, as trees of sequents, constructed by means of the rules, with axioms as leaves and deduced sequents as roots. Clearly, proof search is performed in an upside-down manner; we start with the root-sequent and systematically add above sequents-premises of suitable rules. Below we display an example of a proof (applications of (S) ignored).

$$\begin{array}{ll}
 & \Diamond j, i \Rightarrow @_j p, \Diamond j \qquad\qquad @_j p, \Diamond j, i \Rightarrow @_j p \\ \hline
(H\Box \Rightarrow) & \Box p, \Diamond j, i \Rightarrow @_j p \\ \hline
(@I \Rightarrow) & i, @_i \Box p, \Diamond j \Rightarrow @_j p \\ \hline
(@I \Rightarrow) & i, @_i \Box p, @_i \Diamond j \Rightarrow @_j p \\ \hline
(TERM) & @_i \Box p, @_i \Diamond j \Rightarrow @_j p \\ \hline
(\wedge \Rightarrow) & @_i \Box p \wedge @_i \Diamond j \Rightarrow @_j p \\ \hline
(\Rightarrow \rightarrow) & \Rightarrow @_i \Box p \wedge @_i \Diamond j \rightarrow @_j p
\end{array}$$

The notions of the relation of deducibility between sequents, derivable and admissible rules, as well, as semantic notions of satisfiability and validity of a sequent are defined as in Section 3.1.1 for standard SC.

Properties

Let us consider some important features of Seligman's SC, namely:

- The lack of restrictions on formulae in sequents
- The construction and generality of the hybrid rules
- The presence of elimination rules for nominals and sat-operators
- Admissibility of the cut of the form:

$$(Cut) \quad \frac{\Gamma \Rightarrow \Delta, \varphi \qquad \varphi, \Gamma' \Rightarrow \Delta'}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}$$

By the first, we simply mean that it is an extension of ordinary SC to hybrid language. All formulae are permitted as elements of sequents which is the most fundamental difference with sat-calculus of Blackburn which will be discussed next.[2] Due to this feature, Seligman is able to obtain SC for hybrid logic just by the addition of new rules to standard SC; no substantial modification of the classical basis is needed.

Such a modular approach makes easier the comparison of this calculus with axiomatic system. We can easily prove the following:

Lemma 12.1 *If φ is a thesis of H-$\mathbf{K}^{+}_{\mathbf{H@}}$, then $\Rightarrow \varphi$ is derivable in Seligman's SC*

We omit the proof; it is sufficient to prove all the axioms and show (with the help of cut) a derivability of all the rules of H-$\mathbf{K}^{+}_{\mathbf{H@}}$, which is routine.

Nevertheless, the proof of the above lemma may be instructive; one can find that application of (N_1) is not necessary for the proofs of axioms (although it can shorten them). Also the rule $(S\text{-}NAME)$ is needed only for the proof of derivability of $(NAME)$ from axiom system, so if we need SC equivalent to H-$\mathbf{K}_{\mathbf{H@}}$ it is also dispensable. In [248] we have a direct evidence for this redundancy, because completeness proof is provided for SC formalization of non-modal hybrid language with sat-operators and nominals. In this case, to ordinary SC with cut only three rules are added, a version of $(TERM)$ and two @-introduction rules of the form:

$$(@\Rightarrow')\ \ \frac{\Gamma\Rightarrow\Delta, i \qquad \varphi, \Gamma\Rightarrow\Delta}{@_i\varphi, \Gamma\Rightarrow\Delta} \qquad\qquad (\Rightarrow @')\ \ \frac{\Gamma\Rightarrow\Delta, i \qquad \Gamma\Rightarrow\Delta, \varphi}{\Gamma\Rightarrow\Delta, @_i\varphi}$$

Although these rules are different from the rules described above for @, it may be shown that they are equivalent in the sense of mutual derivability. The lack of rules for @-elimination shows that they are also redundant if we search for a complete system with cut, but they are required for obtaining a cut-free version. In fact, for stronger languages we need them also in cut-free proofs, as well as $(S\text{-}NAME)$.

Seligman's rules are very natural. In [251] the rules are obtained by the series of syntactical transformations, but [249, 250] contains a justification of them by reference to intuitively plausible patterns of reasoning. We just comment on the sense of $(TERM)$ and $(S\text{-}NAME)$. The former means that if some sat-formulae follow from other sat-formulae locally (in some state i), then this entailment holds generally, independently of the state of evaluation. The latter is justified similarly: if in arbitrary state (new i) we have local entailment, then it holds generally.

[2] In fact, Seligman considers in [251] also sat-calculus and mixed calculus as stages in the series of transformations from SC for first-order logic.

However, this calculus may seem strange for researchers familiar with Gentzen approach, since for nominals and sat-operators we have both introduction and elimination rules. It is not a standard solution in SC, where constants are usually characterised by introduction rules only. Also some other of the Seligman's rules lack several properties required from "good" sequent calculi (see e.g. the properties discussed by Avron or Wansing [280], like the lack of symmetry for nominal rules). Despite these drawbacks, Seligman's system presents quite good behaviour. In particular, cut-elimination theorem holds for this calculus. The proof of this fact is not performed directly for this form of SC, but for the form of SC adequate for first-order logic. For this SC the admissibility of cut is proved in the standard way by induction on the rank and the degree of cut applications. Seligman then obtains his calculus for hybrid logic by the series of transformations from this origin-SC. These transformations preserve many important properties, among them cut admissibility. Direct proofs of cut-elimination for SC without modals may be found in [248, 250].

From the point of view of practical utility, as a tool for proof-search, Seligman's system has some drawbacks however, connected with the lack of analyticity. Of course, cut is admissible, but cut-elimination is not in itself the sufficient condition for obtaining practically useful system for proof-search. One can easily note that in Seligman's SC cut is not the only nondeterministic rule. Because of $(TERM)$, $(S\text{-}NAME)$ and two @-elimination rules, cut-free Seligman's system does not satisfy subformula property. It is interesting to note, that the version of SC for nonmodal logic from [250] satisfies some generalised form of subformula property, namely:

Every formula occuring in the derivation of $\Gamma \Rightarrow \Delta$ is a quasi-subformula of elements in $\Sigma = \Gamma \cup \Delta \cup N$, where N is a finite set of nominals and φ is a quasi-subformula of Σ iff either φ is an ordinary subformula of some $\psi \in \Sigma$ or $\varphi := @_i\psi$ and both i and ψ are subformulae of some formulae in Σ.

But in this form of SC both rules for @-elimination additionaly satisfy side conditions to the effect, that φ in eliminated $@_i\varphi$, is not itself sat-formula.

Such a property makes possible to define proof-search procedure and to redefine SC system for Hintikka-style tableau calculus by simply turning upside down all the rules and change all sequents $\Gamma \Rightarrow \Delta$ into sets $\Gamma, \neg\Delta$ like in ordinary modal logic (in fact, some modifications of rules $(H \Rightarrow \Diamond)$ and $(H\Box \Rightarrow)$ are also necessary – cf. the next Section). But it is not clear if a similar form of subformula property may be obtained for considered SC.

Other properties of Seligman's SC are responsible for the fact that we cannot define on the basis of this system any sort of tableau system operating on formulae (like Smullyan's system for classical logic). It is impossible, or at least very difficult, because of the global character of rules for sat-operators and nominal rules described in respective side-conditions. For example, (N_1) is a rule difficult to simulate in a proof system where proof consists of single formulae as basic items, because an application of such a rule requires performing a global transformation on actual proof. In systems like ND or Smullyan's tableau, more natural solution is to use some kind of rewrite rules that operate locally. We will return to this question later. A simulation of rules like $(TERM), (S\text{-}NAME)$ and rules for sat-operators in tableau system would demand a presence of some nominal as a context for the whole branch, which is possible but rather an artificial solution in such systems.

Also the transfer of these rules into the context of ND systems may be difficult in some respect. For example, one can show that in ordinary modal logic, the rules of Ohnishi/Matsumoto SC have natural ND-counterparts in Fitch's style system. One can ask if such a transfer is possible with respect to Seligman's rules. We will focus on this problem in the section devoted to ND-systems.

Extensions

In fact, Seligman's system is stronger than the reduct we've discussed. This is a consequence of the mode of its construction, namely by the transformation of SC for first-order logic. The final calculus contains also rules for $\downarrow, \mathcal{E}, \exists$.

$$(\downarrow\Rightarrow)\ \frac{i, \varphi[u/i], \Gamma \Rightarrow \Delta}{i,\ \downarrow u\varphi,\ \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\downarrow)\ \frac{i, \Gamma \Rightarrow \Delta, \varphi[u/i]}{i, \Gamma \Rightarrow \Delta,\ \downarrow u\varphi}$$

$$(\mathcal{E}\Rightarrow)^1\ \frac{@_i\varphi, \Gamma \Rightarrow \Delta}{E\varphi, \Gamma \Rightarrow \Delta} \qquad (\Rightarrow\mathcal{E})\ \frac{\Gamma \Rightarrow \Delta, @_i\varphi}{\Gamma \Rightarrow \Delta, E\varphi}$$

$$(H\exists\Rightarrow)^1\ \frac{\varphi[u/i], \Gamma \Rightarrow \Delta}{\exists u\varphi, \Gamma \Rightarrow \Delta} \qquad (H \Rightarrow\exists)\ \frac{\Gamma \Rightarrow \Delta, \varphi[u/i]}{\Gamma \Rightarrow \Delta, \exists u\varphi}$$

Side condition: 1. where i does not occur in Γ, Δ and φ.

Since the system is modular, by combining these rules over the basic system we can obtain adequate formalizations of basic hybrid logics in these languages. Seligman does not consider any extension of his system to stronger modal logics than **K**. The fact that it is not sat-calculus opens the problem if this system may be modified for logics in languages without sat-operators. But it is not so obvious how to obtain a system complete (and cut-free) for $\mathbf{K_H}$. It is not enough to get rid of 4 rules for sat-operators, since @ is present also in the rules for modals (they do not satisfy properties of separation and explicitness, see [280]) – different rules are necessary. Tzakova defined a tableau system for logics without sat-operators which may be transformed into cut-free SC, but it applies also external labels (cf. the section on tableau calculi).

12.2.2 Sequent Sat-Calculus of Blackburn

As we mentioned earlier, Blackburn's system is defined on sat-formulae only, so, in a sense, we have a nonstandard form of SC. On the other hand, the form of rules is quite close to standard Gentzen's format as we shall see. All the definitions concerning proof, derivable and admissible rules e.t.c. are the same as for Seligman's SC.

General rules

$(@AX)$ $\Gamma \Rightarrow \Delta$, where $\Gamma \cap \Delta \neq \varnothing$

$(@C\Rightarrow)$ $\dfrac{@_i\varphi, @_i\varphi, \Gamma\Rightarrow \Delta}{@_i\varphi, \Gamma\Rightarrow \Delta}$ $(@\Rightarrow C)$ $\dfrac{\Gamma\Rightarrow \Delta, @_i\varphi, @_i\varphi}{\Gamma\Rightarrow \Delta, @_i\varphi}$

$(@\neg\Rightarrow)$ $\dfrac{\Gamma\Rightarrow \Delta, @_i\varphi}{@_i\neg\varphi, \Gamma\Rightarrow \Delta}$ $(@\Rightarrow\neg)$ $\dfrac{@_i\varphi, \Gamma\Rightarrow \Delta}{\Gamma\Rightarrow \Delta, @_i\neg\varphi}$

$(@\wedge\Rightarrow)$ $\dfrac{@_i\varphi, @_i\psi, \Gamma\Rightarrow \Delta}{@_i(\varphi\wedge\psi), \Gamma\Rightarrow \Delta}$ $(@\Rightarrow\wedge)$ $\dfrac{\Gamma\Rightarrow \Delta, @_i\varphi \qquad \Gamma\Rightarrow \Delta, @_i\psi}{\Gamma\Rightarrow \Delta, @_i(\varphi\wedge\psi)}$

$(@\vee\Rightarrow)$ $\dfrac{@_i\varphi, \Gamma\Rightarrow \Delta \qquad @_i\psi, \Gamma\Rightarrow \Delta}{@_i(\varphi\vee\psi), \Gamma\Rightarrow \Delta}$ $(@\Rightarrow\vee)$ $\dfrac{\Gamma\Rightarrow \Delta, @_i\varphi, @_i\psi}{\Gamma\Rightarrow \Delta, @_i(\varphi\vee\psi)}$

$(@\rightarrow\Rightarrow)$ $\dfrac{\Gamma\Rightarrow \Delta, @_i\varphi \qquad @_i\psi, \Gamma\Rightarrow \Delta}{@_i(\varphi\rightarrow\psi), \Gamma\Rightarrow \Delta}$ $(@\Rightarrow\rightarrow)$ $\dfrac{@_i\varphi, \Gamma\Rightarrow \Delta, @_i\psi}{\Gamma\Rightarrow \Delta, @_i(\varphi\rightarrow\psi)}$

Rules are defined on sequents $\Gamma \Rightarrow \Delta$, where both Γ, Δ are finite multisets of sat-formulae. This is why Blackburn needs structural rules of contraction ($(@C \Rightarrow)$ and $(@ \Rightarrow C)$). In fact, he uses axioms of the form: $@_i\varphi \Rightarrow @_i\varphi$ and in consequence he needs also rules of weakening.

Modal rules

$$(@\Rightarrow)\ \frac{@_i\varphi, \Gamma\Rightarrow \Delta}{@_j@_i\varphi,\ \Gamma\Rightarrow \Delta} \qquad (\Rightarrow @)\ \frac{\Gamma\Rightarrow \Delta, @_i\varphi}{\Gamma\Rightarrow \Delta,\ @_j@_i\varphi}$$

$$(@\Diamond\Rightarrow)^1\ \frac{@_i\Diamond j,\ @_j\varphi, \Gamma\Rightarrow \Delta}{@_i\Diamond\varphi, \Gamma\Rightarrow \Delta} \qquad (@\Rightarrow\Diamond)\ \frac{\Gamma\Rightarrow \Delta, @_j\varphi}{@_i\Diamond j,\ \Gamma\Rightarrow \Delta, @_i\Diamond\varphi}$$

$$(@\Box\Rightarrow)\ \frac{@_j\varphi, \Gamma\Rightarrow \Delta}{@_i\Box\varphi, @_i\Diamond j, \Gamma\Rightarrow \Delta} \qquad (@\Rightarrow\Box)^1\ \frac{@_i\Diamond j, \Gamma\Rightarrow \Delta, @_j\varphi}{\Gamma\Rightarrow \Delta, @_i\Box\varphi}$$

Side condition: 1. where j does not occur in $\Gamma \cup \Delta \cup \{\varphi\}$.

Special rules

$$(@Ref)\ \frac{@_i i, \Gamma\Rightarrow \Delta}{\Gamma\Rightarrow \Delta} \qquad (@Sym)\ \frac{@_i j, \Gamma\Rightarrow \Delta}{@_j i, \Gamma\Rightarrow \Delta}$$

$$(@Nom)\ \frac{@_i\varphi,\ \Gamma\Rightarrow \Delta}{@_i j, @_j\varphi, \Gamma\Rightarrow \Delta} \qquad (@Bridge)\ \frac{@_i\Diamond j, \Gamma\Rightarrow \Delta}{@_i\Diamond k, @_k j,\ \Gamma\Rightarrow \Delta}$$

In fact, both $(@Sym)$ and $(@Bridge)$ are derivable with the help of cut. Below we display a proof of the derivability of $(@Sym)$:

$$\dfrac{\dfrac{@_i j, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, @_i\neg j}\ (@\Rightarrow\neg) \qquad \dfrac{\dfrac{\dfrac{@_j j \Rightarrow @_j j\ (@AX)}{@_j j, @_j\neg j \Rightarrow}\ (@\neg\Rightarrow)}{@_j\neg j \Rightarrow}\ (@Ref)}{@_j i, @_i\neg j \Rightarrow}\ (@Nom)}{@_j i, \Gamma \Rightarrow \Delta}\ (Cut)$$

Blackburn's sat-SC is also cut-free but this fact is not proved constructively but rather shown indirectly. The calculus is obtained from cut-free tableau system (see the Section on tableau calculi) for which Hintikka style constructive completeness proof is provided (by constructing suitably defined downward-saturated sets from open branches).

It is instructive to compare this calculus with previously presented SC of Seligman. One can easily notice that rules of Blackburn differ from those of Seligman not only with respect to the kind of formulae they use. In fact, Seligman obtains also sat-calculus as one of the stages in the process of transformations leading to SC described in the previous paragraph. So our comparison will be more direct if we refer to this sat-calculus of Seligman, instead of the final form of his SC.

The fact that sequents in Blackburn SC are defined on multisets is not essential, we can define this SC also on sequents made of lists of sat-formulae and just add rules of permutation. More serious differences concern some rules:

- for $(@ \Rightarrow \Diamond)$ and $(@\Box \Rightarrow)$
- forms of $(@Ref)$
- nominal rules

Seligman's rules for modalities look as follows:

$$(H \Rightarrow\Diamond')\ \frac{\Gamma\Rightarrow \Delta, @_i\Diamond j \quad \Gamma\Rightarrow \Delta, @_j\varphi}{\Gamma\Rightarrow \Delta, @_i\Diamond\varphi} \qquad (H\Box\Rightarrow')\ \frac{\Gamma\Rightarrow \Delta, @_i\Diamond j \quad @_j\varphi, \Gamma\Rightarrow \Delta}{@_i\Box\varphi, \Gamma\Rightarrow \Delta}$$

They are interderivable (i.e. mutually derivable) with Blackburn's rules which may be shown by referring to Lemma 3.1. Seligman's variants are better from the proof-theoretical perspective, but Blackburn's rules are better if we need a calculus for actual proof-search. In such a case it is handy to redefine the calculus in terms of sequents built up from sets of formulae, but two respective rules must be changed in order to keep the effect of contraction:

$$(@ \Rightarrow\Diamond')\ \frac{@_j\Diamond i, \Gamma\Rightarrow \Delta, @_j\Diamond\varphi, @_i\varphi}{@_j\Diamond i,\ \Gamma\Rightarrow \Delta, @_j\Diamond\varphi} \qquad (@\Box\Rightarrow')\ \frac{@_i\varphi, @_j\Box\varphi, @_j\Diamond i, \Gamma\Rightarrow \Delta}{@_j\Box\varphi, @_j\Diamond i, \Gamma\Rightarrow \Delta}$$

Seligman's sat calculus makes use of one more axiom $(R@)$ of the form: $\Gamma \Rightarrow \Delta, @_i i$ instead of Blackburn's rule $(@Ref)$. They are also interderivable.

Finally, note that Blackburn's SC in addition to $(@Ref)$ has three additional special rules, whereas Seligman's system uses only one pair of rules:

$$(@L_1)\ \frac{@_ij, \Gamma[i]\Rightarrow \Delta[i]}{@_ij, \Gamma[j]\Rightarrow \Delta[j]} \qquad (@L_2)\ \frac{@_ij, \Gamma[j]\Rightarrow \Delta[j]}{@_ij, \Gamma[i]\Rightarrow \Delta[i]}$$

where $\Gamma[i]$ means that i occur in Γ and $\Gamma[j]$ is the result of replacement of j for i in Γ

This pair of rules is contracted in one – (N_1) in the final calculus, since $@_ij$ is replaced by a pair of nominals i, j. One can check that three rules of Blackburn are derivable in Seligman's version and that his substitution rules are admissible in Blackburn's SC (which needs more complicated proof). If we use Seligman's substitution rules we can often obtain shorter proofs than in Blackburn's version. It is because of the global character of their application. On the other hand, local rewrite rules, like these in Blackburn's version, may be naturally applied in ND systems or tableau systems defined on formulae. In fact – as we remarked above – Blackburn obtains his SC from such tableau system which explains why he prefers local rules as primitives.

Neither Seligman nor Blackburn consider extensions to stronger logics. Blackburn's general proposal for tableau sat-calculus will be described in the next section. We return to this question also in the section on ND by the way of presenting a system of Braüner. On the basis of his ND-system Braüner presents yet another variant of sat SC-calculus of a uniform character with very general and strong rule covering all logics characterized by geometric theories. For the time being we just note that on the basis of Blackburn SC we may formalize all logics characterized by universal implications. It may be done by translation of the general rule of Negri [194] which was mentioned in Secton 9.1. Very briefly, for each universal implication of the form $\forall x_1...x_i(\varphi_1 \wedge ... \wedge \varphi_k \rightarrow \psi_1 \vee ... \vee \psi_n)$, where all φ's and ψ's are relational formulae, the general schema of SC rule is:

$$(HR\text{-}UI) \quad \frac{\psi'_1, \varphi'_1, ..., \varphi'_k, \Gamma \Rightarrow \Delta \mid, ..., \mid \psi'_n, \varphi'_1, ..., \varphi'_k, \Gamma \Rightarrow \Delta}{\varphi'_1, ..., \varphi'_k, \Gamma \Rightarrow \Delta}$$

where each φ'_i (and ψ'_i) is $HT(\varphi_i)$ (cf. the definition of HT-translation in the preceding Chapter.).

12.2.3 Nonstandard Sequent Calculi

As we have mentioned in Chapter 7, there is a lot of nonstandard sequent calculi for ordinary modal logic, substantially enriching and modifying original Gentzen ideas (see [280, 281] for an overview). In contrast, the number of nonstandard sequent calculi for hybrid modal logic is poor. Sat-calculus of Blackburn, although nonstandard, represents rather small departure from the original Gentzen's approach. It is perhaps due to the fact that hybrid languages are more expressive, and all this metalogical apparatus applied in nonstandard calculi to deal with the limitations of ordinary languages, is indeed of no use.

Anyway, one should note that two really nonstandard calculi were devised. One of them, due to Goré and Demri [79], belongs to the familly of display calculi, so we are not going to describe it here, because such a presentation would require an introduction of too many technical details.

So we only point out that [79] contains display calculus for hybrid tense logic with difference modality. One can find a good exposition of display calculi for modal logics in Wansing [280].

The second system, due to Demri [78], is the calculus for $\mathbf{Kt_H}$ and a huge class of its extensions. Because in this case the departures from standard SC are not so great we describe briefly its main distinctive features.

1. The calculus is based on the idea of using "implicit prefixes" applied by Konikowska [164]. This role is played by nominals. Since sat-operators are not present in the language, all the rules are defined on the formulae of the shape $i \rightarrow \varphi$. It means that even the rules for boolean constants must be suitably transformed. For instance, standard ($\Rightarrow\rightarrow$) obtains a form:

$$(D \Rightarrow\rightarrow) \quad \frac{i \rightarrow \varphi, \Gamma \Rightarrow \Delta, i \rightarrow \psi}{\Gamma \Rightarrow \Delta, i \rightarrow (\varphi \rightarrow \psi)}$$

Similarly for other rules. So $i \rightarrow$ plays the role of $@_i$ in sat-calculi, and that's why we put Demri's system in the class of sat-calculi in our taxonomy of proof systems. Demri's solution shows how to dispense with sat-operators in the presence of backward-looking modalities. The transmission between such formulae and ordinary hybrid formulae we want to prove, is realized by the rule:

$$(Start) \quad \frac{\Rightarrow i \rightarrow \varphi}{\Rightarrow \varphi} \quad \text{where } i \notin \varphi$$

This is clearly a sequent version of ($NAME'$) discussed on the ground of axiomatic formulations. This rule is applied only once in a proof – its place is just at the root of the proof-tree.

2. In order to reduce branching, Demri based his calculus on tableau-like system KE of D'Agostino and Mondadori [2] (cf. Chapter 3). Sequent counterparts of nonbranching β-rules, e.g. for negated conjunction, are realized in the following form:

$$(D{\Rightarrow}\wedge_1) \quad \frac{\Gamma, i \rightarrow \varphi \Rightarrow \Delta, i \rightarrow \psi}{\Gamma, i \rightarrow \varphi \Rightarrow \Delta, i \rightarrow \varphi \wedge \psi} \qquad (D{\Rightarrow}\wedge_2) \quad \frac{\Gamma, i \rightarrow \psi \Rightarrow \Delta, i \rightarrow \varphi}{\Gamma, i \rightarrow \psi \Rightarrow \Delta, i \rightarrow \varphi \wedge \psi}$$

Note that $i \rightarrow \varphi$ ($i \rightarrow \psi$) from the antecedent of the conclusion-sequent must be still present in the premise-sequent, otherwise the calculus would be incomplete (it may be used more than once in the course of proof-search). Similar pairs of one-premise rules for $\rightarrow$ and $\vee$ must be introduced instead of ordinary two-premise rules ($\vee \Rightarrow$) and ($\rightarrow\Rightarrow$).

3. This SC is incomplete without cut, similarly as KE, but since cut without restriction on its applicability makes the calculus practically useless, it raises the question of convenient delimitation of its applications. We recall that in ordinary KE for **CPL** it is sufficient to use cut only for introducing lacking minor premises (and their negations) for nonbranching β-rules. Demri's system also satisfies some restrictions on the applicability of cut, namely, a cut-formula $i \rightarrow \psi$ introduced in the proof of φ must obey the following:

- i is a nominal which was introduced as "new" by the application of rules that satisfy suitable side-condition
- ψ is either a subformula of φ, or j introduced as "new", or has a form $G\neg j$ with j introduced as "new"

The shape of all rules and some more restrictions put on their applications make this calculus satisfy 3 conditions which give it almost an analytic character. However Demri himself is sceptical with respect to usefulness of this calculus in the field of automated deduction, he is rather concerned with defining a uniform complete framework.

4. The number of rules for nominals and temporal constants is rather numerous (and redundant), so we display below only 4 central rules for G and H:

$$(DG\Rightarrow)\ \frac{\Gamma, i \rightarrow G\varphi, j \rightarrow \varphi \Rightarrow \Delta, i \rightarrow G\neg j}{\Gamma, i \rightarrow G\varphi \Rightarrow \Delta, i \rightarrow G\neg j} \quad (D\Rightarrow G)\ \frac{\Gamma \Rightarrow \Delta, j \rightarrow \varphi, i \rightarrow G\neg j}{\Gamma \Rightarrow \Delta, i \rightarrow G\varphi}$$

$$(DH\Rightarrow)\ \frac{\Gamma, j \rightarrow H\varphi, i \rightarrow \varphi \Rightarrow \Delta, i \rightarrow G\neg j}{\Gamma, j \rightarrow H\varphi \Rightarrow \Delta, i \rightarrow G\neg j} \quad (D\Rightarrow H)\ \frac{\Gamma \Rightarrow \Delta, j \rightarrow \varphi, j \rightarrow G\neg i}{\Gamma \Rightarrow \Delta, i \rightarrow H\varphi}$$

Clearly, in $(D \Rightarrow G)$ and $(D \Rightarrow H)$, j does not occur in the conclusion sequent (this is the proviso concerning "new" nominals when rules are read off from bottom to top). One can easily note that in these rules, formulae of the form $i \rightarrow G\neg j$ in the succedent of a sequent just inform that j is after i in the flow of time. Similarly, in many other rules, formulae of the form $i \rightarrow j$ in the antecedent identify points i and j, whereas in the succedent they serve as an information on their inequality. Interested reader should consult [78] for the exposition of the whole calculus.

5. The last thing of great importance is the definition of the schema of one more (multi-) branching rule which covers a huge class of first-order frame defining conditions. We omit the details because of their complexity

and the lack of space, but in the section on ND we briefly comment on the scope of this extension.

12.3 Tableau Systems

There are two groups of different tableau systems for MHL. The first one has a mixed character, i.e. except nominals it applies extra metalinguistic labels called (after Fitting) prefixes. This solution is due to Tzakova [277] and was recently refined and improved by Bolander, Braüner and Blackburn [44, 45][3]; a similar kind of TS was devised also by van Eijck [83]. The expressive strength of hybrid languages may suggest that the application of external labels is of no use in hybrid logics. This is not the whole truth however, at least if we are concerned with hybrid logics without sat-operators.

The second group contains a sat-calculus due to Blacburn [32] and its modifications from [44, 45]. It is TS-counterpart of sat-SC presented in the preceding Section. In fact, tableau calculus was presented as primary system in [32], then SC was extracted from them.

In both groups all the proposed systems are of Smullyan's type, i.e. with nodes of a tableau being single formulae not sets of formulae like in Hintikka style tableau calculi.[4] The definition of a tableau for φ in both systems is standard; it is a tree of formulae with φ (with a prefix in Tzakova's system) as the root (on the top) which is expanded by expansion rules, typically decomposing formulae on its parts. A tableau is closed if all its branches are closed. $\vdash_T \varphi$ iff there is a closed tableau for $\neg\varphi$ (again with a prefix in Tzakova's system – see below). Additionaly, we will show how to simulate very powerfull and uniform, strongly labelled TS of Baldoni [18], on the basis of Blackburn's sat-calculus.

12.3.1 Mixed Calculi

Tzakova's system is the first TS constructed for hybrid logics. As we mentioned above it is a labelled system of mixed character with extra prefixes added to formulae. Labels are exactly as in Fitting's system so we continue to use σ, τ for denoting them and $\sigma : \varphi$ to denote a labelled formula. But it is not just one more system with medium labelling since it contains elements of strong labelling as well. Tzakova uses two types of formulae:

[3] I would like to thank anonymous referee for drawing my attention to these valuable papers.

[4] Although [44] contains a reformulation of van Eijck's TS in terms of Hintikka format.

- *labelled sentences* of the form $\sigma : \varphi$, where φ is a hybrid formula and σ is a label
- *accessibility sentences* of the form $\sigma < \tau$, where both σ and τ are labels

We say that τ is accessible from σ if either $\tau = \sigma.i$ or $\sigma < \tau$ is on the branch. Hence, a proof of φ in Tzakova's system is a closed tableau for $1 : \neg\varphi$ built up by means of the following rules:

The rules for the weakest logic $\mathbf{K_H}$:

$(T\bot_1)$	$\sigma : \varphi,\ \sigma : \neg\varphi \ / \ \bot$
$(T\bot_2)$	$\sigma : i,\ \tau : i,\ \sigma : \varphi,\ \tau : \neg\varphi \ / \ \bot$
$(T\neg\neg)$	$\sigma : \neg\neg\varphi \ / \ \sigma : \varphi$
$(T\alpha)$	$\sigma : \alpha \ / \ \sigma : \alpha_1,\ \sigma : \alpha_2$
$(T\beta)$	$\sigma : \beta \ / \ \sigma : \beta_1 \ \mid \ \sigma : \beta_2$
$(T\Box E)$	$\sigma : \Box\varphi \ / \ \tau : \varphi$, for any τ accessible from σ
$(T\neg\Box E)$	$\sigma : \neg\Box\varphi \ / \ \tau : \neg\varphi$, where τ is a new label accessible from σ
$(T\Diamond E)$	$\sigma : \Diamond\varphi \ / \ \tau : \varphi$, where τ is a new label accessible from σ
$(T\neg\Diamond E)$	$\sigma : \neg\Diamond\varphi \ / \ \tau : \varphi$, for any τ accessible from σ
(Lab)	$\sigma : \varphi \ / \ \sigma : i$, where i is a new nominal
$(S\text{-}Id)$	$\sigma : i,\ \tau : i \ / \ \sigma < \sigma'$, provided σ' is accessible from τ
$(L\text{-}Id)$	$\sigma : i,\ \tau : i,\ \sigma' : j,\ \tau : j \ / \ \sigma : j,\ \sigma' : i$

Note that there are two rules for closing branches. One of them is standard, whereas the other reflects identity of labels by two additional premises ($\sigma : i$ and $\tau : i$). A similar situation is present in case of $(S\text{-}Id)$ and $(L\text{-}Id)$. Tzakova provided also rules for $@, \downarrow$ and $\forall$:

$(T@)$	$\sigma : @_i\varphi \ / \ \tau : i,\ \tau : \varphi$
$(T\neg@)$	$\sigma : \neg@_i\varphi \ / \ \tau : i,\ \tau : \neg\varphi$
$(T\downarrow)$	$\sigma : \downarrow u\varphi,\ \sigma : i \ / \ \sigma : \varphi[u/i]$
$(T\neg\downarrow)$	$\sigma : \neg\downarrow u\varphi,\ \sigma : i \ / \ \sigma : \neg\varphi[u/i]$
$(T\forall)$	$\sigma : \forall u\varphi \ / \ \sigma : \varphi[u/i]$
$(T\neg\forall)$	$\sigma : \neg\forall u\varphi \ / \ \sigma : \neg\varphi[u/i]$, where i is a new nominal

In both rules for @, if $\tau : i$ is already present on the branch, we just add the second conclusion of a rule, otherwise τ is a new label on the branch.

The system of Tzakova is weakly complete for $\mathbf{K_H}, \mathbf{K_{H@}}, \mathbf{K_{H\downarrow}}, \mathbf{K_{H@\downarrow}}$, $\mathbf{K_{H\forall}}$ and $\mathbf{K_{H@\forall}}$. There are no rules for stronger logics but they can be obtained by adding labelled rules from Chapter 8. The proof of completeness is via construction of suitable downward saturated sets but it does not yield a decision procedure for $\mathbf{K_H}$ and $\mathbf{K_{H@}}$ which are decidable logics. Problems with loop generation may be avoided, if we change (S-Id) into the following rules:

(S-Id')	$\sigma : i,\ \tau : i,\ \sigma : \Box\varphi\ /\ \sigma' : \varphi$, provided σ' is accessible from τ
(S-Id'')	$\sigma : i,\ \tau : i,\ \sigma : \neg\Diamond\varphi\ /\ \sigma' : \neg\varphi$, provided σ' is accessible from τ

The reproduced proof of $\Diamond(i \wedge \Box(j \rightarrow p)) \rightarrow \neg\Diamond(i \wedge \Diamond(j \wedge \neg p))$ shows well the application of rules, including (S-Id):

1	$1 : \neg(\Diamond(i \wedge \Box(j \rightarrow p)) \rightarrow \neg\Diamond(i \wedge \Diamond(j \wedge \neg p)))$		
2	$1 : \Diamond(i \wedge \Box(j \rightarrow p))$		$(1, T\alpha)$
3	$1 : \neg\neg\Diamond(i \wedge \Diamond(j \wedge \neg p))$		$(1, T\alpha)$
4	$1.1 : i \wedge \Box(j \rightarrow p)$		$(2, T\Diamond E)$
5	$1.1 : i$		$(4, T\alpha)$
6	$1.1 : \Box(j \rightarrow p)$		$(4, T\alpha)$
7	$1 : \Diamond(i \wedge \Diamond(j \wedge \neg p))$		$(3, T\neg\neg)$
8	$1.2 : i \wedge \Diamond(j \wedge \neg p)$		$(7, T\Diamond E)$
9	$1.2 : i$		$(8, T\alpha)$
10	$1.2 : \Diamond(j \wedge \neg p)$		$(8, T\alpha)$
11	$1.2.1 : j \wedge \neg p$		$(10, T\Diamond E)$
12	$1.2.1 : j$		$(11, T\alpha)$
13	$1.2.1 : \neg p$		$(11, T\alpha)$
14	$1.1 < 1.2.1$		$(5, 9, S\text{-}Id)$
15	$1.2.1 : j \rightarrow p$		$(6, 14, T\Box E)$
16	$1.2.1 : \neg j$	$1.2.1 : p$	$(15, T\beta)$
17	$\bot\ (16, 12, \bot_1)$	$\bot$	$(16, 13, T\bot_1)$

Since Tzakova uses Fitting's prefixes it is natural to consider if accessibility sentences are really needed. In standard Fitting's tableau system for modal logics, the construction of labels gives all the required information necessary for extraction of a falsifying model from open branch. But in hybrid logic there is an interplay between labels and nominals. The latter give additional information about links between states in attempted falsifying model. Note that (S-Id) is the only rule that enters accessibility sentences as nodes of a tableau. In all other rules having provisos concerning accessibility between labels, the presence of such sentences on a branch is not necessary for application since all the required information is implicit in the shape of prefixes. When (S-Id) is applied, such accessibility sentences appear in a tableau and may be used as actual second premise for application of $(T\Box E)$, $(T\neg\Diamond E)$ or $(S\text{-}Id)$. Since the presence of accessibility sentences is necessary for hybrid logics, it would be in fact simpler and more elegant to formulate this system with the help of strong labelling (cf. [99] or [21]) using just natural numbers (or any other symbols) as labels and accessibility sentences as the only explicit source of information about the structure of attempted falsifying model. Obviously, in such a variant we must change a formulation of some rules:

$(T\Box E')$ $\sigma < \tau,\ \sigma : \Box\varphi \ /\ \tau : \varphi$
$(T\neg\Box E')$ $\sigma : \neg\Box\varphi \ /\ \sigma < \tau,\ \tau : \neg\varphi$, where τ is a new label
$(T\Diamond E')$ $\sigma : \Diamond\varphi \ /\ \sigma < \tau,\ \tau : \varphi$, where τ is a new label
$(T\neg\Diamond E')$ $\sigma < \tau,\ \sigma : \neg\Diamond\varphi \ /\ \tau : \varphi$
$(S\text{-}Id')$ $\sigma : i,\ \tau : i,\ \tau < \sigma' \ /\ \sigma < \sigma'$

In fact, Bolander and Braüner [44] provided an improvement of Tzakova's system along these lines. There are three points worth noting in the context of their work:

1. replacement of Fitting's labels by strong labelling
2. simplification of rules
3. providing decision procedures for considered logics

The first point was discussed above. But changes in the set of rules are not only the consequence of changes in the type of labelling. They follow also from the fact that Tzakova's termination proof for decidable logics is flawed. [44] contains a system for $\mathbf{K_{H@A}}$, where Tzakova's rules for modal constants are replaced with the variants described above. Moreover, (Lab) and $(T\bot_2)$ are deleted and two Id-rules are replaced by one:

(Id) $\sigma : i,\ \sigma : \varphi, \tau : i \ /\ \tau : \varphi$

The rules for @ are like in Tzakova's system but both with the proviso that τ in the conclusion is new. The same proviso is added to one additional rule which is not necessary but introduced for simplification of completeness proof:

$(T\neg)$ $\sigma : \neg i \ /\ \tau : i$

The system contains also two rules for global modality (only for $\mathcal{E}$ – dual rules for $\mathcal{A}$ are straightforward):

$(\mathcal{E}E)$ $\quad \sigma : \mathcal{E}\varphi \ / \ \tau : \varphi$, provided τ is new
$(\neg\mathcal{E}E)$ $\quad \sigma : \neg\mathcal{E}\varphi \ / \ \sigma' : \neg\varphi$, provided σ' is already on the branch

Further improvements of this approach are provided by Bolander and Blackburn in [45] which contains complete formalizations and decision procedures for multimodal versions of: $\mathbf{K_H}, \mathbf{K_{H@}}, \mathbf{K_{H@A}}$ and $\mathbf{K_{H@A}}$ with converse modalities (this include $\mathbf{KB_{H@A}}$, as well as $\mathbf{Kt_{H@A}}$, due to admitted multimodality[5]). The set of rules is essentially the same as in [44] (with minor difference that accessibility sentences are now written as $\sigma\Diamond_i\tau$), it is shown however, that small changes in the set of rules may improve decision procedure in some cases. If in $\mathbf{K_H}$ and $\mathbf{K_{H@}}$ we replace (Id) with two more constrained rules:

(νId) $\quad \sigma : \varphi,\ \sigma : i,\ \tau : i \ / \ \tau : \varphi$,
where τ is the earliest label with i on the branch
$(TNom)$ $\quad \sigma : i,\ \sigma : j,\ \tau : j \ / \ \tau : i$

then proof-search procedure does not involve loop-check. In the remaining systems some kind of loop-check is necessary to provide termination.

It is important to note that labelled TS's of this sort are the only non-axiomatic formalizations of hybrid logics without sat-operators. In fact, it is exactly the lack of internalized sat-operators in a language, which makes sense to consider external labels in addition to nominals. It was also believed that mixed systems represent better behaviour with respect to proof-search but [45] shows that it is not necessarily the case.

One should note that not all rules in this approach are typical expansion rules of tableau calculi. For example, $(S\text{-}Id), (L\text{-}Id)$ or (Id) and both rules for $\downarrow$ or $(T\Box E'), (T\neg\Diamond E')$ have more than one premise, so their application requires scanning of all the branch above to find the additional premises. It makes them more similar to KE or ND in defining suitable proof-search procedures (cf. a discussion in Chapter 10). The same remark applies also to some rules in Blackburn's sat-system discussed below.

12.3.2 Blackburn's Sat-Calculi

We still use $\alpha-, \beta-$notation but in a slightly modified form represented in the table. Such reformulation is needed for a uniform representation of rules, because tableau calculus of Blackburn is defined on all sat-formulae including negated forms, in contrast to his SC or Braüner's ND-system

[5] In fact, the approach represented in [45] is even more general because modal constants of any arity are considered, as well as their converse modalities wrt. every argument.

where there is no negation. A proof of φ is a closed tableau for $\neg @_i\varphi$, where i is not in φ.

α	α_1	α_2	β	β_1	β_2
$@_i(\varphi \wedge \psi)$	$@_i\varphi$	$@_i\psi$	$\neg @_i(\varphi \wedge \psi)$	$\neg @_i\varphi$	$\neg @_i\psi$
$\neg @_i(\varphi \vee \psi)$	$\neg @_i\varphi$	$\neg @_i\psi$	$@_i(\varphi \vee \psi)$	$@_i\varphi$	$@_i\psi$
$\neg @_i(\varphi \to \psi)$	$@_i\varphi$	$\neg @_i\psi$	$@_i(\varphi \to \psi)$	$\neg @_i\varphi$	$@_i\psi$

Rules for $\mathbf{K_{H@}}$

1. Sat-versions of classical expansion rules:

$(B\neg)$ $@_i\neg\varphi \;/\; \neg @_i\varphi$
$(B\neg\neg)$ $\neg @_i\neg\varphi \;/\; @_i\varphi$
$(B\bot)$ $@_i\varphi,\ \neg @_i\varphi \;/\; \bot$
$(B\alpha)$ $\alpha \;/\; \alpha_1, \alpha_2$
$(B\beta)$ $\beta \;/\; \beta_1 \mid \beta_2$

2. Modal and nominal expansion rules:

$(B@E)$ $@_j @_i\varphi \;/\; @_i\varphi$
$(B\neg @E)$ $\neg @_j @_i\varphi \;/\; \neg @_i\varphi$
$(BRef)$ $\varnothing \;/\; @_i i$, provided i is on the branch
$(BSym)$ $@_i j \;/\; @_j i$
$(BNom)$ $@_i j,\ @_j\varphi \;/\; @_i\varphi$
$(BBridge)$ $@_i j,\ @_k\Diamond i \;/\; @_k\Diamond j$
$(B\Box E)$ $@_i\Box\varphi,\ @_i\Diamond j \;/\; @_j\varphi$
$(B\neg\Box E)$ $\neg @_i\Box\varphi \;/\; @_i\Diamond j,\ \neg @_j\varphi$, j new on the branch
$(B\Diamond E)$ $@_i\Diamond\varphi \;/\; @_i\Diamond j,\ @_j\varphi$, j new on the branch
$(B\neg\Diamond E)$ $\neg @_i\Diamond\varphi,\ @_i\Diamond j \;/\; \neg @_j\varphi$

Blackburn provides also rules for $\downarrow$:

$(B\downarrow E)$ $@_i{\downarrow}u\varphi \;/\; @_i\varphi[u/i]$
$(B\neg\downarrow E)$ $\neg @_i{\downarrow}u\varphi \;/\; \neg @_i\varphi[u/i]$

One can easily check that this calculus is in exact one-to-one correspondence with the earlier described SC. Blackburn considers also its variant for the basic tense hybrid logic $\mathbf{Kt_{H@}}$. We replace 4 rules for modalities by 8 rules for F, P instead of $\Diamond$ and G, H instead of $\Box$; also $(BBridge)$ must be doubled for F and P. Additionaly we need rules:

$(Transpose\text{-}P)$ $@_i Pj \;/\; @_j Fi$
$(Transpose\text{-}F)$ $@_i Fj \;/\; @_j Pi$

Again, it is easy to observe that the rules like $(BSym)$, $(BNom)$, $(BBridge)$, as well as $(BRef)$ are not of a kind characteristic for tableau systems (also two rules for modal operators: $(B\Box E)$ and $(B\neg\Diamond E)$ are of different kind, like in ND-system). In fact, we have a set of rewrite rules added to standard expansion rules. These rules seem to be necessary to handle the theory of equality of nominals but in fact, numerous reductions are possible. We have already remarked that $(BSym)$ is redundant but further investigations have shown that at least some of them are not. In [44] a decision procedure is provided for $\mathbf{K_{H@A}}$, where instead of one unrestricted $(BNom)$ and $(BBridge)$ there are two more specific ones:

$(BNom1)$ $@_i j,\; @_i\varphi \;/\; @_j\varphi$
$(BNom2)$ $@_i j,\; @_j\Diamond k \;/\; @_i\Diamond k$

Clearly, two rules for $\mathcal{E}$ are also provided:

$(B\mathcal{E}E)$ $@_i\mathcal{E}\varphi \;/\; @_j\varphi$, provided j is new
$(B\neg\mathcal{E}E)$ $\neg @_i\mathcal{E}\varphi \;/\; \neg @_j\varphi$, provided j is already on the branch

The restriction to $\mathbf{K_{H@A}}$ yields even simpler calculus. In [45], all rules like $(BRef)$, $(BSym)$, $(BNom)$, $(BBridge)$ are dispensed with, and the only rule of this kind is:

(BId) $@_i j,\; @_i\varphi \;/\; @_j\varphi$

with a constraint that none of the premises are accessibility sentences, i.e. formulae of the shape $@_i\Diamond j$ introduced on the branch as the first conclusion of an application of $(B\Diamond E)$ or $(B\neg\Box E)$. Such a system is not only simpler in the sense of the set of primitive rules but also in the sense of simpler proof-search since no loop-test is needed.

It must be said that in the context of internalised TS also the question of extension to logics stronger than **K** was undertaken. In [32], Blackburn considered only extensions to stronger logics obtained by the addition of pure axioms. For such calculi he proved strong completeness theorem by Hintikka method, using downward saturated sets. In [40] there is a considerable extension provided by the use of so-called *node creating rules.* These rules were already mentioned in the preceding Chapter in connection with PUENF-formulae; they are tableau counterparts of existential saturated rules. Theorem 11.13 concerning existential saturated rules defined for axiomatic systems may be restated:

Theorem 12.1 *Every PUENF-formula (PF)* $\forall u_1, ..., u_m \exists v_1, ..., v_n \varphi$ *corresponds to the node creating rule (NCR) of the form:*

$$\varnothing \ / \ \varphi[u_1/i_1, ..., u_m/i_m, v_1/j_1, ..., v_n/j_n],$$

provided $i_1, ..., i_m$ *occur on the branch,* $j_1, ..., j_n$ *are distinct, unequal to* $i_1, ..., i_m$ *and do not occur on the branch*

Strong completeness theorem for tableau system with (NCR)-rules holds with respect to every class of frames defined by respective PUENF-formulae. The drawback of such a solution lies in the shape of these rules. Instead of expansion rules we must use in fact, special instances of suitable axioms. In many respects these may be replaced by tableau-like rules. For example, to every property defined by Geach Axiom there corresponds the rule covered by the following schema:

$$(@GR) \ \ @_i\Diamond i_1, @_{i_1}\Diamond i_2, ..., @_{i_{m-1}}\Diamond i_m, @_i\Diamond j_1, @_{j_1}\Diamond j_2, ..., @_{j_{s-1}}\Diamond @_{j_s} \ \ / \ @_{i_m}\Diamond k_1, @_{k_1}\Diamond @_{k_2}, ..., @_{k_{n-1}}\Diamond @_m, @_{j_s}\Diamond l_1, @_{l_1}\Diamond @_{l_2}, ..., @_{l_{t-1}}\Diamond @_m,$$

where, $k_1, ..., k_{n-1}, l_1, ..., l_{t-1}, m$, are new nominals.

In order to understand the sense of this rather complicated schema one should recognize, that i is the denotation of x, i_m of y, j_s of z and m of v in Condition (5.7) stated in Section 5.4.3. For example, Church-Rosser property is defined by the rule:

$$(@CR) \ \ @_{i_1}\Diamond i_2, @_{i_1}\Diamond i_3 \ / \ @_{i_2}\Diamond j, @_{i_3}\Diamond j, \text{ where } j \text{ is new}$$

In [36], there is a tableau formalization of **QMHL** presented in the preceding Chapter. To the set of rules for $\mathbf{K_{H@\downarrow}}$ one should add:

$(@\forall E)$	$@_s\forall x\varphi \ / \ @_s\varphi[x/\tau]$, where τ is any grounded term
$(@\neg\forall E)$	$\neg @_s\forall x\varphi \ / \ \neg @_s\varphi[x/c]$, where c is a new parameter
$(@\exists E)$	$@_s\exists x\varphi \ / \ @_s\varphi[x/c]$, where c is a new parameter
$(@\neg\exists E)$	$\neg @_s\exists x\varphi \ / \ \neg @_s\varphi[x/\tau]$, where τ is any grounded term
(ID)	$\varnothing \ / \ \tau = \tau$
(LR)	$\tau_1 = \tau_2, \varphi \ / \ \varphi[\tau_1//\tau_2]$
$(@DD)$	$@_{s_1}s_2 \ / \ @_{s_1}f = @_{s_2}f$
$(@=)$	$@_s(\tau_1 = \tau_2) \ / \ \tau_1 = \tau_2$
$(@\neg =)$	$\neg @_s(\tau_1 = \tau_2) \ / \ \neg(\tau_1 = \tau_2)$

Where s is a nominal or a state variable and a term is grounded if it is a constant (rigid), a parameter or rigidified term (i.e. $@_i f$). Rules for quantifiers are classical since these are possibilistic quantifiers. The last two rules state that equality is rigid and make possible to keep standard rules (ID) and (LR) (not the sat-versions!) since they delete sat-operators. The system is proved complete by translation of every tableau into tableau calculus of the corresponding first-order logic.

Although the tableau system of Blackburn was designed rather for doing refutations by hand, not for automated deduction, one can find an implementation of it, called Hydra, accessible on hybrid web page.

12.3.3 Hybrid Simulation of Baldoni's Strongly Labelled TS

In order to obtain an extensive and uniform formalizations one may also take advantage of solutions obtained for strongly labelled systems. We have already illustrated such a possibility in the preceding section, by providing a hybrid counterpart of Negri's rule for universal implications. In Chapters 8 and 9 we have mentioned Baldoni's strongly labelled TS from [18] which is a uniform and extensive formalization of all mutimodal logics axiomatizable by *a, b, c, d-incestuality axiom*: $\langle a \rangle [b]\varphi \to [c]\langle d \rangle \varphi$, where a, b, c, d are indices of, not necessarily different and possibly complex, modalities (cf. Section 5.3). To define tableau sat-calculus simulating Baldoni's result it is sufficient to take a multimodal version of Blackburn's sat-system (i.e. ($BBridge$) and four modal rules defined for each modality) for $\mathbf{K_{H@}}$ and add the following rules:

$(@;E)$	$@_i\langle a;b\rangle j \ / \ @_i\langle a\rangle k,\ @_k\langle b\rangle j,\ \ k$ new on the branch
$(@\cup E)$	$@_i\langle a\cup b\rangle j \ / \ @_i\langle a\rangle j \ \mid \ @_i\langle b\rangle j$
$(@GGA)$	$@_i\langle a\rangle j,\ @_i\langle c\rangle k \ / \ @_j\langle b\rangle m,\ @_k\langle d\rangle m,\ \ m$ new on the branch

The rules $(@;E)$ and $(@\cup E)$ govern decomposition of complex modalities whereas $(@GGA)$ (from generalized Geach Axiom) corresponds to a, b, c, d-incestuality axiom. In such an extension of Blackburn's calculus one may easily step-wise simulate every proof-tree of Baldoni's TS.

Remark 12.1 We have pointed out that some rules of TS's for hybrid logics use more than one premise which makes them in some respects more similar to KE or ND systems. It is evident especially on the field of automation since more complicated procedures are required than in ordinary TS (cf. considerations in Section 4.2.1). Note however, that e.g., any rule of the form $A,\ B\ /\ C$ may be transformed into $A\ /\ -B \mid C$, where A, B, C are some data structures and $-B$ is a suitable form of the complement of B. Such an operation may be generalised for rules with more premises and more consequences as well. This way we avoid multi-premise rules in TS but at the expense of introducing more branching rules which has other bad consequences. Clearly, this technique may be applied in the other direction if we want to dimnish the number of branching rules in the system. For instance, $(@ \cup E)$ may be replaced by modus tollendo ponens-like pair of rules: $@_i\langle a{\cup}b\rangle j,\ \neg @_i\langle a\rangle j\ /\ @_i\langle b\rangle j$ and $@_i\langle a{\cup}b\rangle j,\ \neg @_i\langle b\rangle j\ /\ @_i\langle a\rangle j$. Clearly, at least an analytic cut is unavoidable in such a system. ♣

12.4 Natural Deduction Systems

The first investigation on ND-systems for hybrid logics was, in fact, undertaken by Seligman [248] but in a slightly different context of logic of correct description. Much of the work on the sat-ND was done by Braüner, in particular in [55]. Below we present three systems: the first two due to the author and the third due to Braüner.

The first two systems are standard in the sense that they are just extensions of standard KM with additional rules. Moreover, in both cases rules of the system are defined on all kinds of formulae (not only on sat-formulae). We propose two versions of standard ND system: the first is closer to axiomatic system, and the second is based on Seligman's SC. Both were constructed with simplicity (in the sense of doing proofs by hand) in mind. In consequence, they are highly redundant.

The third ND is a sat-calculus due to Braüner. It is more elegant and concise, because it was constructed mainly for theoretical purposes. Its set of rules is defined with particular goal in mind – the proof of normalization theorem.

12.4.1 Standard ND-Systems for $\mathbf{K_{H@}}$

For the first ND we take as a basis full KM for standard modal logic $\mathbf{K}$ with $[NEC]$ and $[POS]$ as described in Chapter 6. The set of additional rules is modelled on the axiomatic formulation from the preceding Chapter (Section 11.3). To obtain an ND-system for the basic hybrid logic $\mathbf{K_{H@}}$ we need two additional sets of rules:

3. Inference Hybrid rules:

$(IS\text{-}D)$	$\neg @_i\varphi$ // $@_i\neg\varphi$
$(I\bot @)$	$@_i\bot$ // $\bot$
$(I@I)$	$i,\ \varphi$ / $@_i\varphi$
$(I@E)$	$i,\ @_i\varphi$ / φ
$(I@@E)$	$@_j@_i\varphi$ // $@_i\varphi$
$(IRef)$	$\varnothing$ / $@_i i$
$(I\Diamond E)$	$\Diamond @_i\varphi$ / $@_i\varphi$
$(I\Box I)$	$@_i\varphi$ / $\Box @_i\varphi$
$(I\vdash @E)$	$@_i\varphi$ / φ, provided $\vdash @_i\varphi$ and i is not in φ

4. Hybrid proof construction rules

$[@I]$	If $\Gamma\vdash\varphi$, then $@_i\Gamma\vdash @_i\varphi$, where $@_i\Gamma=\{@_i\varphi:\varphi\in\Gamma\}$
$[@\Box]$	If $\Gamma,\ @_i\Diamond j\vdash @_j\varphi$, then $\Gamma\vdash @_i\Box\varphi$, where j is new in a derivation (hence also not in φ)

In Kalish/Montague format both rules are represented as follows:

	$@_i\Gamma$		Γ
i	SHØW: $@_i\varphi$	i	SHØW: $@_i\Box\varphi$
	Γ	$i+1$	$@_i\Diamond j$
	$\vdots$		Γ
	$\vdots$		$\vdots$
k	φ	k	$@_j\varphi$

One should note that $[@I]$ creates a strict subproof since we must use reiteration on sat-formulae changing them into ordinary (perhaps different sat-formulae) by deletion of $@_i$, whereas $[@\Box]$ makes an ordinary subproof (all formulae from the outer derivation are permitted in the subproof). Hence, on the level of realization we need not only additional rules for closure of a subderivation and for introducing assumption (in $[@\Box]$), but also a modification of reiteration rule allowing φ to be put below S-formula $@_i\psi$ if $@_i\varphi$ is above it. We leave suitable formulation along the lines of Section 6.3 to the reader. Notice however, that this form of reiteration is admissible only on sat-formulae with the same nominal i which is present in the S-formula opening a subderivation.

The set of rules of this system is redundant. $(I@I)$ is interderivable with $(I@E)$ and $(I\Box I)$ with $(I\Diamond E)$ but it is nice to have them in pairs for symmetry. $(I\bot@)$ is derivable, (by $[@I]$ on $\top$, $(IS\text{-}D)$ and $(\bot I)$ in one direction, and by (DS) in the second) but simplifies proofs. Both proof construction rules are necessary if we want to have ND-equivalent of H-$\mathbf{K}^{+}_{\mathbf{H@}}$. But note that $[@\Box]$ is just an ND-realisation of (BG), so it may be omitted if we need only complete ND-formalization of $\mathbf{K}_{\mathbf{H@}}$. The same applies to $(I\vdash @E)$ which is a counterpart of axiomatic $(NAME)$.

Although the set of rules is redundant, it does not include the rules corresponding to symmetry of @, or to some forms of Leibniz' rule (like $(@Nom)$ in Blackburn's SC); rules like $(I@E)$ and $(I@I)$ are sufficiently strong to make them derivable (cf. with the axiomatic formulation of $\mathbf{K}_{\mathbf{H@}}$ in Chapter 11).

Below we present two proofs as an illustration of how this system works. The first is a proof of $Nom2$, the second should be compared with $(B\Box E)$ in Braüner's system.

1	SHØW: $@_i j \land @_j p \to @_i p$	$[6, COND]$
2	$@_i j \land @_j p$	*ass.*
3	$@_i j$	$(2, \alpha E)$
4	$@_j p$	$(2, \alpha E)$
5	$@_i @_j p$	$(4, I@@E)$
6	SHØW: $@_i p$	$[9, @I]$
7	$@_j p$	$(5, Reit.)$
8	j	$(3, Reit.)$
9	p	$(7, 8, I@E)$

1	SHØW: $@_i \Box p \land @_i \Diamond j \to @_j p$	$[13, COND]$
2	$@_i \Box p \land @_i \Diamond j$	*ass.*
3	$@_i \Box p$	$(2, \alpha E)$
4	$@_i \Diamond j$	$(2, \alpha E)$
5	SHØW: $@_i @_j p$	$[12, @I]$
6	$\Box p$	$(3, Reit.)$
7	$\Diamond j$	$(4, Reit.)$
8	SHØW: $\Diamond @_j p$	$[11, POS]$
9	j	*mod. ass.*
10	p	$(6, Reit.)$
11	$@_j p$	$(9, 10, I@I)$
12	$@_j p$	$(8, I\Diamond E)$
13	$@_j p$	$(5, I@@E)$

It is quite an easy task to prove completeness of this system with respect to $\mathbf{K_{H@}}$. Since ordinary modal basis for $\mathbf{K}$ is provided by resources of KM-$\mathbf{K}$, it is sufficient to prove all the axioms given in Section 11.3 which is immediate: $@K$ is provable by $[@I]$, $Selfdual@$ by $(IS\text{-}D)$, e.t.c. As for the rules of H-$\mathbf{K_{H@}}$, $(RG@)$ is simulated by $[@I]$ with empty Γ and (BG) by $[@\Box]$. Hence, the pure completeness theorem stated for H-$\mathbf{K^+_{H@}}$ holds also for the present ND-system.

To establish soundness one must prove that all inference rules are $\mathbf{K_{H@}}$-normal and that proof construction rules preserve $\mathbf{K_{H@}}$-normality. We will show the case of $[@\Box]$:

PROOF Assume that $\Gamma,\ @_i\Diamond j \models @_j\varphi$ but $\Gamma \not\models @_i\Box\varphi$, hence, for some w, $w \vDash \Gamma$ and $w \nvDash @_i\Box\varphi$, i.e. $V(i) \nvDash \Box\varphi$. So there is w' such that $V(i)\mathcal{R}w'$ and $w' \nvDash \varphi$. Since j is arbitrary we may assume $\{w'\} = V(j)$, so $@_i\Diamond j$ holds in every world, in particular $w \vDash @_i\Diamond j$, which implies $w \vDash @_j\varphi$. But then $V(j) \vDash \varphi$ and we have a contradiction. ■

We are entitled to state:

Theorem 12.2 *KM-$\mathbf{K^+_{H@}}$ in the first version is adequate for $\mathbf{K^+_{H@}}$*

Moreover, this result applies to all extensions obtained by means of pure axioms. As a result, the extension to many stronger modal logics may be done in two ways. The first runs as in KM for standard modal logics, by modifying reiteration rule (see Chapters 6 and 7). But in the light of our pure completeness results we may obtain much more uniform ND formalization of several hybrid logics by addition of new inference rules modelled on pure axioms. Interesting and very general solution of this kind due to Braüner will be considered in the next subsection. Also the question of extension of this system to stronger languages (e.g. with $\downarrow$ or global modalities) does not generate any problems. We can add suitable rules from Braüner's system or from Blackburn's TS.

The second choice in defining ND for hybrid logic is to follow SC of Seligman. This time we need as a basis only KM-**CPL** from Chapter 2. Let us first display the additional proof construction rules:

$[NAME]$	If $\Gamma, i \vdash \varphi$, then $\Gamma \vdash \varphi$, where both Γ and φ are sat-formulae
$[\Box]$	If $\Gamma,\ \Diamond i \vdash @_i\varphi$, then $\Gamma \vdash \Box\varphi$, where i is new in the proof

In Kalish/Montague format they are represented as follows:

	Γ	
i	SHØW: φ	
$i+1$		i
		Γ
		$\vdots$
k		φ

	Γ	
i	SHØW: $\Box\varphi$	
$i+1$		$\Diamond i$
		Γ
		$\vdots$
k		$@_i\varphi$

One can easily note that $[NAME]$ covers $(TERM)$ from Seligman's SC, whereas $[\Box]$ corresponds to his $(H \Rightarrow \Box)$ (and $(H\Diamond \Rightarrow)$ by interdefinability). Once again, $[NAME]$ is a strict subderivation creating rule, whereas $[\Box]$ introduces an ordinary subderivation. Note however, that in contrast to $[@I]$, the requirements concerning $[NAME]$ are not so strict. First, sat-formulae are simply reiterated into a subderivation without deletion of sat-operators. It is also not demanded that all reiterated sat-formulae and S-formula have the same nominal as an argument of sat-operator. Also, a nominal assumption under S-line may be any nominal we need. The introduction of this assumption under S-line containing sat-formula is optional, but if we want to close this subderivation by $[NAME]$ its presence is obligatory. We leave the precise statement of these rules on the level of realization to the reader.

These proof construction rules are deductively so strong that not only we can get rid of $[NEC]$ but also the set of inference rules proposed for the former system may be limited to: $(IS\text{-}D), (I@I), (I@E)$. In fact, for completeness one half of $(IS\text{-}D)$ (from left to right), and one of interderivable $(I@I), (I@E)$ is enough (cf. remarks above). On the other hand, we need at least one rule from the pair of additional (interderivable) rules:

$(I\neg\Diamond E)$ $\quad \neg\Diamond\varphi,\ \Diamond j \ /\ @_j\neg\varphi$
$(I\Box E)$ $\quad \Box\varphi,\ \Diamond j \ /\ @_j\varphi$

Clearly, to obtain a formalization of $\mathbf{K}^{+}_{\mathbf{H@}}$ we must add $(I \vdash @E)$ (for simulation of $(NAME)$) as well, or to introduce a stronger form of $[NAME]$ instead:

$[NAME']$ If $\Gamma, i \vdash \varphi$, then $\Gamma \vdash \varphi$, where either both Γ and φ are sat-formulae or i is new.

This proof construction rule amalgamate both Seligman's $(TERM)$ and $(S\text{-}NAME)$. On the other hand, $[\Box]$ is needed not only to simulate (BG) but is indispensable for completeness even if we want to capture $\mathbf{K_{H@}}$ only.

Although the proposed rules are modelled on SC rules of Seligman's system we have no counterpart of (N_1). Such a global rule is not easy to obtain for ND-system like this but the system is already complete for $\mathbf{K}^+_{\mathbf{H@}}$ without it as we will show. One may doubt about this when comparing $[\Box]$ with $[\Box @]$ which was the exact counterpart of (BG). But the latter rule is easily proved admissible in the current system. Namely, every application of $[\Box @]$ in a proof may be substituted by the subproof using $[NAME]$ and $[\Box]$, as the following schema shows:

1	SHØW: $@_i\Box\varphi$	$[8, NAME]$
2	i	*ass.*
3	SHØW: $\Box\varphi$	$[7, \Box]$
4	$\Diamond j$	*ass.*
5	$@_i\Diamond j$	$(2, 4, I@I)$
6	$\vdots$	
7	$@_j\varphi$	(5 by assumption)
8	$@_i\Box\varphi$	$(2, 3, I@I)$

One may establish completeness of this system for $\mathbf{K}^+_{\mathbf{H@}}$ by proving all axioms and showing admissibility of all rules of H-$\mathbf{K}^+_{\mathbf{H@}}$. This time proofs are more involved and we must additionaly prove K and demonstrate admissibility of (RG) (since we do not have $[NEC]$). We will prove in detail K, one half of *Agree*, and admissibility of (RG), leaving the rest to the reader (but with some hints how to do it).

As for the simulation of primitive rules of H-$\mathbf{K}^+_{\mathbf{H@}}$ one may read off from the proof schema given above how to replace the applications of (BG) by $[NAME]$ and $[\Box]$. For (RG) let $[\mathcal{D}]$ be a proof of φ, then it may be transformed into the proof of $\Box\varphi$ in the following way:

1	SHØW: $\Box\varphi$	$[3, \Box]$
2	$\Diamond i$	*ass.*
3	SHØW: $@_i\varphi$	$[k, NAME]$
4	i	(*nom.ass.*)
5	SHØW: φ	$[by\mathcal{D}]$
	$\mathcal{D}$	*by hyp.*
k	$@_i\varphi$	$(4, 5, I@I)$

By the way, the above schema shows also in lines $3 - k$ how $(RG@)$ is simulated in our ND. The case of (MP) and $(Sub@)$ is obvious. $(NAME)$ is covered either by $(I \vdash @E)$ or by $[NAME']$ with Γ empty and an application of $(I@E)$. To prove $Ref@$ we need only $[NAME]$ and $(I@I)$. One half of $Selfdual$ is provided by $(IS\text{-}D)$, whereas the other is proved by $[NAME]$ and $(I@E)$. One half of $Agree$ is proved below, the other needs $[NAME]$ and $(I@E)$.

1	SHØW: $@_jp \to @_i@_jp$	$[3, COND]$
2	$@_jp$	$ass.$
3	SHØW: $@_i@_jp$	$[10, RED]$
4	$\neg @_i@_jp$	$(ass.)$
5	SHØW: $\neg @_jp$	$[9, NAME]$
6	i	$nom.\ ass.$
7	$\neg @_i@_jp$	$(4, Reit.)$
8	$@_i\neg @_jp$	$(7, IS\text{-}D)$
9	$\neg @_jp$	$(6, 8, I@E)$
10	$\bot$	$(2, 5, DS)$

$Intro@$ is proved directly by $(I@I)$, whereas for $Back$ we need $[\Box]$ and $Agree$ to simplify proof (or repeat the above proof inside). Finally, K may be proved in this way:

1	SHØW: $\Box(p \to q) \to (\Box p \to \Box q)$	$[19, RED]$
2	$\neg(\Box(p \to q) \to (\Box p \to \Box q))$	$ass.$
3	$\Box(p \to q)$	$(2, \alpha E)$
4	$\neg(\Box p \to \Box q)$	$(2, \alpha E)$
5	$\Box p$	$(4, \alpha E)$
6	$\neg\Box q$	$(4, \alpha E)$
7	SHØW: $\Box q$	$[11, \Box]$
8	$\Diamond i$	$(ass.)$
9	$@_i(p \to q)$	$(3, 8, I\Box E)$
10	$@_ip$	$(5, 8, I\Box E)$
11	SHØW: $@_iq$	$[18, NAME]$
12	i	$nom.\ ass.$
13	$@_i(p \to q)$	$(9, Reit.)$
14	$@_ip$	$(10, Reit.)$
15	$p \to q$	$(12, 13, I@E)$
16	p	$(12, 14, I@E)$
17	q	$(15, 16, \beta E)$
18	$@_iq$	$(12, 17, I@I)$
19	$\bot$	$(6, 7, DS)$

Note that this proof in lines 8–18 provides also a proof of ($RG@$) and we are done. Hence, by completeness of H-$\mathbf{K}^+_{\mathbf{H@}}$ we obtain a completeness of this ND system with respect to $\mathbf{K}^+_{\mathbf{H@}}$.

Soundness proof is easier and partly follows from the one obtained for the first version. It is enough to prove normality of ($I\Box E$) or ($I\neg\Diamond E$) and normality preservation of proof construction rules. For [$\Box$] the proof is analogous as for [$@\Box$]; we demonstrate the case of [$NAME$]:

Proof Assume that $\Gamma, i \models \varphi$ but $\Gamma \not\models \varphi$, hence, for some w, $w \vDash \Gamma$ and $w \nvDash \varphi$. Since all elements of Γ are sat-formulae, then they are satisfied in all worlds, in particular in $V(i)$. Hence $V(i) \vDash \Gamma$ and $V(i) \vDash i$ which implies $V(i) \vDash \varphi$. But this leads to contradiction since φ is also sat-formula and must be satisfied in all worlds, in particular in w. ■

As a result we have:

Theorem 12.3 *KM-*$\mathbf{K}^+_{\mathbf{H@}}$ *in the second version is adequate for* $\mathbf{K}^+_{\mathbf{H@}}$

For practical purpose (to make simpler proofs) in the second version one may of course use all other inference rules stated above for the first version; they are all derivable. In fact, we may just combine them both in one very redundant system since [$@I$] and [NEC] is also admissible in the second version. Note that if we want to have inference rules corresponding to ($H\Diamond \Rightarrow$) and ($H \Rightarrow \Box$), instead of [$\Box$], then the following pair will work:

($I\Diamond E'$) $\Diamond\varphi$ / $\Diamond i$, $@_i\varphi$, where i is new
($I\neg\Box E$) $\neg\Box\varphi$ / $\Diamond i$, $@_i\neg\varphi$, where i is new

The extensions to other logics or stronger languages may be obtained in the same way as was stated for the first version.

12.4.2 Braüner's ND-System

As we already remarked, ND-system of Braüner is an example of sat-calculus and shows many similarities to Blackburn's SC and tableau calculus. It is a great value of this system that it represents a uniform formalization of a wide class of logics. Before we specify what kind of strengthenings is dealt with, we present a basic system for $\mathbf{K}_{\mathbf{H@}}$.

1. Inference rules

$(B\wedge E)$	$@_i(\varphi\wedge\psi)\ /\ @_i\varphi,\ @_i\psi$
$(B\wedge I)$	$@_i\varphi,\ @_i\psi\ /\ @_i(\varphi\wedge\psi)$
$(B\to E)$	$@_i(\varphi\to\psi),\ @_i\varphi\ /\ @_i\psi$
$(B\bot I)$	$@_i\bot\ /\ @_j\bot$
$(B@I)$	$@_i\varphi\ /\ @_j@_i\varphi$
$(B@E)$	$@_j@_i\varphi\ /\ @_i\varphi$
$(BRef)$	$\varnothing\ /\ @_ii$
$(BNom_1)$	$@_ij,\ @_i\varphi\ /\ @_j\varphi$, where $\varphi\in AT$
$(BNom_2)$	$@_ij,\ @_i\Diamond k\ /\ @_j\Diamond k$
$(B\Box E)$	$@_i\Box\varphi,\ @_i\Diamond j\ /\ @_j\varphi$

2. Proof construction rules

$[@COND]$	If $\Gamma,\ @_i\varphi\vdash @_i\psi$, then $\Gamma\vdash @_i(\varphi\to\psi)$
$[@RAA]$	If $\Gamma,\ @_i\neg\varphi\vdash @_i\bot$, then $\Gamma\vdash @_i\varphi$, where $\varphi\in AT$
$[B\Box]$	If $\Gamma,\ @_i\Diamond j\vdash @_j\varphi$, then $\Gamma\vdash @_i\Box\varphi$, where j is not in φ or in any undischarged assumption in Γ

Since rules of Braüner differ remarkably both from SC sat-calculus of Blackburn and from axiomatic formulation, we make some comments on them.[6] It is obvious that for booleans we have ordinary ND-rules but with $@_i$ as added context; it applies to inference rules, and to proof construction rules as well. Since the proof of normalization theorem is the main goal of Braüner, the calculus is $\neg$-free, with $\bot$ instead; $\neg$ and $\Diamond$ are used in proof-schemata as obvious definitional shorthands. Moreover, note that in this system $\bot$ is treated locally (with $@_i$ added), that's why Braüner needs $(B\bot I)$ as a kind of (inconsistency) propagation rule, necessary to perform $[@RAA]$.

As for hybrid rules, $[B\Box]$ is essentially the same as $[@\Box]$ and – as we remarked earlier – it is just ND-form of (BG). $(BRef), (B@I)$ and $(B@E)$ are direct counterparts of axioms $Ref@$ and $Agree$ from Section 11.3, but note that $(BNom_2)$ is not the same as a thesis $Nom2$ and it is also different from $Bridge$ (although $(BNom_1)$ is just a thesis $Nom1$). A comparison with Blackburn's TS shows that, despite appearances, it is not that $(BNom_1)$ corresponds to Blackburn's $(BNom)$ and $(BNom_2)$ to $(BBridge)$, although $(BNom_1)$ is the same as $(BNom1)$ from [44]. In fact, both Braüner's rules $(BNom_1)$ and $(BNom_2)$ may be covered by one general rule:

[6] On the other hand, one may note that some of the rules are identical to Blackburn's rules; in this case we use the same names for Braüner's and tableau rules.

$(BNom')$ $@_i j,\ @_i\varphi\ /\ @_j\varphi$

which is interderivable (by selfduality of @) with Blackburn's:

$(BNom)$ $@_i j,\ @_j\varphi\ /\ @_i\varphi$

On the other hand, to derive $(BBridge)$ we need a rule $(B\Box E)$ (the same as in Blackburn's TS) which in fact plays a role of axiom *Back*.

Regarding extensions, one should note first that Braüner provided normalization theorem for ND-system adequate not only for special extensions over $\mathbf{K_{H@}}$ but also over $\mathbf{K_{H@\downarrow}}$ and $\mathbf{K_{H@\forall}}$. The rules for $\downarrow$ and $\forall$ are the following:

$(B\downarrow E)$	$@_i \downarrow u\varphi,\ @_i j\ /\ @_j\varphi[u/j]$
$[B\downarrow I]$	if $\Gamma,\ @_i j \vdash @_j\varphi[u/j]$, then $\Gamma \vdash @_i \downarrow u\varphi$, where j is not in φ or in any undischarged assumption in Γ
$(B\forall E)$	$@_i\forall u\varphi\ /\ @_i\varphi[u/j]$
$(B\forall I)$	$@_i\varphi[u/j]\ /\ @_i\forall u\varphi$, where j is not in $@_i\forall u\varphi$ or any undischarged assumption

In fact, the formulation of rules for binders in Braüner's system is a bit different since he uses only state variables, so ordinary conditions concerning proper substitution of variables must be satisfied and side conditions forbid only free occurences of substituted variable which in our formulation is just a nominal j.

Now we consider what kind of extensions is considered by Braüner. He has proved a general completeness theorem (and general normalization theorem) for all logics (in one of the hybrid languages $\mathbf{L_{H@}}, \mathbf{L_{H@\downarrow}}, \mathbf{L_{H@\forall}}$) whose classes of frames (i.e. accessibility conditions) are expressed by geometric theories (see [286] and Section 1.1.5). For reader's convenience we recall here the schema of the basic geometric formula:

$$(bgf)\quad \forall x_1, ..., x_k(\varphi_1 \wedge ... \wedge \varphi_n \to \exists y_1, ..., y_l(\psi_1 \vee ... \vee \psi_m)),$$

where $k \geq 1, l, n, m \geq 0$, each φ_i is an atom and each ψ_i is an atom or finite conjunction of atoms.

Every case of (bgf) corresponds in hybrid language via HT-translation function to ND-rule of the following form:

$[BGR]$ If $\Gamma_1, \Psi_1 \vdash \chi, ..., \Gamma_m, \Psi_m \vdash \chi$, then $\Delta, \Gamma_1, ..., \Gamma_m, \varphi'_1, ..., \varphi'_n \vdash \chi$, where $k \geq 1, l, n, m \geq 0$, each $\varphi'_i = HT(\varphi_i)$, each Ψ_i is a set of HT-translations of atoms that make together a conjunction ψ_i, and no nominal corresponding to y_i occurs in $\chi, \Gamma_1 - \Gamma_m, \Delta, \varphi'_1 - \varphi'_n$.

This rather complicated general characteristics may become clearer if we take a look at some examples. For instance, conditions of symmetry, asymmetry, antisymmetry, transitivity, irreflexivity belong to this category. Notice, that in case of irreflexivity and asymmetry, because of the lack of negation, we have in mind the following formulae:

$\forall x(\mathcal{R}xx \to \bot)$
$\forall xy(\mathcal{R}xy \wedge \mathcal{R}yx \to \bot)$

This is not the whole story – we note the following cases of bgf-s:

- every instance of Geach axiom, in particular Church-Rosser property
- every Horn clause

As for Horn clauses, just note that it is a bgf with $l = 0, m = 1$ and ψ_1 being an atom. In case of Horn clauses the schema of the corresponding rule may be simplified:

(HR) $\varphi_1, ..., \varphi_n$ / ψ_1

Notice, that this result is a generalization of that obtained by Basin, Matthews and Vigano [21], since their labelled ND-system covers only logics axiomatized by Horn clauses. One should also note, that not every case of (bgf) is expressible by pure formula unless $\forall$ is present. So in weaker hybrid languages Braüner's completeness theorem covers some logics not captured by pure completeness theorem.

The result of Braüner is similar to that of Demri [78] mentioned earlier. His general rule corresponds to the class of restricted Π^0_2-formulae of the form:

$$\forall x_1, ..., x_k \exists y_1, ..., y_l(\varphi_1 \wedge ... \wedge \varphi_n \to (\psi_1 \vee ... \vee \psi_m)),$$

where $k \geq 1, l, n, m \geq 0$, each φ_i is a literal (atom in the sense defined above or its negation) with variables only from $\{x_1, ..., x_k\}$ and each ψ_i is a literal or finite conjunction of literals.

This class of formulae also includes Horn formulae and is equivalent to the class of all first-order formulae which are primitive in the sense of Kracht [165].

We finish this discussion with two examples of rules corresponding to concrete bgf-s: antisymmetry and Church-Rosser:

$[ANTISYM]$	If $\Gamma, @_i j \vdash \chi$, then $\Delta, \Gamma, @_i\Diamond j, @_j\Diamond i \vdash \chi$
$[C\text{-}R]$	If $\Gamma, @_j\Diamond l, @_k\Diamond l \vdash \chi$, then $\Delta, \Gamma, @_i\Diamond j, @_i\Diamond k \vdash \chi$ where l is not in χ, Γ, Δ

The first of them may be simplified to inference rule, since it is an instance of a Horn clause:

$(Antisym)$ $@_i\Diamond j, @_j\Diamond i \ / \ @_i j$

Braüner uses tree-format of proof-representation, since it is well behaved with respect to proving normalization theorem. But of course we can display proofs in his system as Kalish-Montague style proofs. Here is an example:

1	SHØW: $@_i(\Diamond j \wedge @_j p \rightarrow \Diamond p)$	$[6, @COND]$
2	$@_i(\Diamond j \wedge @_j p)$	*ass.*
3	$@_i\Diamond j$	$(2, B\wedge E)$
4	$@_i @_j p$	$(2, B\wedge E)$
5	$@_j p$	$(4, B@E)$
6	SHØW: $@_i\Diamond p$	$[10, @RAA]$
7	$@_i\Box(p \rightarrow \bot)$	*ass.*
8	$@_j(p \rightarrow \bot)$	$(3, 7, B\Box E)$
9	$@_j\bot$	$(5, 8, B \rightarrow E)$
10	$@_i\bot$	$(9, B\bot I)$

Clearly, in general, one encounters similar problems with realization of $[BGR]$ in KM format as with other proof construction rules which close many subderivations in one step (cf. Section 6.5). But as we will see, in hybrid form of RND this may be easily overcome.

Results of Braüner may be transferred to SC; in fact, Braüner himself did it. On the basis of his ND-system he defines cut-free sat SC similar to that of Blackburn. The differences are the following:

- one more axiom of the form: $@_i\bot, \Gamma \Rightarrow \Delta$
- a rule for $(\Box \Rightarrow)$ like in Seligman's calculus with 2 premises

- different special rules

As for the last point: $(@Ref)$ is the same, but instead of $(@Sym), (@Nom)$ and $(@Bridge)$ there are two SC counterparts of $(BNom_1)$ and $(BNom_2)$:

$$(BNom_1 \Rightarrow)^1 \quad \frac{\Gamma \Rightarrow \Delta, @_i j \qquad \Gamma \Rightarrow \Delta, @_i \varphi}{\Gamma \Rightarrow \Delta, @_j \varphi}$$

$$(BNom_2 \Rightarrow) \quad \frac{\Gamma \Rightarrow \Delta, @_i j \qquad \Gamma \Rightarrow \Delta, @_i \Diamond k \qquad @_j \Diamond k, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

1. where $\varphi \in ATOM$

For $\downarrow$ and $\forall$ the rules are as follows:

$$(B \downarrow\Rightarrow) \quad \frac{\Gamma \Rightarrow \Delta, @_i j \qquad @_j \varphi[u/j], \Gamma \Rightarrow \Delta}{@_i \downarrow u\varphi, \Gamma \Rightarrow \Delta} \qquad (B \Rightarrow\downarrow)^1 \quad \frac{@_i j, \Gamma \Rightarrow \Delta, @_j \varphi[u/j]}{\Gamma \Rightarrow \Delta, @_i \downarrow u\varphi}$$

$$(B\forall\Rightarrow) \quad \frac{@_i \varphi[u/j], \Gamma \Rightarrow \Delta}{@_i \forall u\varphi, \Gamma \Rightarrow \Delta} \qquad (B \Rightarrow\forall)^1 \quad \frac{\Gamma \Rightarrow \Delta, @_i \varphi[u/j]}{\Gamma \Rightarrow \Delta, @_i \forall u\varphi}$$

1. j does not occur in the conclusion.

SC schema of rules for (bgf) is of the form:

$$(BGR) \quad \frac{\Gamma \Rightarrow \Delta, \varphi'_1 \quad \dots \quad \Gamma \Rightarrow \Delta, \varphi'_n \qquad \Psi_1, \Gamma \Rightarrow \Delta \quad \dots \quad \Psi_m, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

where $k \geq 1, l, n, m \geq 0$, each $\varphi'_i = HT(\varphi_i)$, each Ψ_i is a set of HT-translations of atoms that form conjunction ψ_i and no nominal that corresponds to y_i occurs in $\Gamma_1 - \Gamma_m, \Delta, \varphi'_1 - \varphi'_n$.

This calculus satisfies the following quasi-subformula property:

Every formula occuring in a derivation of $\Gamma \Rightarrow \Delta$ is a quasi-subformula of elements in $\Gamma \cup \Delta$ or a quasi-subformula of some $@_i j$ or $@_i \Diamond j$, where $@_i \varphi$ is a quasi-subformula of $@_j \psi$ iff φ is an ordinary subformula of ψ.

Clearly, his ND-systems also satisfies this property for normal proofs, but the formulation in this context is more complicated, so we address the reader to [55] for details.

Finally, we should note that the possibility of defining such a uniform calculus is essentially dependent on the capability of hybrid languages to express such things like state identity, state succession and internalization of satisfaction statements. For ordinary modal languages it is possible only if we use strong labelling like in [21] or [237].

12.5 Resolution

On the field of application of resolution to modal logics, hybrid languages seem to offer far reaching simplification due to the machinery of nominals and sat-operators. But the presence of @ is essential; without sat-operators we are unable to take a formula out of the scope of modal operator. Because of that, the two resolution systems for hybrid logics presented below belong to the group of sat-calculi, i.e. both are defined on clauses containing only sat-formulae. On the other hand, clauses are in generalized form; they contain not only literals prefixed with $@_i$ but any sat-formulae.

First of these calculi was constructed by Areces, de Nivelle, de Rijke and Heguiabehere [10] and later implemented [11] under the name HyLoRes. The main motivation was to find an efficient reasoning system useful for automated theorem proving. Recent form of this system applies many optimization techniques discovered for first-order resolution, like ordering and selection functions (see [12]), and is still improved.

The second system presented in this section is the final generalization of our RND system. In contrast to HyLoRes it was not constructed with automated deduction in mind but rather for obtaining a general, user's friendly framework, open for free application of several proof techniques.

12.5.1 HyLoRes

Resolution system constructed by Areces and others [10] is an effective system defined for automated deduction. Its implementation is called HyLoRes and may be download from hybrid logic web page. For convenience we will apply the name of a prover to the system as well. The version presented below is taken from [10], where it is embedded in a more general setting of labelled resolution containing systems for ordinary modal logics (**K** and some of its extensions), description logic **ALCR** and hybrid $\mathbf{K_{H\downarrow @}}$.

As we already remarked the system does not operate on ordinary clauses obtained as a result of previous transformation to normal form. But the formulae in clauses are assumed to be in negation normal form by the application of the following rewriting procedure nf:

$$\begin{aligned} nf(\neg\neg\varphi) &= \varphi \\ nf(\Diamond\varphi) &= \neg\Box\neg\varphi \\ nf(\varphi\vee\psi) &= \neg(\neg\varphi\wedge\neg\psi) \end{aligned}$$

As a result, special rules for negation are dispensable. The rules of the basic version for $\mathbf{K_{H@}}$ are the following:

$(@Res)$	$\Gamma, @_i\varphi \; ; \; \Delta, @_i\neg\varphi \; / \; \Gamma, \Delta$
$(C@\wedge)$	$\Gamma, \; @_i(\varphi \wedge \psi) \; / \; \Gamma, \; @_i\varphi \; ; \; \Gamma, \; @_i\psi$
$(C@\neg\wedge)$	$\Gamma, \; @_i\neg(\varphi \wedge \psi) \; / \; \Gamma, \; @_i nf(\neg\varphi), \; @_i nf(\neg\psi)$
$(C@\neg\Box)$	$\Gamma, @_i\neg\Box\varphi \; / \; \Gamma, @_i\neg\Box\neg j \; ; \; \Gamma, @_j nf(\neg\varphi)$, where j is a new nominal
$(C@\Box)$	$\Gamma, @_i\neg\Box\neg j \; ; \; \Delta, @_i\Box\varphi \; / \; \Gamma, \Delta, @_j\varphi$
$(C@)$	$\Gamma, @_i@_j\varphi \; / \; \Gamma, @_j\varphi$
$(C@Ref)$	$\Gamma, @_i\neg i \; / \; \Gamma$
$(C@Sym)$	$\Gamma, @_i j \; / \; \Gamma, @_j i$
$(C@Param)$	$\Gamma, @_i j \; ; \; \Delta, \varphi \; / \; \Gamma, \Delta, \varphi[i//j]$

Note the similarity of these rules to the rules of Blackburn's tableau calculus. In fact, most of the tableau rules are just special forms of these resolution rules with $\Gamma = \Delta = \varnothing$ and with obvious differences of β-rules, since here we have no branching but just transformation of a clause. The only important differences concern $(C@Ref)$ and $(C@Param)$ (from paramodulation). The latter covers Blackbourn's rules $(BNom)$ and $(BBridge)$ but in the more general form. $(C@Ref)$ obviously in both systems covers reflexivity of the identity relation between nominals, but note that the present form has genuinely resolution character (deletion, not addition, of a suitable formula). Needless to say that $(C@Sym)$ is derivable, as in other calculi with the similar set of rules for nominals.

Essential similarities of rules in non-clausal forms of resolution systems to tableau expansion rules are rather unavoidable since resolution steps are interleaved with simplification steps. It is more natural and simpler to use resolution on any formulae, not only on literals, especially in the context of modal logics.

In fact, essential resolution steps are connected not only with the application of $(@Res)$ and $(C@Ref)$. Closer analysis of $(C@\Box)$ also shows that it is a kind of a resolution rule. If we apply standard translation we can see that it is an ordinary resolution on $\mathcal{R}ij$ and $\neg\mathcal{R}ix$ with unification on x. One can ask if it is possible to define more rules that are resolution-like rather than tableau-like. For example, instead od $(C@Sym)$ we can use:

$(C@Sym')$ $\Gamma, @_i j, \; \Delta, \neg @_j i \; / \; \Gamma, \Delta$

We return to this question in a more detailed way after the presentation of the second system. One should also observe that $(C@\neg\Box)$ is a kind of skolemization but limited to introduction of constants only.

A deduction of a clause Γ from a set of clauses X ($X \vdash \Gamma$) in HyLoRes is defined as a finite sequence of sets of clauses $X_1, ..., X_n$, where $X_1 = X, \Gamma \in X_n$ and each $X_i, 1 < i$ consists of the set of clauses obtained by the application of one of the rules to X_{i-1}. If $X_n = \bot$ then we have a refutation of X. Obviously, the proof of φ is a refutation of $\{@_i nf(\neg\varphi)\}$, where $i \notin \varphi$, exactly as in other sat-calculi.

Since HyLoRes is a universal proof system we can use it also for constructing falsifying models (model extraction from nonsuccessful refutations). [10] contains a constructive completeness proof of resolution system for description logic which applies with small modifications to the system above and from which a suitable decision procedure may be obtained.

HyLoRes may be extended also to undecidable $\mathbf{K_{H@\downarrow}}$ just by adding one rule:

$(C@\downarrow)$ $\quad \Gamma, @_i {\downarrow} u\varphi \ / \ \Gamma, @_i\varphi[u/i]$

Below we reproduce an example of a proof. Let our $\varphi := {\downarrow} u\Diamond(u \wedge p) \to p$ which is a thesis of $\mathbf{K_{H@\downarrow}}$. Then $\{@_i nf(\neg\varphi)\} := \{@_i(({\downarrow} u\Diamond(u \wedge p)) \wedge \neg p)\}$. The proof in HyLoRes looks like that:

1	$@_i(({\downarrow} u\neg\Box\neg(u \wedge p)) \wedge \neg p)$	
2	$@_i({\downarrow} u\neg\Box\neg(u \wedge p))$; $@_i\neg p$	$(C@\wedge)$
3	$@_i(\neg\Box\neg(i \wedge p))$; $@_i\neg p$	$(C@\downarrow)$
4	$@_i\neg\Box\neg j$; $@_j(i \wedge p)$; $@_i\neg p$	$(C@\neg\Box)$
5	$@_j i$; $@_j p$; $@_i\neg p$	$(C@\wedge)$
6	$@_i p$; $@_i\neg p$	$(C@Param)$
7	$\bot$	$(@Res)$

No extension to other logics over $\mathbf{K_{H@}}$ or $\mathbf{K_{H@\downarrow}}$ is considered but three such rules are presented for labelled resolution system for ordinary modal logic that may be applied also in the hybrid setting so we display suitable transformations below:

$(C@T)$ $\quad \Gamma, @_i\Box\varphi \ / \ \Gamma, @_i\varphi$
$(C@D)$ $\quad \Gamma, @_i\Box\varphi \ / \ \Gamma, @_i\neg\Box nf(\neg\varphi)$
$(C@4)$ $\quad \Gamma, @_i\Box\varphi j \ ; \ \Delta, @_i\neg\Box\neg j \ / \ \Gamma, \Delta, @_j\Box\varphi$

Notice, that these rules do not correspond to pure axioms. Moreover, $(C@4)$ introduces the risk of a loop, so procedure from completeness proof must be modified accordingly in order to save termination. We will turn to the problem of extension to stronger logics in the next subsection.

In the actual form of HyLoRes, several constraints on the applicability of rules are involved that serve to increase its efficiency. Since a discussion of advanced optimization techniques is beyond the scope of this text, conditions on selection functions etc. are omitted in our simplified presentation.

12.5.2 HRND – Hybrid RND-System

Once again we return to RND system. Now, we will consider the possibility of extending its application to hybrid logics. In the first place we present a sat-calculus for basic hybrid logic. Next we will pay an attention to the problem of extensions to stronger hybrid logics.

HRND sat-calculus defined on generalised clauses consists of:

1. Sat-versions of classical RND inference rules

$(@W)$	$\Gamma \;/\; \Gamma, @_i\varphi$
$(C@\neg)$	$\Gamma, \neg @_i\varphi \;//\; \Gamma, @_i\neg\varphi$
$(@Res)$	$\Gamma, @_i\varphi \;;\; \Gamma, @_i - \varphi \;/\; \Gamma$
$(C@NN)$	$\Gamma, @_i\neg\neg\varphi \;//\; \Gamma, @_i\varphi$
$(C@\alpha)$	$\Gamma, @_i\alpha \;//\; \Gamma, @_i\alpha_1 \;;\; \Gamma, @_i\alpha_2$
$(C@\beta)$	$\Gamma, @_i\beta \;//\; \Gamma, @_i\beta_1, @_i\beta_2$

2. Modal inference rules

$(C@\pi)$	$\Gamma, @_i\pi^i \;/\; \Gamma, @_i\Diamond j \;;\; \Gamma, @_j\pi$, where j is a new nominal in a derivation
$(C@\nu)$	$\Gamma, @_i\nu^i \;;\; \Delta, @_i\Diamond j \;/\; \Gamma, \Delta, @_j\nu$
$(C@)$	$\Gamma, @_i@_j\varphi \;//\; \Gamma, @_j\varphi$
$(C@Ref)$	$\Gamma, @_i\neg i \;/\; \Gamma$
$(C@Sym)$	$\Gamma, @_ij \;/\; \Gamma, @_ji$
$(C@Nom)$	$\Gamma, @_ij \;;\; \Delta, @_j\varphi \;/\; \Gamma, \Delta, @_i\varphi$
$(C@Bridge)$	$\Gamma, @_ij \;;\; \Delta, @_k\Diamond i \;/\; \Gamma, \Delta, @_k\Diamond j$

Similarly as in the basic form of RND defined in Chapter 4, we need only one proof construction rule $[SUB]$. It is defined exactly as for **CPL** but note that every φ in the schema is a sat-formula and X is a set of generalized clauses built up from sat-formulae only.

One may easily note that this form of RND is much simpler than MRND from Chapter 7 or two forms of labelled RND introduced in Chapter 8. The set of rules is similar to that of HyLoRes, but we admit also building up rules (// in some rules instead of /). It is because we want to have a system of more general character than typical analytic systems like tableau calculi. This generality is needed for a system which may be easily tailored in order to simulate different proof systems, including in particular ND.

Soundness theorem is proven similarly as for other forms of RND but by reference to hybrid models defined in the preceding Chapter. Completeness of HRND will be shown by simulation of Braüner's ND sat-system.

Lemma 12.2 *Inference rules of Braüner's ND-system are derivable in HRND*

$(B\wedge E), (B\wedge I), (B@I), (B@E), (B\Box E)$ are just special cases of $(C@\alpha)$, $(C@)$ and $(C@\nu)$, where $\Gamma = \Delta = \varnothing$). Braüner's $(BRef)$ is derived easily by our version of the rule, and his $(B \to E)$ by $(C@\beta)$, $(@Res)$ and $(@W)$. Both $(BNom_1)$ and $(BNom_2)$ have simple and similar proofs. Below we display one of them:

1	SHØW: $\neg @_i j,\ \neg @_i \Diamond k,\ @_j \Diamond k$	$[5, SUB]$
2	$@_i j$	*ass*
3	$@_i \Diamond k$	*ass*
4	$@_j i$	$(2, C@Sym)$
5	$@_j \Diamond k$	$(3, 4, C@Nom)$

$(B\bot I)$ is just a special feature of Braüner's system, where inconsistency is local (with @) hence we need some way of its propagation into other states. In RND inconsistency is global so this rule is not required. ■

The same applies to proof construction rules.

Lemma 12.3 *Proof construction rules of Braüner's system are admissible in RND*

$[@RAA]$ is a special form of $[SUB]$ with Γ being unit clause, $k = 1$ and $\Delta = \bot$, $[@COND]$ is admissible as in the classical case (cf. Chapter 4).

Every application of $[B\Box]$ is eliminable in favor of the following sub-derivation:

	X	
k	SHØW: $@_i\Box\varphi$	$[l+1, SUB]$
$k+1$	$\neg @_i\Box\varphi$	*ass*
$k+2$	$@_i\neg\Box\varphi$	$(k+1, C@\neg)$
$k+3$	$@_i\Diamond j$	$(k+2, C@\pi)$
$k+4$	$@_j\neg\varphi$	$(k+2, C@\pi)$
	$\vdots$	
l	$@_j\varphi$	$(X, k+3, by\ assumption)$
$l+1$	$\bot$	$(k+4, l, @Res)$

where X is the set of elements of Γ from the formulation of $[B\Box]$ but treated as unit clauses. ■

On the basis of these two lemmata we obtain (strong) completeness of HRND-$\mathbf{K_{H@}}$.

We have shown in Chapter 4 that RND is general enough to simulate proof techniques from many known systems and that it may be simply restricted to an analytic form. It holds as well for HRND, in particular, we can simulate resolution system HyLoRes by stipulating that we always write down all possible assumptions (hence we always attempt an indirect proof), then we apply only elimination rules (only one direction of $(C@\neg)$, $(C@\alpha)$ and $(C@\beta)$). The applications of $(C@Param)$ are easily simulated by $(C@Nom)$ and $(C@Bridge)$. The difference is only in the form of setting out a proof: each line of a derivation in HyLoRes corresponds to a stage of construction of a derivation in HRND, where each clause is put in one line. So by vertical displaying of horizontally oriented elements (with omission of clauses that occur in more than one line) we can simulate in HRND every proof and disproof from HyLoRes. In this way we do not need to apply building-up rules (including $(@W)$) at all, and our derivations obey subformula property. We can apply known strategies used for resolution and tested on HyLoRes and, in our opinion, thus obtained derivations are more readable. In a similar way we may show that Blackburn's tableau sat-calculus (and decision procedure) may be simulated in HRND. It makes HRND much better tool for proof search in modal logics than variants of RND considered in Chapters 7, 8, and 9.

Extensions

Extending HRND to stronger hybrid languages does not make serious problems, e.g. we can use HyLoRes rules for $\downarrow$. Defining suitable rules for quantifiers is also easy. We can extend HRND to first-order logic, as well. One way is to use clausal versions of Blackburn's rules described in Section 12.3. The other consists in generalizing the rule $[SUB]$ and it was presented in Indrzejczak [150] in two versions: for classical and free logic quantifiers (cf. Chapter 4). In what follows we rather focus on defining rules for several modal logics over $\mathbf{K_{H@}}$. HRDN may be extended to stronger modal logics in a similar fashion as MRND in Chapter 7. However, some more general approach is better; Indrzejczak [148] considered three forms of rules:

- with 1-parameter-formula φ

  $(1R\text{-}A)$ $\Gamma, \varphi \ / \ \Gamma$

- with 2-parameter-formulae φ and ψ

  $(2Exp\text{-}A)$ $\Gamma, \varphi \ / \ \Gamma, \psi$ or

  $(2R\text{-}A)$ $\Gamma, \varphi \ ; \ \Delta, -\psi \ / \ \Gamma, \Delta$

- with 3-parameter-formulae φ, ψ and χ

  $(3Exp\text{-}A)$ $\Gamma, \varphi \ / \ \Gamma, \psi, \chi$ or

  $(3RExp\text{-}A)$ $\Gamma, \varphi \ ; \ \Delta, -\psi \ / \ \Gamma, \Delta, \chi$ or

  $(3R\text{-}A)$ $\Gamma, \varphi \ ; \ \Delta, -\psi \ ; \ \Sigma, -\chi \ / \ \Gamma, \Delta, \Sigma$

Theorem 12.4 *Rules of the type* $(2Exp\text{-}A), (2R\text{-}A)$ *and* $(3RExp\text{-}A), (3Exp\text{-}A), (3R\text{-}A)$ *are interderivable in HRND*

The proof of interderivability of rules $(2Exp\text{-}A)$ and $(2R\text{-}A)$ is the same as in the case of $(Exp\text{-}A)$ and $(Res\text{-}A)$ in MRND (cf. Section 7.4) and it may be directly extended for the rest of the rules.

Clearly, we may also introduce the contrapositives of these rules, obtained by interchanging conclusion-clause with one of the premise-clause and changing parameters with theirs complements. For example, both rules:

$(3RExp\text{-}A')$ $\Gamma, \varphi \ ; \ \Delta, -\chi \ / \ \Gamma, \Delta, \psi$ and

$(3RExp\text{-}A'')$ $\Gamma, -\chi \ ; \ \Delta, -\psi \ / \ \Gamma, \Delta, -\varphi$

are contrapositives of ($3RExp$-A); obviously, every schema of a rule is interderivable with its contrapositives either.

In fact, some of these types of rules were already present in the basic set. The rule ($C@Ref$) represents a particular case of a schema ($1R$-A). It may be seen as an expansion rule but it is rather a kind of one-premise resolution-rule. There is no reason to look for some equivalent.

Many rules represent the schema ($2Exp$-A). In general, it is a tableau-like expansion rule which may be replaced by interderivable resolution rule ($2R$-A).

Finally, one may observe that the rules ($C@\nu$), ($C@Nom$) and ($C@Bridge$) represent a schema ($3RExp$-A). At first sight, it may seem to be of essentially resolution-character. This is not the whole truth however, since some additional formula appears in the conclusion which makes it partly expansion rule as well. This kind of a rule is also interchangeable with an equivalent pure expansion rule of the form ($3Exp$-A) or with a more involved but pure kind of a resolution rule of the form ($3R$-A).

On the basis of this variety of forms one can extend HRND to stronger logics in different ways depending on the proof strategy which is under consideration. The type of a rule is encoded in its name, where the number says how many parameters must be specified, R – means resolution, Exp means expansion and A is a variable in the name substituted by the name of suitable axiom when parameters in the rule-schema are specified. In particular, all the rules of the type Exp are tableau-like, whereas rules of the type R are forms of resolution modulo substitution of parameters. The schema ($3RExp$-A) denotes rules of mixed character – something is cut out from premises and something new is added in the conclusion.

The table on the top of the next page specifies what substitutions for parameters we must perform in order to obtain the rules equivalent to suitable pure axioms in a modular way. If the places under the heading ψ and χ are blank it means that we have only unique rule of the form ($1R$-A); if only the place under χ is blank we can introduce either the rule of the form ($2Exp$-A) or ($2R$-A), otherwise we have three possible characterizations.

Notice also that for $H3'$ and HL' in the table we have in fact, more general schemata since χ does not refer to a single formula but to a clause. So, we have χ_1, χ_2 and χ_1, χ_2, χ_3 respectively, instead of a single χ. For instance, ($3RExp$-HL') has a form Γ, φ ; $\Delta, -\psi$ / $\Gamma, \Delta, \chi_1, \chi_2, \chi_3$. Due to this multiplication of the third parameter we should rather generalize the schemata of rules for more than 3 parameters. For example, in case of $H3'$ the following forms (and their contrapositives) are possible:

axiom	φ	ψ	χ
DC'	$@_i\Diamond j$	$\neg @_i\Diamond k$	$@_j k$
T'	$\neg @_i\Diamond i$	–	–
Irr	$@_i\Diamond i$	–	–
$4'$	$@_i\Diamond j$	$\neg @_j\Diamond k$	$@_i\Diamond k$
$5'$	$@_i\Diamond j$	$\neg @_i\Diamond k$	$@_j\Diamond k$
B'	$@_i\Diamond j$	$@_j\Diamond i$	–
As	$@_i\Diamond j$	$\neg @_j\Diamond i$	–
Ant	$@_i\Diamond j$	$\neg @_j\Diamond i$	$@_i j$
$Dich$	$\neg @_i\Diamond j$	$@_j\Diamond i$	–
Tri	$\neg @_i\Diamond j$	$@_j\Diamond i$	$@_i j$
$H3'$	$@_i\Diamond j$	$\neg @_i\Diamond k$	$@_j\Diamond k,\ @_k\Diamond j$
HL'	$@_i\Diamond j$	$\neg @_i\Diamond k$	$@_j\Diamond k,\ @_k\Diamond j,\ @_j k$

($4RExp$-A) $\Gamma, \varphi \ ; \ \Delta, -\psi \ ; \ \Sigma, -\chi_1 \ / \ \Gamma, \Delta, \Sigma, \chi_2$

($4R$-A) $\Gamma, \varphi \ ; \ \Delta, -\psi \ ; \ \Sigma, -\chi_1 \ ; \ \Pi, -\chi_2 \ / \ \Gamma, \Delta, \Sigma, \Pi$

Here is an example of a proof:

1	SHØW: $@_i(j \to \Box(\Diamond j \to j))$	$[12, SUB]$
2	$\neg @_i(j \to \Box(\Diamond j \to j))$	ass
3	$@_i j$	$(2, C@\alpha)$
4	$@_i \neg\Box(\Diamond j \to j)$	$(2, C@\alpha)$
5	$@_i\Diamond k$	$(4, C@\pi)$
6	$@_k\neg(\Diamond j \to j)$	$(4, C@\pi)$
7	$@_k\Diamond j$	$(6, C@\alpha)$
8	$@_k\neg j$	$(6, C@\alpha)$
9	$\neg @_k j$	$(8, C@NN)$
10	$@_j i$	$(3, C@Sym)$
11	$@_j\Diamond k$	$(10, 5, C@Nom)$
12	$\bot$	$(7, 11, 9, 3R\text{-}Ant)$

Nominal Existence Rules

Some of the important conditions need rules of a different form, similar to node-creating rules of Blackburn (see Section 12.3.2) defined for instances of Geach Axiom. Below, we give examples of rules for density and for Church-Rosser property:

$(HRND\text{-}4C')$	$\Gamma, @_i\Diamond j$ / $\Gamma, @_i\Diamond k$; $\Gamma, @_k\Diamond j$, where k is a new nominal	
$(HRND\text{-}CR)$	$\Gamma, @_i\Diamond j$; $\Delta, @_i\Diamond k$ / $\Gamma, \Delta, @_j\Diamond l$; $\Gamma, \Delta, @_k\Diamond l$, where l is a new nominal	

One more example of a proof:

1	SHØW: $@_i(\Diamond\Box j \rightarrow \Box\Diamond j)$	$[13, SUB]$
2	$\neg @_i(\Diamond\Box j \rightarrow \Box\Diamond j)$	ass
3	$@_i\Diamond\Box j$	$(2, C@\alpha)$
4	$@_i\neg\Box\Diamond j$	$(2, C@\alpha)$
5	$@_i\Diamond k$	$(3, C@\pi)$
6	$@_k\Box j$	$(3, C@\pi)$
7	$@_i\Diamond l$	$(4, C@\pi)$
8	$@_l\neg\Diamond j$	$(4, C@\pi)$
9	$@_k\Diamond m$	$(5, 7, HRND\text{-}CR)$
10	$@_l\Diamond m$	$(5, 7, HRND\text{-}CR)$
11	$@_m j$	$(6, 9, C@\nu)$
12	$@_m\neg j$	$(8, 10, C@\nu)$
13	$\bot$	$(11, 12, @Res)$

Such rules differ from those considered above not only because of the presence of side condition but also because they admit more than one conclusion-clause. We can easily define a general schema for rules corresponding to Geach axiom by generalizing the schema $(@GR)$ from Blackburn's tableau calculus. It takes the form:

$$\Gamma_1, \varphi_1 \ ; ....; \Gamma_{m+s}, \varphi_{m+s} \ / \ \Gamma_1, ..., \Gamma_{m+s}, \psi_1 \ ; ...; \Gamma_1, ..., \Gamma_{m+s}, \psi_{n+t},$$

where, formulae φ_i $(i \leq m+s)$ are sat-formulae displayed as premises, and ψ_i $(i \leq n+t)$ are sat-formulae displayed as consequences of this tableau rule-schema.

HRND enables also a simulation of other generalizations obtained so far on the ground of several proof methods. To close the discussion we recall Braüner's general result for geometric theories. We can define HRND counterpart of Braüner's rule from his SC version. It is enough to change every sequent $\Gamma \Rightarrow \Delta$ into corresponding clause $-\Gamma, \Delta$ and rewrite the rule-schema (BGR) accordingly. One may equally easily redefine in clausal form hybrid versions of Baldoni's rules stated at the end of Section 12.3 – we leave it to the reader.

Presented proof systems show that passing to hybrid languages may help to overcome many limitations of proof theory for standard modal logics. They also open new perspectives for development of reasoning methods and inventing new techniques of proof theory in general. In particular, HRND seems to be quite handy deductive system because its essentially hybrid character fits pretty well with the spirit of MHL. Easiness of simulation of other systems, shows that it may be used as a convenient framework for uniform treatment of a great number of modal logics based on solutions borrowed from different fields. Perhaps HRND may be also used for experimentation with different strategies of proof-search in order to measure their efficiency. But this claim requires further investigation.

Chapter 11

Modal Hybrid Logics

In this Chapter we briefly describe a powerfull extension of standard modal logic obtained by some modifications of the language. The fundamental change, forming the basis of the whole family of hybrid languages, involves the addition of special symbols called nominals. They enable explicit reference to states in Kripke models. The name of this approach reflects the fact that nominals are at the same time names of states in a model, and sentences of a modal language.

Although the first attempts in this field are quite old and can be traced back to Arthur Prior's work on modal and tense logics in late 1950s, the serious and systematic studies started in 1990s. Contemporary modal hybrid logic (MHL in short) seems to be one of the most dynamic branches of modern modal logic and offers a lot of improvements over classical results.

This Chapter is devoted mainly to the presentation of the problem of hybrid languages expressivity. We provide a survey of the most important hybrid languages, logics and their hierarchy. In every case the weakest logic is defined and complete axiomatization is presented. In particular, after informal introduction and a sketch of historical development, we introduce in Section 11.2 the basic hybrid language with nominals and its variant with satisfaction operators. Two subsequent sections present an axiomatic treatment of basic hybrid logics and general completeness results. In Section 11.5 we consider temporal hybrid logics and their expressive abilities. The next Section is devoted to a short presentation of the very strong hybrid languages using additional modal functors and binders of nominal variables. We conclude this survey with an exposition of first-order modal hybrid logic, short remarks on decidability and complexity of hybrid logics, and results concerning interpolation property.

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*, Trends in Logic 30, DOI 10.1007/978-90-481-8785-0_11,

It must be stressed that this Chapter has a rudimentary character; it is merely a collection of results with references. Usually no proofs are offered since interested reader may find them in referred papers. The presentation is heavily based on our survey [155]; more comprehensive introductions to MHL may be found in [13], [273] or [57].

11.1 Hybrid Logic in Nutshell

11.1.1 Motivation

Before we present MHL we should say a few words on motivations for the introduction of this variant of modal logic. In the last three chapters we have focused on labelled approach, trying to show that it is in many respects a better approach to formalization of modal logics than standard syntactical approach. Still we restricted our interests to some form of external labelling, while, in Section 6.1, we have distinguished also internalised and mixed labelling. The main problem with the application of ordinary proof methods to standard modal logics is connected with the fact that they are hardly suitable to handle the information which is under the scope of modal operators. Labels help to overcome the problem, but in case of medium labelling we have noted that the success is limited. Strong labelling seems to offer a uniform syntactic frame comparable to successfull semantic framework provided by relational models. But internalised labelling, present in hybrid logics, is even stronger and we focus on this approach as the final proposal.

Generally, internalised labelling consists of the enrichment of the object language obtained via sorting (of the atoms) and addition of the new operators and/or modalities. As for the sorting, the most basic and important innovation is the introduction of nominals – variables being names of states in a model. This is the fundamental step because in this way we introduce a local perspective into language which was not accessible in standard modal languages.

What do we get with the help of such an enrichment? In particular, do we have some substantial advantages over standard modal languages? This question is particularly interesting in the context of sorting. It is well known that in the case of first-order languages we do not get more expressiveness if we use many sorts of variables – we may only obtain a more compact and simpler formulation of things already expressible in standard one-sorted language. However, in the context of modal languages the use of several sorts of (propositional) variables leads to real changes in expressive power and in consequence to further improvements. So, hybrid modal languages are constructed mainly as tools for repairing the situation of asymmetry between elements of relational structures and language abilities. In short, an introduction of hybrid languages give us the following advantages:

- more expressive language
- better behavior in completeness theory
- more natural and simpler proof theory
- good behavior in decidability, complexity, interpolation and other important features

The first item, in the most literal sense, means that we have more validities in the logic formulated in enriched language. But more important fact is that hybrid languages allow us to define many frame properties which are not expressible in standard modal languages.

These improved expressive capabilities lead to more straightforward, and in fact complete, theory of frame definability. General completeness theorems obtained in MHL are also simpler than respective results in the standard ML, like famous Sahlqvist completeness theorem.

In what sense proof systems for MHL are more natural and simpler, we will show in the next Chapter but a few words of explanation are in order. We have mentioned that the application of standard proof methods to modal logics is complicated because of the difficulties with handling the sentences which are under the scope of modal operators. As we will see, in modal hybrid logics there are natural tools, namely nominals and satisfaction operators (shortly sat-operators), to deal with this problem. Every modal sentence in MHL may be broken into separate parts; one of them carry information on the structure of a model, whereas the other gives us directly the sentence being previously in the scope of modal operators. This natural way of decomposing a complex information into simpler parts, makes easier the transfer of non-axiomatic methods from classical logics to modal logics. Hence, richer languages of MHL offer a more general and uniform syntactical setting for modal proof theory.

Last thing worth mentioning is that in many cases (but not all) we do not need to pay for the improved expressive power of the language. One of the very important features of logics is their decidability and complexity of decision procedures. As we will see, hybrid counterparts of decidable modal logics are still decidable and usually complexity is also untouched (for example sat-problem for basic hybrid logic is $PSPACE$-complete as in standard modal logics **K**). Moreover, in many respects, hybrid logics behave better than standard modal logics – it is evident, for example, in the case of interpolation theorems.

11.1.2 Historical Remarks

Although MHL is quite a fresh branch of modal logic it has origins in late 1950s. But the importance of hybrid logic was not recognized properly until 1990s. I'm not going to enter into historical details (one should consult [13, 199] for Prior's ideas), but few words are in order.

All the sources agree that the name of the inventor of MHL belongs to Arthur Prior. He is well known as a father of standard tense logic, but some of his later contributions passed unnoticed. Prior devised two different calculi formally related to McTaggart's analysis of time in terms of A- and B-series. Standard tense logic (T-calculus) using tense constants F and P corresponds to A-series (time expressed in terms of past, present and future). I-calculus (later called U-calculus), using binary I-relation over instants of time, corresponds to B-series (earlier/later).

Although I-calculus is more expressive than T-calculus, Prior was convinced that tenses are metaphysically more fundamental. I-calculus provides only a convenient, but indirect way of speaking. So the Prior's problem was: how to show the primacy of T-calculus over I-calculus?

The solution he finally proposed was to develop I-calculus inside T-calculus via extension of the language, and this led him to the invention of strong hybrid logic with instant-variables and $\forall$. In [225], inspired by Quine's famous considerations on modality, he introduced the concept of four grades of tense-logical involvement. Whereas in the first grade, tenses are regarded just as handy definitions added to I-calculus, further grades offer essentially hybrid ideas. In the second grade, Prior introduced formulae of the form $T(a, \varphi)$ meaning "φ is true at time a" and, moreover, he admitted that instant variables a, b, c should also represent propositions. So, two essential ideas of contemporary MHL were introduced: internalization of satisfaction relation (here relative to time instants) and sorting of propositions into ordinary and nominals (as they are commonly called nowadays).

The first idea, of using some syntactical operators which encode semantical satisfaction relation, was quite popular. One may recall at least three early well known constructions that make use of such operators: the situation calculus of J. McCarthy and P. Hayes [188], topological logic of N. Rescher, A. Urquhart [231], and "Holds" operator of J. Allen [4] in his language for temporal representation in AI. Independent line of thought leading to similar ideas is present in the work of J. Perzanowski [208, 209, 210] introducing the general theory of modal operators ("makers") in formal ontology. A similar concept, but developed on the metalevel, was inherent in the idea of labelled deduction, described in Chapters 8, 9, and 10 as the external approach to representation of states.

The second idea, although more fundamental for MHL (there are hybrid languages with only nominals), was for a long time forgotten. The early work of Prior's student R. Bull [58] introduces "history variables" for representing paths in branching tense logic, but it was unnoticed and for a long time there are no traces of interest in using nominals. The idea of using nominals comes back in the number of papers (e.g. [202, 203, 102, 114]) written by logicians from Sofia school (Gargov, Tinchev, Passy, Goranko) and devoted particularly to the development of **CPDL** (Combinatory Propositional Dynamic Logic). By the way, except the reinvention of nominals, in the aforementioned works we have also a development of hybrid binders, see e.g. [114].

A genuine hybrid logic movement started with the works of P. Blackburn [29, 30] devoted to nominal tense logic, and with the works of J. Seligman [248] devoted to proof methods for situation theory. Since then, many researchers, including M. Tzakova, M. Marx, C. Areces, T. Braüner, Balder ten Cate and many others, took part in the development of strong and versatile theory of MHL.

11.2 Basic Hybrid Logic

11.2.1 Basic Hybrid Language

We obtain the *basic hybrid propositional modal language* $\mathbf{L_{H@}}$ by adding to $\mathbf{L_M}$ (or $\mathbf{L_T}$):

(a) the second sort of propositional symbols called *nominals*. We assume denumerable set $NOM = \{i, j, k, ...\}$ such that $PROP \cap NOM = \varnothing$; $PROP \cup NOM = AT$ is the set of atomic formulae. Members of NOM are introduced for naming states of a model domain.

(b) a denumerable collection of unary satisfaction operators indexed by nominals $@_i$. The new clause for non-atomic formulae is:

- if $\varphi \in FOR$ and $i \in NOM$, then $@_i\varphi \in FOR$

and it reads "formula φ is satisfied in a state i".

It is convenient to distinguish some classes of formulae. Every formula built up from nominals and logical constants only, is called *pure formula*, every formula of the shape $@_i\varphi$ or $\neg @_i\varphi$ is called *sat-formula*. Some examples:

$\Diamond(i \wedge p)$	–	neither pure nor sat-formula
$i \rightarrow \Diamond j$	–	pure but not sat-formula
$@_i(p \rightarrow \Diamond q)$	–	sat- but not pure formula
$@_i j$, $@_i\Diamond j$	–	both pure and sat-formulae

It should be observed that both examples of pure sat-formulae play very important roles. The first one expresses identity of states named i and j, and the second one expresses accessibility of j from i.

Note two important features of $\mathbf{L_{H@}}$:

- Both nominals and satisfaction operators are genuine language elements not an extra metalinguistic machinery. This is what we've called internalised approach in contrast to external approach present in Fitting's or Gabbay's solutions.

- Although nominals are terms, they are treated as ordinary sentences. In particular, they can be connected with the help of boolean operators and combined with modal and tense operators. In fact, they play a double role:

  - of propositional symbols representing propositions of the form "the name of the actual state is i";
  - of names of states when they occur as indexes of unary satisfaction operators.

Some authors (e.g. Blackburn [30], Tzakova [277], Demri [78]) prefer to have a weaker language, with only nominals added but without satisfaction operators, as the basic hybrid language. In what follows, we use $\mathbf{L_H}$ to denote such a language and we will call it *the weak hybrid propositional language.*

11.2.2 Hybrid Models

What is nice with MHL is the fact that changes in the language are sometimes so small that they do not affect seriously the rest of the machinery applied in ML. In particular, the modifications in the relational semantics are minimal. The concept of a frame is the same as in ordinary normal modal (or tense) logics, only on the level of models we have some changes. A model on the frame $\mathfrak{F}$ is any structure $\mathfrak{M} = \langle \mathfrak{F}, V \rangle$, where V is a valuation function on atoms ($V : AT \longrightarrow \mathcal{P}(\mathcal{W})$) such that for any $i \in NOM$, $V(i)$ is a singleton. Hence, in case of propositional symbols V behaves as usual, whereas in case of a nominal i, $V(i)$ picks up the unique state assigned to i by the model.

Satisfaction of new formulae in states of a model is defined as follows:

$$\begin{array}{lll} \mathfrak{M}, w \vDash i & \text{iff} & \{w\} = V(i) \text{ for any } i \in NOM \\ \mathfrak{M}, w \vDash @_i\varphi & \text{iff} & \mathfrak{M}, w' \vDash \varphi, \text{ where } \{w'\} = V(i) \\ & & (\text{or simpler } \mathfrak{M}, V(i) \vDash \varphi) \end{array}$$

The concepts of global satisfiability and of validity are the same as for ordinary modal language. Also definitions of consequence relations remain intact. The only difference is that if we say "model" we mean a model in the hybrid sense with a constraint on valuation of nominals.

Note that sat-operators enable us to jump to the named state, so in consequence we have:

Lemma 11.1 $\mathfrak{M}, w \vDash @_i\varphi$ *iff* $\mathfrak{M} \vDash @_i\varphi$

Let us focus on some consequences of the above definitions. The most important features of $\mathbf{L_{H@}}$ seem to be:

1. Internalization of local discourse – nominals give a direct representation of states in a language (we have an object-language mechanism for storing model data)

2. Possible jumping to already specified states in a model (we have a mechanism for retrieving model data)

3. Internalization of $\vDash$ by sat-formulae $@_i\varphi$

4. Representation of identity theory (for states) by pure formulae $@_ij$; we have $\mathfrak{M}, w \vDash @_ij$ iff $V(i) = V(j)$

5. Internalization of accessibility relation by pure formulae $@_i\Diamond j$; we have $\mathfrak{M}, w \vDash @_i\Diamond j$ iff $\langle V(i), V(j)\rangle \in \mathcal{R}$

One should note that points 2–5 are due to the presence of satisfaction operators, so in $\mathbf{L_H}$ we have only the first property.

11.2.3 Logic

Let us look at some syntactic properties of our new language elements. First of all, note that satisfaction operators are indeed modal – in fact normal modal – constants. One can easilly check that they satisfy:

$K@$ $@_i(\varphi \to \psi) \to (@_i\varphi \to @_i\psi)$ and
$RG@$ $@_i\varphi$ is valid, whenever φ is valid

Let $\mathbf{K_{H@}}$ denote the set of all valid formulae in $\mathbf{L_{H@}}$. It is easy to check that $\mathbf{K} \subseteq \mathbf{K_{H@}}$ and that $\mathbf{K_{H@}}$ satisfies closure conditions of normal modal logic. $\mathbf{K_{H@}}$ is indeed the weakest normal modal logic in $\mathbf{L_{H@}}$. Analogously we will use $\mathbf{K_H}$ as a name of the basic hybrid logic in $\mathbf{L_H}$ and $\mathbf{Kt_{H@}}$, $\mathbf{Kt_H}$ as names of respective temporal logics in suitable hybrid versions of $\mathbf{L_T}$. All these logics are also normal modal logics.

Clearly, due to the richer language $\mathbf{K_{H@}}$ contains denumerably many new tautologies e.g.

$$\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \to \Diamond(p \wedge q) \tag{11.1}$$

One can easily check that if we change i with some propositional variable we obtain non-valid formula in $\mathbf{K}$ – it is valid on frames with functional accessibility relation. But if we check it in any state w of any hybrid model we can see that both states w' and w'' that must be accessible from w in order to satisfy an antecedent are denotations of i, so they are the same state which guaranties that consequent is satisfied too.

As we shall see in the next sections, the expressivity of hybrid language has more serious character than just the presence of new tautologies. We may state new frame-defining formulae – e.g.: $i \to \neg\Diamond i$ defines irreflexivity and $i \to \neg\Diamond\Diamond i$ defines asymmetry. Moreover, $\mathbf{K_{H@}}$ is decidable and $PSPACE$-complete just like ordinary $\mathbf{K}$ (see [9]).

11.3 Complete Hilbert Calculi for $\mathbf{K_{H@}}$ and $\mathbf{K_{H}}$

We will focus on the proof theory for MHL in the next Chapter but our considerations will be connected with the practically useful formalizations. Axiomatic formulations of suitable hybrid logics will be stated here since they are useful for considerations on expressiveness, in the context of completeness results. The axiomatic (or Hilbert) formalization of the basic hybrid logic $\mathbf{K_{H@}}$ is denoted by H-$\mathbf{K_{H@}}$ and, in addition to axioms of H-$\mathbf{K}$, contains:

Specific Hybrid Axioms:

$K@$	$@_i(p \to q) \to (@_i p \to @_i q)$
$Selfdual@$	$@_i p \leftrightarrow \neg @_i \neg p$
$Intro@$	$i \wedge p \to @_i p$
$Ref@$	$@_i i$
$Agree$	$@_i @_j p \leftrightarrow @_j p$
$Back$	$\Diamond @_i p \to @_i p$

Rules:

(MP)	$\vdash \varphi \to \psi, \vdash \varphi \ / \ \vdash \psi$
(RG)	$\vdash \varphi \ / \ \vdash \Box\varphi$
$(RG@)$	$\vdash \varphi \ / \ \vdash @_i\varphi$
$(SUB@)$	$\vdash \varphi \ / \ \vdash e(\varphi)$,
	where $e : PROP \longrightarrow FOR$, but $e : NOM \longrightarrow NOM$

Note, that this axiomatization is not in a sense structural since we have an important constraint on substitution rule. $(SUB@)$ allows of substitution of any formula (including nominals) for ordinary propositional symbols but for a nominal we may substitute only a nominal. To make this evident we have chosen an axiomatization in object language with explicit substitution rule instead of a system formulated with the help of axiom schemata, as we did in the preceding chapters. Hence, $@_i(j \to k) \to (@_i j \to @_i k)$ is a proper substitution on $K@$, as well as other substitutions of the above axioms with p replaced with some nominal. On the other hand, something like $@_i p$ is an illegitimate "substitution" on $Ref@$.

Our axiomatization is sufficient for completeness but the full character of @ is not evident from it. One can learn more from the following lemma.

Lemma 11.2 *The following are H-$\mathbf{K_{H@}}$-theses (or rather schemata of theses):*

Sym@	$@_i j \leftrightarrow @_j i$
Tran@	$@_i j \wedge @_j k \rightarrow @_i k$
Nom1	$@_j \varphi \wedge @_j i \rightarrow @_i \varphi$
Nom2	$@_j \varphi \wedge @_i j \rightarrow @_i \varphi$
Bridge	$\Diamond i \wedge @_i \varphi \rightarrow \Diamond \varphi$
ConvK@	$(@_i \varphi \rightarrow @_i \psi) \rightarrow @_i(\varphi \rightarrow \psi)$

Now we can read off the identity theory of @ from these theorems.

As an illustration we give a (schema of a) proof of the *Bridge*

1. $\vdash i \wedge \neg\varphi \rightarrow @_i \neg\varphi$	$(Intro@)$
2. $\vdash \Diamond(i \wedge \neg\varphi) \rightarrow \Diamond @_i \neg\varphi$	(1, *by K-admissible rule RM*)
3. $\vdash \Diamond i \wedge \Box\neg\varphi \rightarrow \Diamond(i \wedge \neg\varphi)$	(*K-thesis*)
4. $\vdash \Diamond i \wedge \Box\neg\varphi \rightarrow \Diamond @_i \neg\varphi$	$(2, 3, by\ CPL)$
5. $\vdash \Diamond @_i \neg\varphi \rightarrow @_i \neg\varphi$	$(Back)$
6. $\vdash \Diamond i \wedge \Box\neg\varphi \rightarrow @_i \neg\varphi$	$(4, 5, by\ CPL)$
7. $\vdash \Diamond i \wedge \neg @_i \neg\varphi \rightarrow \neg\Box\neg\varphi$	$(6, by\ CPL)$
8. $\vdash \Diamond i \wedge @_i \varphi \rightarrow \Diamond \varphi$	$(7, Selfdual@, Pos)$

Theorem 11.1 (Completeness): *The above axiomatic system is strongly complete for* $\mathbf{K_{H@}}$

Soundness of the H-$\mathbf{K_{H@}}$ is easy to prove, the proof of completeness is by standard canonical model construction applied in modal logics. But something more is needed for extensions of $\mathbf{K_{H@}}$ if we want to obtain some general completeness theorem. Let H-$\mathbf{K^+_{H@}}$ be H-$\mathbf{K_{H@}}$ with 2 additional rules:

$(NAME)$	$\vdash @_i \varphi \ / \ \vdash \varphi$, provided $i \notin \varphi$
(BG)	$\vdash @_i \Diamond j \rightarrow @_j \varphi \ / \ \vdash @_i \Box\varphi$, provided $i \neq j$ and $j \notin \varphi$

Both rules are admissible in H-$\mathbf{K_{H@}}$, so we have:

Lemma 11.3 *Th(H-$\mathbf{K_{H@}}$)=Th(H-$\mathbf{K^+_{H@}}$)*

Note again that both additional rules are not standard because their applications must satisfy side conditions. In this respect they are similar to famous Gabby-style nonstructural rules applied for defining frame conditions undefinable by standard modal formulae. Let's look at the rule (BG).

The premise says that if the denotation of j is accessible from the denotation of i, then φ is satisfied in j. But j is arbitrary which is guaranteed by the proviso, so it means that φ is satisfied in every accessible (from i) state. This justifies the assertion that $\Box\varphi$ is satisfied in (the denotation of) i. The name (BG) comes from *Bounded Generalization* because it is a modal analog of *Universal Generalization* from first-order logic. But it is bounded because the premise is conditional (j is not simply arbitrary but arbitrary i-accessible state). In that it is more like respective rule from free logic. The sense of $(NAME)$ is clear: if φ is satisfied in an arbitrary state (again by syntactical proviso), then it is simply valid. Despite its simplicity the rule plays an important role in the general completeness theorem stated below. As we shall see in the next Chapter it is also the theoretical basis for many proof systems called there sat-calculi.

As we have noticed, both rules – being admissible – have no impact on the set of theses of H-$\mathbf{K_{H@}}$. But they have a strong influence on the redundancy of the set of primitive rules. For example, ordinary (RG) is derivable in H-$\mathbf{K^+_{H@}}$. Sometimes (see e.g. [35]) different nonstandard rules are applied, particularly useful for completeness proof and when @ is not present.

Lemma 11.4 *The following rules are admissible in H-*$\mathbf{K_{H@}}$ *(or derivable in H-*$\mathbf{K^+_{H@}}$*):*

$(NAME')$	$\vdash i \to \varphi \;/\; \vdash \varphi$, provided $i \notin \varphi$
$(PASTE)$	$\vdash @_i\Diamond j \wedge @_j\varphi \to \psi \;/\; \vdash @_i\Diamond\varphi \to \psi$,
	provided $i \neq j$ and $j \notin \varphi, \psi$

For the sake of illustration we put the proof of derivability of $(NAME')$ in H-$\mathbf{K^+_{H@}}$ (by $(NAME)$)

1. $\vdash i \to \varphi$	$(Premise,\ i \notin \varphi)$
2. $\vdash @_i(i \to \varphi)$	$(1, RG@)$
3. $\vdash @_i i \to @_i\varphi$	$(2, K@)$
4. $\vdash @_i i$	$(Ref@)$
5. $\vdash @_i\varphi$	$(3, 4, MP)$
6. $\vdash \varphi$	$(1, 5, NAME)$

These rules may be used instead of (BG) and $(NAME)$. Moreover, $(PASTE)$ is deductively stronger than (BG) because we may not only show the derivability of this rule by $(PASTE)$, but also deduce one of the axioms from our basis, namely *Back*.

It is also possible to axiomatize $\mathbf{K_H}$ – the set of all valid formulae in $\mathbf{L_H}$. We should add to axioms of H-$\mathbf{K}$ only one (scheme of) axiom:

$Nom\ \ \Diamond^n(i \wedge \varphi) \to \Box^m(i \to \varphi)$ for $n, m \geq 0$

Instead of $(RG@)$ we have (in addition to $(MP), (RG)$ and $(SUB@)$) a rule:

$(NAMELITE)\ \vdash \neg i \ /\ \vdash \bot$

This rule has rather special character. It is admissible in every consistent extension of H-$\mathbf{K_H}$. Note however that $\neg i$ is not valid on any frame. So the function of this rule is only to make inconsistent every logic with $\neg i$ added as an axiom.

If we want an axiomatization of H-$\mathbf{K_H}$ which is an analogon of H-$\mathbf{K^+_{H@}}$ we should add $(NAME')$ and the following @-free version of $(PASTE)$:

$(PASTE')\ \vdash \Diamond^n(i \wedge \Diamond(j \wedge \varphi)) \to \psi \ /\ \vdash \Diamond^n(i \wedge \Diamond\varphi) \to \psi$, for $n \geq 0$ provided $i \neq j$ and $j \notin \varphi, \psi$

In fact, $(NAMELITE)$ is a special case of $(NAME')$ with $\varphi = \bot$, so we can get rid of this rule in the extended axiomatization.

11.4 General Completeness Results

Now we are able to state rather general completeness theorem for considerable number of extensions of H-$\mathbf{K^+_{H@}}$ (or H-$\mathbf{K^+_H}$) obtained with the help of pure axioms.

Theorem 11.2 (Pure completeness) *Let Γ be any set of pure formulae, then H-$\mathbf{K^+_{H@}} + \Gamma$ is strongly complete for the class of frames defined by Γ.*

We will sketch a completeness proof. It is a mix of modal and first-order ideas – essentially a combination of canonical model construction and witnessed Henkin method. In addition to usual concepts of consistent and maximal sets we need:

Definition 11.1

- Γ is *named* iff it contains at least one nominal (it is the name of Γ)

- Γ is $\Diamond$-*saturated* iff for all $@_i\Diamond\varphi \in \Gamma$, there is a nominal j such that $@_i\Diamond j \in \Gamma$ and $@_j\varphi \in \Gamma$

These additional concepts play an important role in suitable modification of Lindenbaum construction.

Lemma 11.5 (Lindenbaum) *Every H-$\mathbf{K}^+_{\mathbf{H}@} + \Gamma$-consistent set can be extended to a named, $\Diamond$-saturated, maximal, H-$\mathbf{K}^+_{\mathbf{H}@} + \Gamma$-consistent set.*

A sketch of a proof: Similarly as in the Henkin proof for first-order logic we must supply a countably infinite set of new nominals, its arbitrary enumeration and some enumeration of all formulae in the extended (by new nominals) language. The procedure of extending our consistent set is mostly standard, by addition of each new formula which does not lead to inconsistency. Two points should be noticed:

- In order to get a named set in the first step of the construction we add the first new nominal. By $(NAME')$ this set must be consistent.
- In order to get $\Diamond$-saturated set, every time we add in a consistent way a formula of the type $@_i\Diamond\psi$ we add also $@_i\Diamond j$ and $@_j\psi$, where j is a new nominal (witness). By $(PASTE)$ such an extended set must be also consistent.

Obviously, the union of all so generated sets satisfies postulated conditions. ■

We do not use canonical model construction from ordinary modal logic, where states are simply (all) maximal consistent sets. Here one set is enough and the states of this model are built up from equivalence classes of nominals from this maximal consistent set. Formally:

Definition 11.2 (Henkin Model) Henkin Model for H-$\mathbf{K}^+_{\mathbf{H}@}$+$\Gamma$-maximal, consistent set Δ is defined as $\mathfrak{M}_\Delta = \langle\, \mathcal{W}_\Delta, \mathcal{R}_\Delta, V_\Delta \rangle$ where:
$\mathcal{W}_\Delta = \{\mid i \mid : i \text{ is a nominal }\}$, where $\mid i \mid = \{j :\ @_i j \in \Delta\}$
$\mathcal{R}_\Delta(\mid i \mid, \mid j \mid)$ iff $@_i\Diamond j \in \Delta$
$V_\Delta(p) = \{\mid i \mid : @_i p \in \Delta\}$
$V_\Delta(i) = \{\mid i \mid\}$

By (almost) ordinary inductive argument we obtain:

Lemma 11.6 (Truth Lemma) $@_i\varphi \in \Delta$ *iff* $\mathfrak{M}_\Delta, \mid i \mid \models \varphi$

A sketch of a proof: One should note that because we make an induction on the formula which on one side of the equivalence is changed into sat-formula we must apply suitable axioms or theses. For example, if φ is a negation we must use $Selfdual@$, if it is an implication we must use $K@$ and $ConvK@$, if it is a sat-formula we need *Agree* and if it is a diamond-formula we need *Bridge*. ■

As a result of this construction we obtain a lemma which gives us automatically general completeness for every set of pure formulae that defines some frame conditions.

Lemma 11.7 (Frame Lemma) *If* Δ *is* $\Diamond$*-saturated* H-$\mathbf{K}^+_{\mathbf{H@}}+\Gamma$*-maximal, consistent set, then the frame of* $\mathfrak{M}_\Delta$ *satisfies all properties defined by* Γ*.*

It is obvious since Δ is named and contains all instances of Γ, so on the frame of this model all elements of Γ are valid. Hence, pure completeness theorem follows in a standard way. This result leads to better completeness theory due to more general theory of frame definability than standard modal logic provides. The following table lists some examples:

Pure axioms		
Name	Axiom	Frame-condition
D'	$\Box i \to \Diamond i$	Seriality (successors)
DC'	$\Diamond i \to \Box i$	Almost functionality
T'	$\Box i \to i$	Reflexivity
$\Box T'$	$\Box(\Box i \to i)$	Almost-reflexivity
Irr	$i \to \Box\neg i$	Irreflexivity
$4'$	$\Box i \to \Box\Box i$	Transitivity
$4C'$	$\Box\Box i \to \Box i$	Density
$Intr$	$\neg\Box i \to \Box\Box i$	Intransitivity
B'	$i \to \Box\Diamond i$	Symmetry
As	$i \to \Box\Box\neg i$	Asymmetry
Ant	$i \to \Box(\Diamond i \to i)$	Antisymmetry
$5'$	$\Diamond i \to \Box\Diamond i$	Euclideaness
Un	$\Diamond i$	Universality
$H3'$	$\Box(\Box i \to j) \vee \Box(\Box j \to i)$	Strong (right) connectedness
HL'	$\Box(\Box i \wedge i \to j) \vee \Box(\Box j \wedge j \to i)$	Weak (right) connectedness
$Dich$	$@_i\Diamond j \vee @_j\Diamond i$	Dichotomy
Tri	$@_i\Diamond j \vee @_j\Diamond i \vee @_i j$	Trichotomy

Note in particular that:

1. Many conditions from the table are not definable in $\mathbf{L_M}$ e.g.: irreflexivity, intransitivity, asymmetry, antisymmetry, universality, dichotomy and trichotomy.

2. All conditions except dichotomy and trichotomy are definable in $\mathbf{L_H}$.

For the sake of illustration we will show that Irr defines irreflexivity. Assume that $\mathcal{R}$ is irreflexive but Irr is not valid, so in some w we have $w \vDash i$ but $w \nvDash \Box\neg i$. So in some accessible w' we have $w' \vDash i$ but then $w = V(i) = w'$ which contradicts the assumption of irreflexivity. Now assume that Irr is valid in the frame where, for some w, $\mathcal{R}ww$. Let $V(i) = w$ (recall that canonical model is named!), so $w \vDash i$ and $w \vDash i \rightarrow \Box\neg i$. But if $w \vDash \Box\neg i$, then $w \nvDash i$ – contradiction.

One should note that this result is also in a sense simpler than celebrated Sahlqvist completeness theorem. The criteria for being Sahlqvist formula are rather complicated whereas the requirement of purity is extremely simple. But there are also some considerable limitations – pure-formulae define only first-order properties but still not all of them!

That second-order properties are not definable by pure formulae should be clear if we look at how standard translation works. For $\mathbf{L_{H@}}$ we add two clauses to the definition of ST_x from Section 5.4.3:

$$
\begin{array}{lcl}
ST_x(i) & = & x = c_i \\
ST_x(@_i\varphi) & = & \exists y(y = c_i \wedge ST_y(\varphi))
\end{array}
$$

where c_i is an individual constant and x, y are distinct variables not occuring in φ.

Lemma 5.2 still holds but recall that second-order quantification deals only with monadic predicates being standard translation of propositional variables from translated formula. But there are no such variables in pure formulae; nominals, despite their syntactic category, play the role of names and in enriched standard translation are mapped onto first-order individual constants. In consequence, every condition definable by pure formula must be elementary. Of course, second-order properties definable in standard modal languages are also expressible in hybrid languages, since trivially the former are contained in the latter, but they are not expressible by pure formulae.

But which first order properties are definable and which are not definable in the basic hybrid language? In particular, which are definable by pure formulae? Balder ten Cate [273] provides the following characterization theorems:

Theorem 11.3 *An elementary class of frames is definable by formulae of* $\mathbf{L_{H@}}$ *iff it is closed under ultrafilter morphic images and generated subframes.*

Theorem 11.4 *A class of frames is definable by pure formulae of* $\mathbf{L_{H@}}$ *iff it is elementary and closed under images of bisimulation systems.*

For suitable definitions and details of proofs one should consult [273]. Here we discuss some concrete negative examples of definability in $\mathbf{L_{H@}}$. For instance, not all Sahlqvist formulae have pure formulae equivalents, e.g. Church-Rosser property), predecessors, right- (left)-directedness.

- Church-Rosser property – $\forall xyz(\mathcal{R}xy \wedge \mathcal{R}xz \rightarrow \exists v(\mathcal{R}yv \wedge \mathcal{R}zv))$ is defined in $\mathbf{L_M}$ by axiom 2 : $\Diamond\Box\varphi \rightarrow \Box\Diamond\varphi$, but $\Diamond\Box i \rightarrow \Box\Diamond i$ doesn't work.
- Predecessors – $\forall x\exists y\mathcal{R}yx$ which is not defined in $\mathbf{L_M}$ either (although the converse, namely seriality, is defined by D and both properties are definable in $\mathbf{L_T}$).
- Right-directedness – $\forall xy\exists z(\mathcal{R}xz \wedge \mathcal{R}yz)$ is not definable in $\mathbf{L_M}$. Note that it is definable in $\mathbf{L_{H@}}$ by $@_i\Box p \rightarrow @_j\Diamond p$, but it is not a pure formula so pure completeness theorem does not apply. Left-directedness is undefinable in $\mathbf{L_{H@}}$ too.

As a result we have a strange situation. For hybrid logics in $\mathbf{L_{H@}}$ we have both: pure completeness, and

Theorem 11.5 Hybrid Sahlqvist completeness: *Let* Γ *be any set of Sahlqvist-formulae, then* H-$\mathbf{K^+_{H@}} + \Gamma$ *is strongly complete for the class of frames defined by* Γ.

But completeness fails for some combinations of pure and Sahlqvist formulae! e.g. H-$\mathbf{K^+_{H@}} + 2 + NG$ is incomplete, where NG is $\Diamond(i \wedge \Diamond j) \rightarrow \Box(\Diamond j \rightarrow i)$ and defines the following condition:

$$\forall xyzu(\mathcal{R}xy \wedge \mathcal{R}xz \wedge \mathcal{R}yu \wedge \mathcal{R}zu \rightarrow y = z) \tag{11.2}$$

Fortunately, the situation slightly changes when we move to hybrid tense language.

It would be also interesting to know if we could obtain an axiomatization which is sufficient for obtaining pure completeness theorem but which is more standard, in the sense that rules like (BG) or $(PASTE)$ are derivable. As we shall see it is possible in case of stronger languages, at least partly – for instance (BG) is derivable in H-$\mathbf{Kt_{H@}}$ (but with the help of $(NAME)$ however). Full elimination of nonstandard rules is possible when we have a local binder $\downarrow$ in a language, but in the case of basic language the presence of such rules is not incidental, which was shown in [40].

11.5 Hybrid Tense Logic

11.5.1 Impact of Past Operators

Hybrid tense logic shows some important differences with modal hybrid logic. It is easy to check that $\mathbf{L_{TH@}}$ (and even $\mathbf{L_{TH}}$)is strictly more expressive than $\mathbf{L_{H@}}$. In particular, three points should be stressed:

1. @ is in principle dispensable in the presence of past-operators, e.g. trichotomy may be defined by $Pi \vee i \vee Fi$. But it does not mean that @ is simply definable. Areces [9] shows that @ is eliminable in $\mathbf{L_{TH@\downarrow}}$ from all nominal-free sentences. A different way of simulating the effect of sat-operators in $\mathbf{L_{TH}}$ is shown in Demri's sequent calculus [78] (see the section on sequent calculi in Chapter 12).

2. Some frame-conditions undefinable in $\mathbf{L_{H@}}$ by pure formulae (although definable in $\mathbf{L_M}$) are definable by means of tense operators, e.g. Church-Rosser property (or directedness) is defined by $Fi \wedge Fj \rightarrow F(i \wedge FPj)$.

3. Some frame-conditions are definable that are not definable in any of $\mathbf{L_M}$, $\mathbf{L_T}$, $\mathbf{L_{H@}}$, e.g.:

- left directedness: $\forall xy \exists z(z < x \wedge z < y)$ is defined by PFi
- right discreteness: $\forall xy(x < y \rightarrow \exists z(x < z \wedge \neg \exists v(x < v < z)))$ is defined by $@_i(F\top \rightarrow FHH\neg i)$ (or $i \rightarrow (F\top \rightarrow FHH\neg i)$)

In fact, every Sahlqvist formula has a pure formula equivalent in $\mathbf{L_{TH@}}$ (see [115]), so we have:

Theorem 11.6 (Sahlqvist/pure completeness) *Let* Γ *be any set of pure or Sahlqvist formulae, then* H-$\mathbf{Kt}^{+}_{\mathbf{H@}} + \Gamma$ *is strongly complete for the class of frames defined by* Γ*.*

H-$\mathbf{Kt}^{+}_{\mathbf{H@}}$ is similar to H-$\mathbf{K}^{+}_{\mathbf{H@}}$ – we simply replace the axioms of H-**K** by the axioms of H-**Kt** and replace (RG) by two tense versions for G and H respectively. But some changes are possible, namely:

1. We can use one pure axiom $@_iFj \leftrightarrow @_jPi$ instead of two standard interaction axioms from H-**Kt**: $p \to GPp$ and $p \to HFp$.

2. If we add two axioms: $@_iGPi$ and $@_iHFi$ we can derive both (for G and H) tense versions of (BG).

Hence, in the completeness theory we can avoid some strange features of $\mathbf{L_{H@}}$ but there are some disadvantages – $\mathbf{Kt_{H@}}$ is still decidable but $EXPTIME$-complete, whereas **Kt** is in $PSPACE$ (as **K** and $\mathbf{K_{H@}}$). So the (basic) hybrid tense logic is more complex than ordinary tense logic **Kt**.

11.5.2 Tenses

A research on hybrid tense logic opens also a new perspective for formalization of English language tenses. Blackburn [31] has noticed that standard Priorean $\mathbf{L_T}$ already has a deictic nature but shows strong limitation in expressing language tenses. $\mathbf{L_{TH@}}$ yields referential perspective which makes possible to express Reichenbachian analysis of tenses in terms of three time points. The table lists the details

Reference	Tense	Example	Formula
E-R-S	Pluperfect	I had seen	$P(i \wedge P\varphi)$
E,R-S	Past	I saw	$P(i \wedge \varphi)$
R-E-S	Future-in-the-Past	I'd see	$P(i \wedge F\varphi)$
R-S,E	Future-in-the-Past	I'd see	$P(i \wedge F\varphi)$
R-S-E	Future-in-the-Past	I'd see	$P(i \wedge F\varphi)$
E-S,R	Present perfect	I've seen	$P\varphi$
S,R,E	Present	I see	φ
S,R-E	Prospective	I'm going to see	$F\varphi$
S-E-R	Future perfect	I'll have seen	$F(i \wedge P\varphi)$
S,E-R	Future perfect	I'll have seen	$F(i \wedge P\varphi)$
E-S-R	Future perfect	I'll have seen	$F(i \wedge P\varphi)$
S-R,E	Future	I'll see	$P(i \wedge \varphi)$
S-R-E	Future-in-the-Future	(Latin: abiturus ero)	$F(i \wedge F\varphi)$

where:
S – the point of speech
E – the point of event
R – the point of reference

[31] contains also other applications of hybrid tense languages to the analysis of language temporal phenomena like indexicals, anaphora, calendar terms. Other papers of Blackburn undertake the problem of extending the expressive power of hybrid language to cover interval based temporal languages, but this requires substantial changes in hybrid machinery by introducing further sorts of atoms (see e.g. [30]). Nevertheless, for many purposes even hybrid languages with backward-looking operators are still too weak. In what follows we describe briefly the most popular extensions.

11.6 Language Extensions

Although the basic hybrid language offers many improvements over standard modal language it has still strong limitations which may be overcome by further strengthenings. Moreover, some of them were historically the first forms of hybrid languages. Below, we consider some of the most important languages and their expressive hierarchy. We describe in particular:

- Extra modalities
    1. Global modalities
    2. Difference modalities
- Modal Binders
    1. Local binder
    2. Quantifiers

It should be stressed that early works on hybrid logics, in particular from Sofia school (like Gargov [102] or Goranko [114]), were concerned with stronger languages than those we presented so far. Studies on the basic and weak hybrid language started later in the middle of 1990s. In this paragraph we briefly recall two, surprisingly strong solutions.

11.6.1 Global Modalities

One of the popular solutions, not necessarily connected with hybrid logic is to use the so called global modalities. Notes to the last Chapter of [35] include interesting historical information on their use. They are called global because they are not characterised by accessibility relations but defined by reference to any state in a model. We use $\mathcal{A}$ (from Aristotle or "always") for universal (global) modality and $\mathcal{E}$ for its dual. Semantically they are defined as follows:

$$\mathfrak{M}, w \vDash \mathcal{A}\varphi \quad \text{iff} \quad \mathfrak{M}, w' \vDash \varphi \text{ for any } w'$$
$$\mathfrak{M}, w \vDash \mathcal{E}\varphi \quad \text{iff} \quad \mathfrak{M}, w' \vDash \varphi \text{ for some } w'$$

Let $\mathbf{L_{H@A}}$ denote $\mathbf{L_{H@}}$ with universal modality $\mathcal{A}$ or (interdefinable) existential modality $\mathcal{E}$. Notice that $\mathbf{L_{HA}} = \mathbf{L_{H@A}}$ since @ is definable:

$$@_i\varphi := \mathcal{A}(i \rightarrow \varphi) := \mathcal{E}(i \wedge \varphi) \tag{11.3}$$

So, in the presence of global modality, the difference between $\mathbf{L_H}$ and $\mathbf{L_{H@}}$ disapears and $\mathbf{L_{HA}}$ is at least as expressive as $\mathbf{L_{H@}}$. In fact, hybrid languages with $\mathcal{A}$ are strictly stronger which is evident if we consider computational behaviour of $\mathbf{K_{HA}}$.

$\mathbf{K_{HA}}$ is also decidable but global modalities are very strong. Even $\mathbf{K_A}$ – the basic logic of $\mathcal{A}$ in the standard language (no nominals) is $EXPTIME$-complete [125]. But if we add nominals, the situation does not change. Even if we add $\mathcal{A}$ to hybrid tense logic we still have $EXPTIME$-completeness, so both $\mathbf{K_{HA}}$ and $\mathbf{Kt_{HA}}$ are in the same complexity class as plain $\mathbf{K_A}$. Hence, at least from the point of view of complexity of $\mathbf{Kt_H}$, we do not loose anything if we add global modalities. But, when compared with $\mathbf{K_{H@}}$ (and if standard beliefs concerning relations between complexity classes are right), the satisfiability problem for hybrid logic with $\mathcal{A}$ is harder.

A complete axiomatization of $\mathbf{K_{HA}}$=$\mathbf{K_{H@A}}$ may be obtained by addition of the following axioms to H-$\mathbf{K_H}$:

$Dual_A$	$\mathcal{E}p \leftrightarrow \neg\mathcal{A}\neg p$
K_A	$\mathcal{A}(p \rightarrow q) \rightarrow (\mathcal{A}p \rightarrow \mathcal{A}q)$
T_A	$\mathcal{A}p \rightarrow p$
B_A	$p \rightarrow \mathcal{A}\mathcal{E}p$
4_A	$\mathcal{A}p \rightarrow \mathcal{A}\mathcal{A}p$
$Incl_\Diamond$	$\Diamond p \rightarrow \mathcal{E}p$
$Incl_i$	$\mathcal{E}i$
Nom_A	$\mathcal{E}(i \wedge p) \rightarrow \mathcal{A}(i \rightarrow p)$

Clearly, to the set of rules of H-$\mathbf{K_H}$ we must add:

(RG_A) $\vdash \varphi$ / $\vdash \mathcal{A}\varphi$

If we want to strengthen H-$\mathbf{K_{HA}}$ in order to get H-$\mathbf{K^+_{HA}}$, a formalization suitable for general pure completeness theorem, we must add $(NAME')$ and:

(BG_E) $\vdash \mathcal{E}(i \wedge \Diamond j) \rightarrow \mathcal{E}(j \wedge \varphi)$ / $\vdash \mathcal{E}(i \wedge \Box\varphi)$, provided $i \neq j$ and $j \notin \varphi$

In the set of axioms, formulae $Dual_A - 4_A$ simply reflect the fact that global modalities are normal **S5**-modalities. Interesting cases are the last three axioms. Alternatively, we could add just $Incl_i$ to H-$\mathbf{K^+_H}$ since the rest of the axioms is derivable.

11.6.2 Difference Modality

Another kind of modality very popular in early works on hybrid logic (see e.g. [102]) is the difference modality. In fact, it was firstly introduced in the context of ordinary modal languages (again, see the notes in [35]). Let $\mathbf{L_{MD}}$ denote $\mathbf{L_M}$ with difference possibility $\mathcal{D}$ or (interdefinable) difference necessity $\bar{\mathcal{D}}$ defined as follows:

$$\begin{array}{lll} \mathfrak{M}, w \models \mathcal{D}\varphi & \text{iff} & \mathfrak{M}, w' \models \varphi \text{ for some } w' \neq w \\ \mathfrak{M}, w \models \bar{\mathcal{D}}\varphi & \text{iff} & \mathfrak{M}, w' \models \varphi \text{ for any } w' \neq w \end{array}$$

Note that $\mathbf{L_{MD}}$ is strictly stronger than $\mathbf{L_{MA}}$ since $\mathcal{A}$ is definable by $\bar{\mathcal{D}}$ but not conversely:

$$\mathcal{A}\varphi := \varphi \wedge \bar{\mathcal{D}}\varphi \tag{11.4}$$

In fact, difference modality is so strong, that in $\mathbf{L_{MD}}$ we can even simulate nominals on the basis of the following definition: p is true at exactly one point iff $\mathcal{E}p \wedge \mathcal{A}(p \rightarrow \neg\mathcal{D}p)$ holds.

On the other hand, with respect to frame definability $\mathbf{L_{HA}}$ is as expressive as $\mathbf{L_{MD}}$, so addition of $\mathcal{D}$ to $\mathbf{L_{HA}}$ does not change its strength. The interested reader should consult [102] or [8] for details.

As a result of these language interdependencies we have the following hierarchy of expressivity:

$$\mathbf{L_{MA}} < \mathbf{L_{MD}} = \mathbf{L_{MAD}} = \mathbf{L_{HAD}} = \mathbf{L_{HA}} = \mathbf{L_{H@A}} \qquad (11.5)$$

We can obtain complete formalization of hybrid logic with $\mathcal{D}$ very simply. It is enough to add pure axiom $\mathcal{D}i \leftrightarrow \neg i$ to H-$\mathbf{K}^{+}_{\mathbf{H@}}$.

In this section we consider only the expressive power of hybrid languages on the set of all frames. Interesting results concerning selected classes of frames will be mentioned later but one fact should be noticed here. One can easily check that $\mathcal{D}\varphi$ is definable in **Kt4.3** (the basic tense logic of linear frames) by $P\varphi \vee F\varphi$, so on linear frames $\mathbf{L_{TH@}}, \mathbf{L_{THD}}$ and $\mathbf{L_{T}}$ have the same expressivity.

11.6.3 Modal Binders

The applications of special modalities described in the last section do not have particularly hybrid character. They were considered independently of investigations on hybrid logic and their importance for this field is connected with nice interplay of these modalities with hybrid machinery. But hybrid languages lead to specific enrichments; if we can name states in a model we can ask why not to quantify over states? So the next step is:

- add the third sort of atoms $SVAR = \{u, v, ...\}$ (state variables) to the basic hybrid language
- add some binders – quantifiers $\forall, \exists$ or local binder $\downarrow$

In fact, the application of quantifiers is present in the earliest approach to hybrid logic due to Prior [225]. His third grade tense logic uses both nominals (or rather state variables) and $\forall$. A local binder $\downarrow$ was invented much later and – in contrast to quantifiers borrowed from first-order language – is essentially hybrid concept, although some forms of it were applied outside MHL earlier (see Blackburn [38] for some historical remarks and Goranko [114] for the first application in hybrid languages). By the way: the application of binders (in particular quantifiers) is one of the sources of the name "hybrid" meant as a combination of propositional modal language and quantification.

An addition of the third sort of atoms is strictly speaking not necessary but it is easier to have distinct state symbols: nominals and variables. The situation is in a sense analogous to that in typical Gentzen-style proof theory for first-order logic, where we distinguish bound occurrences of variables and free occurrences (parameters) (cf. Chapter 3). But note that free state variables will be also admitted. The definitions of free and bound occurences of (state) variables, the scope of the binder, the sentence (no free variables) and other similar concepts, are the exact analogs of the definitions from first-order language stated in the first Chapter.

The definition of the frame and model is the same as for $\mathbf{L_{H@}}$ but we need also the concept of an assignment a for $\mathfrak{M}$ which is a mapping $a : SVAR \longrightarrow W$. The satisfaction of a formula is now defined for a model and an assignment. In particular, for the new elements we have the following conditions:

$\mathfrak{M}, a, w \models u$	iff	$w = a(u)$ for any $u \in SVAR$
$\mathfrak{M}, a, w \models \forall u\varphi$	iff	$\mathfrak{M}, a^u_{w'}, w \models \varphi$ for all w'
$\mathfrak{M}, a, w \models \exists u\varphi$	iff	$\mathfrak{M}, a^u_{w'}, w \models \varphi$ for some w'
$\mathfrak{M}, a, w \models \downarrow u\varphi$	iff	$\mathfrak{M}, a^u_w, w \models \varphi$

where a^u_w is an *u-variant of* a, namely, for any $v \in SVAR$:

$$a^u_w(v) := \begin{cases} w & \text{if } v = u \\ a(v) & \text{if } v \neq u \end{cases}$$

We should also admit free state-variables as arguments of @, so the more general condition is:

$\mathfrak{M}, a, w \models @_s\varphi$ iff $\mathfrak{M}, a, w' \models \varphi$ where $s \in NOM \cup SVAR$ and: $\{w'\} = V(s)$ if $s \in NOM$, or $w' = a(s)$ if $s \in SVAR$

Truth clauses show that we have the exact hybrid analogs of first-order quantifiers, but $\downarrow$ needs some comment. The difference between $\downarrow$ and $\forall$ is between local and global binding. $\downarrow$ enables to name a current state ($\downarrow$ binds state variable to current state). Note also that $\downarrow$ is self-dual.

Let $\mathbf{L_{H\forall}}, \mathbf{L_{H\downarrow}}, \mathbf{L_{H\downarrow\forall}}$ denote weak hybrid languages with added binders and $\mathbf{L_{H@\forall}}, \mathbf{L_{H@\downarrow}}, \mathbf{L_{H@\downarrow\forall}}$ respective languages with satisfaction operators.

Theorem 11.7 $\mathbf{L_{H@\forall}}$ *is strictly stronger than* $\mathbf{L_{H@\downarrow}}$*, since:*

1. $\downarrow$ is definable in $\mathbf{L_{H@\forall}}$: $\downarrow u\varphi := \exists u(u \wedge \varphi)$ but

2. $\mathbf{L_{H@\downarrow}}$ is preserved under generated submodels, whereas $\mathbf{L_{H@\downarrow\forall}}$ is not.

(obviously the same applies to $\mathbf{L_{H\forall}}, \mathbf{L_{H\downarrow}}$)

Theorem 11.8 $\mathbf{L}_{\mathbf{H}@\forall} = \mathbf{L}_{\mathbf{H}@\downarrow\forall}$ *and* $\mathbf{L}_{\mathbf{H}\forall} = \mathbf{L}_{\mathbf{H}\downarrow\forall}$

In contrast to weak hybrid language with $\mathcal{A}$, @ is not definable in $\mathbf{L}_{\mathbf{H}\forall}$. But if we add $\mathcal{A}$ to languages with binders we can obtain the interesting interdefinability results stated below as:

Theorem 11.9 $\mathbf{L}_{\mathbf{H}@\forall} = \mathbf{L}_{\mathbf{H}\downarrow\mathbf{A}} = \mathbf{L}_{\mathbf{H}\forall\mathbf{A}} = \mathbf{L}_{\mathbf{H}@\downarrow\mathbf{A}} = \mathbf{L}_{\mathbf{H}@\forall\mathbf{A}}$

Some of the equations are obvious if we remember that $\mathcal{A}$ defines sat-operator, moreover:

1. $\forall$ is defined in $\mathbf{L}_{\mathbf{H}\downarrow\mathbf{A}}$: $\forall u\varphi := \downarrow v\mathcal{A} \downarrow u\mathcal{A}(v \to \varphi)$, where $v \notin \varphi$

   and

2. $\mathcal{A}$ is defined in $\mathbf{L}_{\mathbf{H}@\forall}$: $\mathcal{A}\varphi := \forall u\, @_u\varphi$, where $u \notin \varphi$

As we shall see, the addition of binders strongly increases expressive power of hybrid languages but there are serious costs. Both basic hybrid logics with added binders $\mathbf{K}_{\mathbf{H}@\downarrow}$ and $\mathbf{K}_{\mathbf{H}@\forall}$ are undecidable (in fact, even $\mathbf{K}_{\mathbf{H}\forall}$ is undecidable!).

11.6.4 Axiomatization

Let H-$\mathbf{K}_{\mathbf{H}@\downarrow}$ be axiomatization of $\mathbf{K}_{\mathbf{H}@\downarrow}$,[1] obtained from H-$\mathbf{K}_{\mathbf{H}@}$ by the addition of:

DA $\;@_i(\downarrow u\varphi \leftrightarrow \varphi[u/i])$

Clearly, the proviso for the rule of Substitution must be changed a bit to deal with the presence of states variables. Nominals and state variables may be substituted for each other but state variables may be substituted for nominal/(free) state variable only if they are still free. One can easily prove the self-duality principle:

$S\text{-}D\downarrow$ $\;\downarrow u\varphi \leftrightarrow \neg\downarrow u\neg\varphi$

The addition of $(NAME)$ and (BG) to H-$\mathbf{K}_{\mathbf{H}@\downarrow}$ yields H-$\mathbf{K}^+_{\mathbf{H}@\downarrow}$. Pure completeness holds for H-$\mathbf{K}^+_{\mathbf{H}@\downarrow}$ exactly as for H-$\mathbf{K}^+_{\mathbf{H}@}$. What's more, we can axiomatize $\mathbf{K}^+_{\mathbf{H}@\downarrow}$ without (BG) and $(NAME)$ but using more standard

[1]For axiomatization of $\mathbf{K}_{\mathbf{H}\downarrow}$ with the use of (COV)-rules, see [38].

rules (no side conditions). Just add to H-$\mathbf{K_{H@\downarrow}}$ the following axioms and rules:

$Name\downarrow$	$\downarrow u(u \to \varphi) \to \varphi$, provided $u \notin \varphi$
$BG\downarrow$	$@_i\Box\downarrow u\ @_i\Diamond u$
$(RG\downarrow)$	$\vdash \varphi\ /\ \vdash\downarrow u\varphi$

We can axiomatize the set of all validities in the strongest hybrid language just by adding to H-$\mathbf{K^{+}_{H@}}$ the following axioms:

$Q1$	$\forall u(\varphi \to \psi) \to (\varphi \to \forall u\psi)$, where $u \notin VF(\varphi)$
$Q2$	$\forall u\varphi \to \varphi[u/s]$,
	where if s is a state variable it is free for u in φ
$Barcan@$	$\forall u @_i\varphi \leftrightarrow @_i\forall u\varphi$

and

$(Gen) \vdash \varphi\ /\ \vdash \forall u\varphi$

However, this system uses nonstandard rules $(NAME)$ and (BG), and we already remarked that even in $\mathbf{L_{H\downarrow}}$ we can avoid them completely. In $\mathbf{L_{H\forall@}}$ we can eliminate these rules in favor of two additional axioms:

$Name_\exists\ \ \exists u\, u$
$Barcan_\Box\ \ \forall u\Box\varphi \leftrightarrow \Box\forall u\varphi$

This is possible also in $\mathbf{L_{H\forall}}$. Suitable axiomatic system H-$\mathbf{K_{H\forall}}$ consists of axioms and rules of H-$\mathbf{K_{H@\forall}}$ without $Barcan@$, but with:

$Nom\ \ \forall u(\Diamond^m(u \wedge \varphi) \to \Box^n(u \to \varphi)),\ m, n \in \omega$

All these axiomatizations are strongly complete for respective logics.

11.6.5 Expressivity

In fact, $\downarrow, \forall, \exists$ are not the only binders considered in hybrid languages. Below we consider briefly some strong versions of quantifiers and local binder considered in [37].

$\mathfrak{M}, a, w \models \Pi u\varphi$ iff $\mathfrak{M}, a^u_{w'}, w' \models \varphi$ for all w'
$\mathfrak{M}, a, w \models \Sigma u\varphi$ iff $\mathfrak{M}, a^u_{w'}, w' \models \varphi$ for some w'
$\mathfrak{M}, a, w \models\Downarrow u\varphi$ iff $\mathfrak{M}, a^u_{w}, w' \models \varphi$ for some w'

where a^u_w is an *u-variant of a*.

If we add to $\mathbf{L_H}$ any of these binders we obtain the following hierarchy:

$\mathbf{L_{H\downarrow}} < \mathbf{L_{H\exists}} < \mathbf{L_{H\Downarrow}}$ and $\mathbf{L_{HA}} < \mathbf{L_{H\Sigma}} < \mathbf{L_{H\Downarrow}}$ since:

$\exists u\varphi := \Downarrow v \Downarrow u(v \wedge \varphi)$, where $v \notin \varphi$

$\mathcal{E}\varphi := \Sigma u\varphi$, where $u \notin \varphi$

$\Sigma u\varphi := \Downarrow v \Downarrow u(u \wedge \varphi)$, where $v \notin \varphi$

but $\Downarrow u\varphi := \downarrow u\mathcal{E}\varphi$, so $\mathbf{L_{H\downarrow A}} = \mathbf{L_{H\Downarrow}}$

The use of these binders may be convenient but languages containing them are not stronger than $\mathbf{L_{H@\forall}}$. In the rest of this subsection we focus on the expressive strength of two central languages with binders: $\mathbf{L_{H@\downarrow}}$ and $\mathbf{L_{H@\forall}}$.

Let us look at the extension of the standard translation function ST from ordinary modal language to all hybrid languages discussed so far. Following [13] we may use a version, where nominals/state variables are identified with first-order constants/variables.

1. Standard Translation ST_t

$$
\begin{array}{lcl}
ST_t(s) & = & t = s \\
ST_t(p) & = & P(t) \\
ST_t(\neg\varphi) & = & \neg ST_t(\varphi) \\
ST_t(\varphi \wedge \psi) & = & ST_t(\varphi) \wedge ST_t(\psi) \\
ST_t(\Diamond\varphi) & = & \exists x(R(tx) \wedge ST_x(\varphi)) \\
ST_t(@_s\varphi) & = & ST_s(\varphi) \\
ST_t(\mathcal{E}\varphi) & = & \exists x ST_x(\varphi) \\
ST_t(\downarrow u\varphi) & = & \exists u(u = t \wedge ST_t(\varphi)) \\
ST_t(\exists u\varphi) & = & \exists u ST_t(\varphi) \\
ST_t(\Sigma u\varphi) & = & \exists u ST_u(\varphi) \\
ST_t(\Downarrow u\varphi) & = & \exists x \exists u(u = t \wedge ST_x(\varphi))
\end{array}
$$

where x is a variable distinct from term t and not occuring in φ.

Pure completeness of H-$\mathbf{K}^+_{\mathbf{H@\downarrow}}$ opens the question if we have something more. There are two points worth noting:

1. $\mathbf{L_{H@}}$ is more expressive (than $\mathbf{L_M}$) at the level of frames but even $\mathbf{L_{H\downarrow}}$ is more expressive at the level of models! For example, we can distinguish between reflexive and nonreflexive states in a model ($\downarrow u\Diamond u$ and $\downarrow u\neg\Diamond u$).

2. Binary temporal operators $\mathcal{U}$ (Until) and $\mathcal{S}$ (Since) are definable in $\mathbf{L_{H@\downarrow}}$ or $\mathbf{L_{TH\downarrow}}$:

- $\mathcal{U}(\varphi, \psi) :=\downarrow u\Diamond\downarrow v(\varphi \wedge @_u\Box(\Diamond v \to \psi))$
- $\mathcal{U}(\varphi, \psi) :=\downarrow uF(\varphi \wedge H(Pu \to \psi))$

Let us recall that $\mathcal{U}$ is semantically defined by the following clause:

$\mathfrak{M}, t \vDash \mathcal{U}(\varphi, \psi)$ iff $\mathfrak{M}, t' \vDash \varphi$ for some t' such that $t < t'$ and $\mathfrak{M}, t'' \vDash \psi$ for every t'' such that $t < t''$ and $t'' < t'$

Notice by the way that $\mathcal{U}$ or $\mathcal{S}$ may be locally defined also in $\mathbf{L_{TH@}}$ in the following way:

$$@_i(\mathcal{U}(\varphi, \psi) \leftrightarrow F(\varphi \wedge H(Pi \to \psi))) \tag{11.6}$$

In fact, we can establish exactly the expressive power of $\mathbf{L_{H@\downarrow}}$ on the level of models and frames. As for the first it holds [9]:

Theorem 11.10 *A formula of first-order correspondence language φ is equivalent to standard translation of a sentence in $\mathbf{L_{H@}}$ iff φ is equivalent to strongly bounded formula.*

Recall that strongly bounded fragment of first-order language covers all formulae built up from atoms with the help of boolean constants and bounded quantification (i.e. $\exists y(Rxy \wedge \varphi)$ and $\forall y(Rxy \to \varphi)$). This is the fragment of first-order language which is invariant under generated submodels. Concerning frame definability we have:

Theorem 11.11 *An elementary class of frames is defined by pure sentences of $\mathbf{L_{H@\downarrow}}$ iff it is closed under generated subframes and reflects finitely generated subframes.*

But the class of elementary frames definable in ordinary modal language is closed under generated subframes and reflects point-generated subframes, so $\mathbf{L_{H@\downarrow}}$ covers all this class (Sahlqvist formulae in particular). For example, Church-Rosser property is definable in $\mathbf{L_{H@\downarrow}}$ by pure formula:

$$\Diamond i \wedge \Diamond j \rightarrow @_i(\Diamond{\downarrow}u @_j \Diamond u) \tag{11.7}$$

On the other hand, some elementary conditions not definable in $\mathbf{L_M}$ like Predecessors ($\forall x \exists y Ryx$) are not definable by pure formulae in $\mathbf{L_{H@\downarrow}}$ either (since the class of frames with this property is not closed under generated subframes).

So far we have pointed out limitations of several hybrid languages when compared with the expressive strength of first-order language. But $\mathbf{L_{H@\forall}}$ has the full first-order language expressivity. It is obvious, because now we can directly rewrite any first order formula as a formula of $\mathbf{L_{H@\forall}}$. Formally, we can define some translation function, first introduced by Prior:

Hybrid Translation HT

$$\begin{array}{lcl} HT(Rtt') & = & @_t \Diamond t' \\ HT(Pt) & = & @_t p \\ HT(t = t') & = & @_t t' \\ HT(\neg\varphi) & = & \neg HT(\varphi) \\ HT(\varphi \wedge \psi) & = & HT(\varphi) \wedge HT(\psi) \\ HT(\exists u \varphi) & = & \exists u HT(\varphi) \end{array}$$

Because of the full first-order expressive power, in case of H-$\mathbf{K_{H@\forall}}$ we need only the basic completeness theorem.

The definition of HT makes obvious why $\mathbf{L_{H@\forall}}$ is strictly stronger than $\mathbf{L_{H\forall}}$. It is important to note that the use of satisfaction operators are essential in the translation. The addition of quantifiers to the weak hybrid language does not yield the full first-order expressivity which at first sight may seem strange.

Note in particular, that all properties not expressible as pure formulae in $\mathbf{L_{H@}}$ (e.g. Geach axioms, directedness) are expressible in $\mathbf{L_{H@\forall}}$ as PUENF-formulae (pure universal existential nominal-free formulae) of the shape: $\forall u_1, ..., u_m \exists v_1, ..., v_n \varphi$, where φ has no quantifiers, propositional variables, nominals (only state variables). e.g.:

- Church-Rosser property –
  $\forall u_1 u_2 u_3 \exists v(@_{u_1} \Diamond u_2 \wedge @_{u_1} \Diamond u_3 \rightarrow @_{u_2} \Diamond v \wedge @_{u_3} \Diamond v)$
- Predecessors – $\forall u \exists v, @_v \Diamond u$
- Right-directedness – $\forall u_1 u_2 \exists v(@_{u_1} \Diamond v \wedge @_{u_2} \Diamond v)$

Theorem 11.12 *A frame condition is defined by PUENF-formula iff it is UE-closure of strongly bounded first-order formula.*

Blackburn [40] stated a conjecture that every Sahlqvist formula is expressible by PUENF-formula. PUENF-formulae are quite interesting since they lead to stronger completeness result for H-$\mathbf{K}^{+}_{\mathbf{H@}}$ (see [41]). First note the following:

Theorem 11.13 *Every PUENF-formula PF* $\forall u_1, ..., u_m \exists v_1, ..., v_n \varphi$ *corresponds to existential saturation rule (RPF) of the form:*

If $\vdash \varphi[u_1/i_1, ..., u_m/i_m, v_1/j_1, ..., v_n/j_n] \rightarrow \psi$, *then* $\vdash \psi$,

provided $j_1, ..., j_n$ *are distinct, unequal to* $i_1, ..., i_m$ *and do not occur in* ψ

For example, for Church-Rosser property we have the rule:

If $\vdash (@_{i_1}\Diamond i_2 \wedge @_{i_1}\Diamond i_3 \rightarrow @_{i_2}\Diamond j \wedge @_{i_3}\Diamond j) \rightarrow \psi$, then $\vdash \psi$, provided $j \notin \psi$ and $j \neq i_1, i_2, i_3$

RPF-rules closely resemble Gabbay's style nonstructural rules for undefinable (in standard ML) conditions. They arise in the effect of skolemization of state variables with the help of nominals in PUENF-formulae. The relation between formulae and rules is clarified in the following:

Lemma 11.8 *If PF defines* $\mathcal{F}$, *then (RPF) is admissible in* $\mathcal{F}$

As a consequence we can prove much stronger completeness result for logics in the basic language.

Theorem 11.14 (Extended Pure Completeness) *Let* Γ *be any set of pure formulae and* R *any set of existential saturation rules, then* H-$\mathbf{K}^{+}_{\mathbf{H@}} + \Gamma + R$ *is strongly complete for the class of frames defined by* Γ *and* R.

Note that addition of existential saturation rules may also strengthen the scope of the pure completeness theorem for H-$\mathbf{K}^{+}_{\mathbf{H@\downarrow}}$.

One should note that the concept of such nonstandard rules enriching seriously the expressive power of $\mathbf{L_{H@}}$ was first independently explored on the field of tableau methods (see Blackburn [40]) and HRND (RND for hybrid logics) in Indrzejczak [148]). Some details will be given in the next Chapter.

11.7 Miscellanea

There are many developments of MHL that we did not even touch. In particular, interesting results concerning expressivity may be obtained not only by adding new constants but also by multiplying sorts of atoms. Multisorted hybrid languages and their application to analysis of linguistic phenomena of tensal discourse were investigated by Blackburn (see e.g. [31, 34]). One can find there, e.g. a hybrid formalization of interval tense logic with two sorts of nominals, denoting instants and intervals. Below we describe briefly the extension of propositional hybrid logic to first-order modal hybrid logic. We also bring together the most basic facts concerning decidability, complexity and interpolation in MHL.

11.7.1 First-Order Modal Hybrid Logic QMHL

The number of papers devoted to first-order hybrid logic is rather small but the effects of such extensions obtained so far are quite promising. The use of hybrid languages makes possible to obtain interesting results concerning the formalization of nonrigid terms and expressing various conditions put on the domains of models. Moreover, we will see that first-order hybrid logic is particularly good behaved with respect to interpolation property.

Below we present a logic **QMHL**, a first-order version of $\mathbf{L_{H@\downarrow}}$ from Blackburn [36].

1. Vocabulary of $\mathbf{L_{H@\downarrow}}$ is enriched with:

- denumerable set of first order variables $VAR = \{x, y, ...\}$
- denumerable set of rigid constants $CON = \{c_1, c_2, ...\}$
- denumerable set of nonrigid constants $FUN = \{f_1, f_2, ...\}$
- denumerable set of predicate symbols of n-arity $PRED = \{P_1, P_2, ...\}$
- first-order (possibilistic) quantifiers and equality predicate: $\forall, \exists, =$

2. the set of terms contains VAR, CON, and is closed under the rule:

- if $f \in FUN$ and $s \in NOM \cup SVAR$, then $@_s f$ is a term

Note! sat-operator is used to form both formulae and terms which is very hybrid solution indeed! In case of terms this is the way for rigidification of nonrigid terms. Let me remind you that nonrigid terms vary their denotation in different worlds, whereas rigid terms have the same denotation in all worlds. When sat-operator is attached to nonrigid term f it means: the designate of f in s, and this compound term has a constant value. This informal remark will become obvious after introduction of semantics.

3. Models are structures of the form $\mathfrak{M} = \langle \mathcal{W}, \mathcal{R}, D, V \rangle$, where D is a nonempty constant domain and V is defined as follows:

– $V(c) \in D$
– $V(i) \in \mathcal{W}$
– $V(P^n) \subseteq D^n \times \mathcal{W}$
– $V(f) \in D^{\mathcal{W}}$

An assignment $a = a_n \cup a_f$, where $a_n : SVAR \longrightarrow \mathcal{W}$ and $a_f : VAR \longrightarrow D$.

The interpretation I of the term τ in a model and under an assignment is defined as follows:

$$I(\tau) := \begin{cases} a(\tau) & \text{if } \tau \in VAR \\ V(\tau) & \text{if } \tau \in CON \\ V(f)(V(i)) & \text{if } \tau = @_i f \text{ for } i \in NOM \text{ and } f \in FUN \\ V(f)(a(v)) & \text{if } \tau = @_v f \text{ for } v \in SVAR \text{ and } f \in FUN \end{cases}$$

The new clauses for satisfaction are:

$\mathfrak{M}, a, w \vDash P^n(\tau_1, ..., \tau_n)$ iff $\langle I(\tau_1), ..., I(\tau_n), w\rangle \in V(P^n)$
$\mathfrak{M}, a, w \vDash \tau_1 = \tau_2$ iff $I(\tau_1) = I(\tau_2)$
$\mathfrak{M}, a, w \vDash \forall x\varphi$ iff $\mathfrak{M}, a^x_o, w \vDash \varphi$ for all $o \in D$
$\mathfrak{M}, a, w \vDash \exists x\varphi$ iff $\mathfrak{M}, a^x_o, w \vDash \varphi$ for some $o \in D$

The version of the semantics we have presented has the constant domain and possibilistic quantifiers just for simplicity. But we can also add the function $d : \mathcal{W} \longrightarrow \mathcal{P}(D)$ and introduce actualist quantifiers – this is easy. One can do that either indirectly in the manner described in Fitting [96], by introducing existence predicate and relativization of quantifiers to this predicate, or directly by treating actualist quantifiers as primitive. The second route is taken in Blackburn [41], where an axiomatization of **QMHL** in $\mathbf{L_{H@}}$ is presented which satisfies general pure completeness theorem. Interestingly enough, it covers not only frame conditions definable by pure axioms (and saturated rules) but also extensions obtained by considering several domain conditions, because they may be expressed by pure axioms. For example, popular conditions ordinarily defined with the help of Barcan Formula and its converse, like monotonicity or antymonotonicity are defined as follows:

$$(MON) \quad E@_i c \to \Box E@_i c$$
$$(AMON) \quad \Diamond E@_i c \to E@_i c$$

Where E is existence predicate defined in a standard way: $E\tau := \exists x, x = \tau$ and $x \neq \tau$. Constant domain is expressed even simpler by $@_i E@_j c \to @_k E@_j c$. Moreover, some other, less known, conditions may be expressed, e.g.:

Full domains	$E@_i c$
Disjoint domains	$@_i E@_j c \wedge @_k E@_j c \to @_i k$
Convex domains	$E@_i c \to \Box(\Diamond E@_i c \to E@_i c)$

Note that hybrid version of **QML** allows of simple form of representation of nonrigid terms which in ordinary modal language lead to some troubles. Here is an example: let $c =$ Caroline (rigid term in Kripke spirit) and $f =$ Miss of America (clearly nonrigid term), then the sentence "Caroline is the present Miss of America" is expressed by $\downarrow u(c = @_u f)$. One can check that:

$$\models \downarrow u(c = @_u f) \to \downarrow u G(c = @_u f)$$

but

$$\not\models \downarrow u(c = @_u f) \to G \downarrow u(c = @_u f)$$

And this is in accordance with our expectations, since the first means: "If Caroline is the present Miss of America, then it always be the case, that she is the Miss of America of now", which is obviously true. On the other hand, the second means: "If Caroline is the present Miss of America, then it always be the case, that she will be the Miss of America", which is obviously false.

11.7.2 Decidability and Complexity

During the discussion of several hybrid languages and logics, we accidentally made some remarks concerning decidability and complexity of them. It is helpful to collect these remarks and add some more in order to get a fuller picture. We consider only the question of decidability for satisfiability problem. It's easy to note that there are three possible effects of changing ordinary modal theories into hybrid theories. We can have:

1. The same complexity class e.g. $\mathbf{K_{H@}}$
2. Worse behaviour e.g. $\mathbf{Kt_{H@}}$
3. Better behaviour – logics of some frame classes.

The last point is particularly interesting and we list some striking examples.

Some concrete results.

1. Bad impact of past operators:

Even $\mathbf{Kt_H}$ with one nominal is $EXPTIME$-complete, whereas **Kt** is $PSPACE$-complete. The same applies also to monomodal hybrid logics of symmetric frames. On the other hand, an addition of @ and $\mathcal{A}$ do not change the complexity, whereas in ordinary modal language it also jumps to $EXPTIME$.

2. Transitive frames:

Hybrid modal logics of transitive frames are in $PSPACE$ even with $\mathcal{A}$ (recall that $\mathbf{K_{HA}}$ is $EXPTIME$-complete). But $\mathbf{Kt4_H}$ is still $EXPTIME$-complete.

3. Linear frames:

The best results we have for hybrid logics of linear frames. They are NP-complete even with $\downarrow$! Note that even $\mathbf{K_{H\downarrow}}$ is undecidable but on linear frames we have not only decidability but also of relatively low complexity since in this respect it is as good as **CPL**.[2]

11.7.3 Interpolation and Beth Definability

Hybrid logics show also remarkable advantages over standard modal logics with respect to interpolation properties. Let's recall the basic definitions in the form suitable for hybrid languages.

[2]Perhaps we should rather say as bad as **CPL**, if we remember that only P-complete problems are considered as practically tractable.

Definition: L has the Strong Interpolation property iff: $\models_L \varphi \rightarrow \psi$ implies that $\models_L \varphi \rightarrow \chi$ and $\models_L \chi \rightarrow \psi$ for some χ such that $P(\chi) \subseteq P(\varphi) \cap P(\psi)$.

Definition: L has the Weak Interpolation property iff: $\varphi \mid\models \psi$ implies that $\varphi \mid\models_L \chi$ and $\chi \mid\models_L \psi$ for some χ such that $P(\chi) \subseteq P(\varphi) \cap P(\psi)$.

Note that we can obtain several forms of interpolation property for MHL if we change the meaning of P. If it is the set of propositional variables, it is a standard notion from ML, but we can consider also the version where P covers additionally the set of nominals, or of only nominals.

The relation between the two concepts is the following:

Theorem 11.15 *If $\models$ is compact, then strong interpolation implies weak interpolation.*

For the two most important basic hybrid logics we have:

Theorem 11.16 $\mathbf{K_{H@\downarrow}}$ *has strong interpolation;* $\mathbf{K_{H@}}$ *has only weak interpolation.*

An example: there is no interpolant for $i \wedge \Diamond i \rightarrow (j \rightarrow \Diamond j)$ but if we limit P to propositional variables only, then strong interpolation holds also for $\mathbf{K_{H@}}$ and for $\mathbf{K_H}$. These results extend to Beth definability in case of $\mathbf{K_{H@}}$, since in order to derive this property we need only interpolation with P limited to propositional variables. But, surprisingly enough, it does not hold for $\mathbf{K_H}$ (see [273])!

There is an interesting relation between decidability and interpolation which is expressed by the following two theorems:

Theorem 11.17 *Every hybrid logic formulated in extension of* $\mathbf{L_H}$ *either is decidable or has strong interpolation (over nominals).*

Theorem 11.18 $\mathbf{K_{H@\downarrow}}$ *is the least logic with strong interpolation; any extension axiomatizable by a set of nominal-free sentences also has this property.*

A good behavior of **QMHL** in $\mathbf{L_{H@\downarrow}}$ is particularly worth mentioning. The following theorem holds:

Theorem 11.19 *Strong interpolation (and Beth definability) holds for any* **QMHL** *between* **K** *and* **S5**.

This is in strong contrast to ordinary **QML** where we have the following negative result due to Fine:

Theorem 11.20 *Interpolation fails for any* **QML** *between* **K** *and* **S5** *with constant domains and for* **S5** *with varying domains.*

Bibliography

[1] Abadi, M., and Z. Manna. 1990. Nonclausal deduction in first-order temporal logic. *Journal of the Association for Computing Machinery* 37: 279–317.

[2] D'Agostino, M. 1999. Tableau methods for classical propositional logic. In *Handbook of tableau methods*, eds. M. D'Agostino et al., 45–123. Dordrecht: Kluwer Academic Publishers.

[3] Aho, A.V., and J.D. Ullman. 1995. *Foundations of computer science.* New York: W.H. Freeman and CO.

[4] Allen, J. 1984. Towards a general theory of knowledge and action. *Artificial Intelligence* 23: 123–154.

[5] Anderson, A., and N.D. Belnap, Jr. 1975. *Entailment: The logic of relewance and necessity*, vol I. Princeton: Princeton University Press.

[6] Anderson, A., and H. Johnstone. 1962. *Natural deduction, the logical basis of axiom systems.* Belmont: Wadsworth.

[7] Andou, Y. A note on modal logic S4 in natural deduction. Available on www.hosei.ac.jp/museum/html/kiyo/58/articles/Andou.pdf

[8] Areces, C., P. Blackburn, and M. Marx. 2000. The computational complexity of hybrid temporal logics. *Logic Journal of the IGPL* 8(5): 653–679.

[9] Areces, C., P. Blackburn, and M. Marx. 2001. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic* 66: 977–1010.

[10] Areces, C., H. DeNivelle, and M. DeRijke. 2001. Resolution in modal, description and hybrid logic. *Journal of Logic and Computation* 11: 717–736.

A. Indrzejczak, *Natural Deduction, Hybrid Systems and Modal Logics*,
Trends in Logic 30, DOI 10.1007/978-90-481-8785-0,

[11] Areces, C., and J. Heguiabehere. HyLoRes: Direct resolution for hybrid logics. Available on Hybrid Logic Homepage.

[12] Areces, C., D. Gorin. 2005. Ordered resolution with selection for h(@). In *Proceedings of LPAR 2004*, eds. F. Baader and A. Voronkov, 125–141, LNCS 3452.

[13] Areces, C., B. ten Cate. 2006. Hybrid logics. In *Handbook of modal logic*, eds. P. Blackburn, J. van Benthem, and F. Wolter, 821–868. Amsterdam: Elsevier.

[14] Arlo-Costa, H., and E. Pacuit. 2006. First-order classical modal logic. *Studia Logica* 84: 171–210.

[15] Avron, A. 1993. Gentzen-type systems, resolution and tableaux. *Journal of Automated Reasoning* 10(2): 265–281.

[16] Avron, A. 1996. The method of hypersequents in the proof theory of propositional non-classical logics. In *Logic: From foundations to applications*, eds. W. Hodges et al., 1–32. Oxford: Oxford Science Publication.

[17] Avron, A., F. Honsell, M. Miculan, and C. Paravano. 1998. Encoding modal logics in logical frameworks. *Studia Logica* 60: 161–202.

[18] Baldoni, M. 1998. *Normal multimodal logics: Automatic deduction and logic programming extension* PhD thesis, Torino.

[19] Baldoni, M. 2000. Normal multimodal logics with interaction axioms. In *Labelled deduction* vol 17 of Applied Logic Series, eds. D. Basin et al., 33–53. Dordrecht: Kluwer.

[20] Barth, E.M., E.C. Krabbe. 1982. *From Axiom to Dialogue*. Berlin: Walter de Gruyter.

[21] Basin, D., S. Matthews, and L. Vigano. 1997. Labelled propositional modal logics: Theory and practice. *Journal of Logic and Computation* 7(6): 685–717.

[22] Basin, D., S. Matthews, and L. Vigano. 1998. Natural deduction for non-classical logics. *Studia Logica* 60: 119–160.

[23] Basin, D., S. Matthews 2002. Logical frameworks. In *Handbook of philosophical logic*, eds. D. Gabbay, and F. Guenthner, 2nd ed., vol IX, 89–163. New York: Springer.

[24] Belnap, N.D. 1982. Display logic. *Journal of Philosophical Logic* 11: 375–417.

[25] Bencivenga, E. 1986. Free logics. In *Handbook of Philosophical Logic*, eds. D. Gabbay, and F. Guenthner, vol III, 373–426. Dordrecht: Reidel Publishing Company.

[26] Ben-Sasson, E., and A. Widgerson. 2001. Short proofs are narrow – resolution made simple. *Journal of the ACM* 48(2): 149–169.

[27] Beth, E. 1955. Semantic entailment and formal derivability. *Mededelingen der Kon. Ned. Akad. v. Wet.* 18(13).

[28] Beth, E.W. 1959. The foundations of mathematics. Amsterdam: North Holland.

[29] Blackburn, P. 1990. *Nominal tense logic and other sorted intensional frameworks*, Ph.D thesis, Centre for Cognitive Science, University of Edinburgh.

[30] Blackburn, P. 1992. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 34: 56–83.

[31] Blackburn, P. 1994. Tense, temporal reference and tense logic. *Journal of Semantics* 11: 83–101.

[32] Blackburn, P. 2000. Internalizing labelled deduction. *Journal of Logic and Computation* 10(1): 137–168.

[33] Blackburn, P. 2000. Representation, reasoning, and relational structures: A hybrid logic manifesto. *Logic Journal of the IGPL* 8(3): 339–365.

[34] Blackburn, P. 2001. Fine grained theories of time. In *Non-classical logic*, ed. H. Wansing. Cambridge: King's College University Press.

[35] Blackburn, P., M. DeRijke, and Y. Venema. 2001. *Modal logic*. Cambridge: Cambridge University Press.

[36] Blackburn, P., and M. Marx. 2002. Tableaux for quantified hybrid logic. In *Tableaux 2002, LNAI 2381,* eds. U. Egly, and C. Fermuller, 38–52.

[37] Blackburn, P., and J. Seligman. 1995. Hybrid languages. *Journal of Logic, Language and Information* 4: 251–272.

[38] Blackburn, P., and M. Tzakova. 1998. Hybrid completeness. *Logic Journal of the IGPL* 6(4): 625–650.

[39] Blackburn, P., and M. Tzakova. 1999. Hybrid languages and temporal logic. *Logic Journal of the IGPL* 7(1): 27–54.

[40] Blackburn, P., and B. ten Cate. 2002. Beyond pure axioms: Node creating rules in hybrid tableaux. In *Hybrid logics*, eds. C. Areces et al., 21–35.

[41] Blackburn, P., and B. ten Cate. 2006. Pure extensions, proof rules and hybrid axiomatics. *Studia Logica* 75(3): 345–376.

[42] Blamey, S., and L. Humberstone. 1991. A perspective on modal sequent logic. *Publications of the Research Institute for Mathematical Sciences, Kyoto University* 27: 763–782.

[43] Bocheński, I.M. 1956. *Formale Logik*. Freiburg/München: Alber.

[44] Bolander, T. and T. Braüner. 2007. Tableau-based decision procedures for hybrid logic. *Journal of Logic and Computation* 16: 737–763.

[45] Bolander, T. and P. Blackburn. 2007. Termination for hybrid tableaus. *Journal of Logic and Computation* 17(3): 517–554.

[46] Bolotov, A., V. Bocharov, A. Gorchakov, and V. Shangin. 2005. Automated first order natural deduction. In *Proceedings of IICAI*, 1292–1311.

[47] Bolotov, A., A. Basukoski, O. Grigoriev, and V. Shangin. 2006. Natural deduction calculus for linear-time temporal logic. *Proceedings of jelia 2006*. LNAI, Springer 4160.

[48] Bolotov, A., O. Grigoriev, and V. Shangin. 2006. Natural deduction calculus for CTL. *IEEE John Vincent Atanasoff Symposium on Modern Computing*, 175–183.

[49] Bonette, N., and R. Goré. 1998. A labelled sequent system for tense logic Kt. In *AI98: Proceedings of the Australian joint conference on artificial inteligence*, LNAI Springer, 1502: 71–82.

[50] Boolos, G. 1979. *The unprovability of consistency*. Cambridge: Cambridge University Press.

[51] Boolos, G. 1984. Don't eliminate cut. *Journal of Philosophical Logic* 7: 373–378.

[52] Bonevac, D. 1987. *Deduction*, Mountain View: Mayfield Press.

[53] Boričić, B.R. 1985. On sequence-conclusion natural deduction systems. *Journal of Philosophical Logic* 14: 359–377.

[54] Borkowski, L., and J. Słupecki. 1958. A logical system based on rules and its applications in teaching mathematical logic. *Studia Logica* 7: 71–113.

[55] Braüner, T. 2004. Natural deduction for hybrid logic. *Journal of Logic and Computation* 14(3): 329–353.

[56] Braüner, T. 2009. Hybrid logic and its proof theory. *Computer science research report* 124, Roskilde: Roskilde University.

[57] Braüner, T. 2009. Hybrid logic. In *Handbook of philosophical logic*, 2nd ed., vol XVI. New York: Springer, to appear.

[58] Bull, R. 1970. An approach to tense logic. *Theoria* 36: 282–300.

[59] Bull, R., and K. Segerberg. 1984. Basic modal logic. In *Handbook of Philosophical Logic*, eds. D. Gabbay, and F. Guenthner, vol II, 1–88. Dordrecht: Reidel Publishing Company.

[60] Burgess, J.P. 1984. Basic tense logic. In *Handbook of philosophical logic*, eds. D. Gabbay, and F. Guenthner, vol II, 89–133. Dordrecht: Reidel Publishing Company.

[61] Carnielli, W.A. 1991. On sequents and tableaux for many-valued logics. *Journal of Non-Classical Logic* 8(1): 59–76.

[62] Castellini, C. 2005. *Automated reasoning in quantified modal and temporal logic*, PhD thesis, University of Edinburgh.

[63] Castellini, C. and A. Smaill. 2000. A systematic presentation of quantified modal logics. *Logic Journal of the IGPL* 10: 571–599.

[64] Catach, L. 1991. TABLEAUX: A general theorem prover for modal logics. *Journal of Automated Reasoning* 7(4): 489–510, 1991.

[65] Cellucci, C. 1992. Existential instatiation and normalization in sequent natural deduction. *Annals of Pure and Applied Logic* 58: 111–148.

[66] Cerrato, C. 1994. Natural deduction based upon strict implication for normal modal logics. *Notre Dame Journal of Formal Logic* 35(4): 471–495.

[67] Chagrov, A., and M. Zakharyaschev. 1997. *Modal logic*. Oxford: Oxford University Press.

[68] Chang, C.L., and R.C.T. Lee. 1973. *Symbolic logic and mechanical theorem proving*. Orlando: Academic Press.

[69] Chellas, B. 1980. *Modal logic*. Cambridge: Cambridge University Press.

[70] Church, A. 1956. *Introduction to Mathematical Logic*, vol I. Princeton: Princeton University Press.

[71] Copeland, J.B. 2002. The genesis of possible world semantics. *Journal of Philosophical Logic* 31: 99–137.

[72] Copi, I.M. 1954. *Symbolic logic*, New York: The Macmillan Company.

[73] Corcoran, J. 1972. Aristotle's natural deduction system. In *Ancient logic and its modern interpretations*, ed. J. Corcoran. Dordrecht: Reidel.

[74] Corcoran, J., and G. Weaver. 1969. Logical consequence in modal logic: Natural deduction in S5. *Notre Dame Journal of Formal Logic* 10: 370–384.

[75] Curry, H.B. 1950. *A theory of formal deducibility*. Notre Dame: University of Notre Dame Press.

[76] Curry, H.B. 1952. The elimination theorem when modality is present. *Journal of Symbolic Logic* 17: 249–265.

[77] Davis, M., and H. Putnam. 1960. A computing procedure for quantification theory. *Journal of the Association for Computing Machinery* 7: 201–215.

[78] Demri, S. 1999. Sequent calculi for nominal tense logics: A step towards mechanization¿ In *Tableaux 99, LNAI 1617*, ed. N. Murray, 140–154. Berlin: Springer.

[79] Demri, S., and R. Goré. 1999. Cut-free display calculi for nominal tense logics. In *Tableaux 99, LNAI 1617*, ed. N. Murray, 155–170. Berlin: Springer.

[80] Došen K. 1985. Sequent-systems for modal logic. *Journal of Symbolic Logic* 50: 149–159.

[81] Dyckhoff, R. 1987. Implementing a simple proof assistant. In *Proceedings of workshop on programming for logic teaching,* eds. J. Derrick, and H.A. Lewis, 49–59.

[82] Ebbinghaus, H.D., J. Flum, and W. Thomas. 1984. *Mathematical logic.* Berlin: Springer.

[83] van Eijck, J. 2002. *Constraint tableaux for hybrid logics.* Amsterdam: Manuscript CWI.

[84] Farinas del Cerro, L., and A. Herzig. 1995. Modal deduction with applications in epistemic and temporal logics. In *Handbook of logic in artificial intelligence and logic programming*, eds. D. Gabbay et al., vol IV, 499–594. Oxford: Clarendon Press.

[85] Feys, R. 1950. Les systemes formalises des modalites. *Revue philosophicue de Louvain* 48: 478–509.

[86] Feys, R., and J. Ladriere. 1955. Supplementary notes. In *Recherches sur la deduction logique*, french translation of Gentzen, Paris: Press Univ. de France.

[87] Fine, K. 1985. *Reasoning with arbitrary objects.* Oxford: Blackwell.

[88] Fisher, M., C. Dixon, and M. Peim. 2001. Clausal temporal resolution. *ACM Transactions on Computational Logic* 1(4).

[89] Fitch, F. 1952. *Symbolic logic.* New York: Ronald Press Co.

[90] Fitch, F. 1966. Tree proofs in modal logic (abstract). *The Journal of Symbolic Logic* 31:152.

[91] Fitch, F. 1966. Natural deduction rules for obligation. *American Philosophical Quaterly* 3:27–38.

[92] Fitting, M. 1972. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic* 13(2): 237–247.

[93] Fitting, M. 1983. *Proof methods for modal and intuitionistic logics.* Dordrecht; Rei-del.

[94] Fitting, M. 1990. Destructive modal resolution. *Journal of Logic and Computation* 1(1): 83–97.

[95] Fitting, M. 1996. *First-order logic and automated theorem proving.* Berlin; Springer.

[96] Fitting, M., and R.L. Mendelsohn. 1998. *First-order modal logic.* Dordrecht: Kluwer.

[97] Fitting, M. 2007. Modal proof theory. In *Handbook of modal logic*, eds. P. Blackburn, J.F.A.K van Benthem, and F. Wolter, 85–138. Amsterdam: Elsevier.

[98] Gabbay, D. 1981. An irreflexivity lemma with applications to axiomatizations of conditions on linear frames. In *Aspects of philosophical logic*, ed. U. Munnich, 67–89. Dordrecht: Reidel.

[99] Gabbay, D. 1996. *LDS-labelled deductive systems.* Oxford: Clarendon Press.

[100] Gabbay, D., A. Kurucz, F. Wolter, and M. Zakharyaschev, *Many-dimensional modal logics: Theory and applications.* Amsterdam: Elsevier.

[101] Gallier, J.H. 1986. *Logic for computer science.* New York: Harper and Row.

[102] Gargov, G., and V. Goranko. 1993. Modal logic with names. *Journal of Philosophical Logic* 22: 607–636.

[103] Garson, J.W. 1984. Quantification in modal logic. In *Handbook of philosophical logic*, eds. D. Gabbay, and F. Guenthner, vol II, 249–307. Dordrecht: Reidel Publishing Company.

[104] Garson, J.W. 2005. Unifying quantified modal logic. *Journal of Philosophical Logic* 34: 621–649.

[105] Garson, J.W. 2006. *Modal logic for philosophers*. Cambridge: Cambridge University Press.

[106] Gasquet, O. 1996. Completeness results in neighbourhood semantics for monotonic and regular logics. *Journal of the IGPL* 4(3): 417–426.

[107] Gasquet, O., and A. Herzig. 1996. From classical to normal modal logics. In *Proof Theory for Modal Logic*, ed. H. Wansing. Dordrecht: Kluwer.

[108] Gentzen, G. 1932. Über die Existenz Unabhängiger Axiomensysteme zu Unendlichen Satzsystemen. *Mathematische Annalen* 107: 329–350.

[109] Gentzen, G. 1934. Untersuchungen über das Logische Schliessen. *Mathematische Zeitschrift* 39: 176–210, 405–431.

[110] Gentzen, G. 1936. Die Widerspruchsfreiheit der Reinen Zahlentheorie. *Mathematische Annalen* 112: 493–565.

[111] Girle, R. 2000. *Elementary modal logic*. London: Acumen.

[112] Goldblatt, R.I. 1987. *Logics of time and computation*. Stanford: CSLI Lecture Notes.

[113] Goranko, V. 1994. Refutation systems in modal logic. *Studia Logica* 53(2): 229–234.

[114] Goranko, V. 1994. Temporal logic with reference pointers. In *Temporal logic*, LNiAI 827, 133–148. New York: Springer.

[115] Goranko, V., D. Vakarelov. 2001. Sahlqvist formulas in hybrid polyadic modal logics. *Journal of Logic and Computation* 11(5): 737–754.

[116] Goré, R. 1992. *Cut-free sequent and tableau systems for propositional normal modal logics*, PhD thesis, University of Cambridge.

[117] Goré, R. 1999. Tableau methods for modal and temporal logics. In *Handbook of tableau methods*, eds. M. D'Agostino et al., 297–396. Dordrecht: Kluwer Academic Publishers.

[118] Goubault-Larrecq J., and P.H. Schmitt. 1997. A tableau system for linear temporal logic. *Proceedings of TACAS'97*, 130–144, Springer.

[119] Hähnle, R. 1994. *Automated deduction in multiple-valued logics*. Oxford: Oxford University Press.

[120] Hähnle, R. 2001. Tableaux and related methods. In *Handbook of automated reasoning*, eds. A. Robinson, and A. Voronkov, pp. 101–177. Amsterdam: Elsevier.

[121] D'Agostino, M. et al. (eds.). 1999. *Handbook of tableau methods*. Dordrecht: Kluwer Academic Publishers.

[122] Hawthorn, J. 1990. Natural deduction in normal modal logic. *Notre Dame Journal of Formal Logic* 31: 263–273.

[123] Hazen, A. 1990. Actuality and quantification. *Notre Dame Journal of Formal Logic* 31: 498–508.

[124] Hansen, H.H. 2003. *Monotonic modal logics*. MA thesis, University of Amsterdam.

[125] Hemaspaandra, E. 1996. The price of universality. *Notre Dame Journal of Formal Logic* 37: 174–203.

[126] Herbrand, J. 1928. Abstract In *Comptes Rendus des Seances de l'Academie des Sciences*, vol. 186, 1275 Paris.

[127] Herbrand, J. 1930. Recherches sur la theorie de la demonstration. In *Travaux de la Societe des Sciences et des Lettres de Varsovie, Classe III, Sciences Mathematiques et Physiques, Warsovie*, 1930.

[128] Hermes, H. 1963. *Einführung in die Mathematische Logik*. Stuttgart: Teubner.

[129] Hertz, P. 1929. Über Axiomensysteme für Beliebige Satzsysteme. *Mathematische Annalen* 101: 457–514.

[130] Heuerding, A., M. Seyfried, and H. Zimmermann. 1996. Efficient loop-check for backward proof search in some non-classical propositional logics. In *Tableaux 96, LNCS 1071*, eds. P. Miglioli et al., 210–225. Berlin: Springer.

[131] Hintikka, J. 1955. Form and content in quantification theory. *Acta Philosophica Fennica* 8: 8–55.

[132] Hintikka, J. 1957. Quantifiers in deontic logic. *Societas Scientiarum Fennica, Commentationes Humanarum Literarum XXIII.*

[133] Horrocks, I. 1997. *Optimising tableaux decision procedures for description logics*, PhD Thesis, University of Manchester.

[134] Horrocks, I., U. Satler, and S. Tobies. 2000. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL* 8(3): 239–263.

[135] Hughes, G.E., and M.J. Cresswell. 1968. *Introduction to modal logic.* London: Methuen.

[136] Hughes, G.E., and M.J. Cresswell. 1996. A *new introduction to modal logic.* London: Routledge.

[137] Hustadt, U., and R.A. Schmidt. 2000. Issues of decidability for description logics in the framework of resolution. In *Automated deduction in classical and non-classical logics*, eds. R. Caferra, and G. Salzer, 192–206. New York: Springer.

[138] Hustadt, U., and R.A. Schmidt. 2002. Using resolution for testing modal satisfiability and building models. *Journal of Automated Reasoning* 28(2): 205–232.

[139] Indrzejczak, A. 1994. Natural deduction in quantifier modal logic. *Bulletin of the Section of Logic* 23(1): 30–40.

[140] Indrzejczak, A. 1994. Natural deduction system for tense logics. *Bulletin of the Section of Logic* 23(4): 173–179.

[141] Indrzejczak, A. 1995. Dedukcja naturalna w logikach modalnych pierwszego rzędu. In *Filozofia/Logika: Filozofia logiczna 1994*, eds. J. Perzanowski, A. Pietruszczak, and C. Gorzka, 289–302, Wyd. Toruń: Uniwersytetu Mikołaja Kopernika.

[142] Indrzejczak, A. 1996. Cut-free sequent calculus for S5. *Bulletin of the Section of Logic* 25(2): 95–102.

[143] Indrzejczak, A. 1997. Generalised sequent calculus for propositional modal logics. *Logica Trianguli* 1: 15–31.

[144] Indrzejczak, A. 1998. Jaśkowski and Gentzen approaches to natural deduction and related systems. In *The Lvov-Warsw school and contemporary philosophy*, eds. K. Kijania-Placek and J. Woleński, 253–264, Dordrecht: Kluwer.

[145] Indrzejczak, A. 1999. A survey of natural deduction systems for modal logics. *Logica Trianguli* 3: 55–84.

[146] Indrzejczak, A. 2000. Multiple sequent calculus for tense logics. *Abstracts of AiML and ICTL* 2000: 93–104, Leipzig.

[147] Indrzejczak, A. 2001. Decision procedure for modal logics in natural deduction setting. Unpublished version, Łódź.

[148] Indrzejczak, A. 2002. Hybrid system for (not only) hybrid logic. *Abstracts of AiML* 2002: 137–156, Toulouse.

[149] Indrzejczak, A. 2002. Labelled analytic tableaux for S4.3. *Bulletin of the Section of Logic* 31(1): 15–26.

[150] Indrzejczak, A. 2002. Resolution based natural deduction. *Bulletin of the Section of Logic* 31(3): 159–170.

[151] Indrzejczak, A. 2003. A labelled natural deduction system for linear temporal logic. *Studia Logica* 75(3): 345–376.

[152] Indrzejczak, A. 2005. Sequent calculi for monotonic modal logics. *Bulletin of the Section of logic* 34(3): 151–164.

[153] Indrzejczak, A. 2006. *Introduction to the methodology of sequent calculi.* Unpublished manuscript in polish.

[154] Indrzejczak, A. 2007. Labelled tableau calculi for weak modal logics. *Bulletin of the Section of logic* 36(3–4): 159–173.

[155] Indrzejczak, A. 2007. Modal hybrid logic. *Logic and Logical Philosophy* 16: 147–257.

[156] Jaśkowski, S. 1929. Teoria dedukcji oparta na dyrektywach założeniowych. In *Księga Pamiątkowa I Polskiego Zjazdu Matematycznego.* Kraków: Uniwersytet Jagielloņski.

[157] Jaśkowski, S. 1934. On the rules of suppositions in formal logic. *Studia Logica* 1: 5–32.

[158] Kalish, D., and R. Montague. 1957. Remarks on descriptions and natural deduction. *Archiv. für Mathematische Logik und Grundlagen Forschung* 3: 50–64, 65–73.

[159] Kalish, D., and R. Montague. 1964. *Logic, techniques of formal reasoning.* New York: Harcourt, Brace and World.

[160] Kanger, S. 1957. *Provability in Logic*, Stockholm: Almqvist & Wiksell.

[161] Kashima, R. 1994. Cut-free sequent calculi for some tense logics. Studia Logica 53: 119–135.

[162] Kleene, S.C. 1952. *Introduction to metamathematics*. Amsterdam: North Holland.

[163] Kneale, W., and M. Kneale. 1962. The development of logic. Oxford: Clarendon Press.

[164] Konikowska, B. 1997. A logic for reasoning about relative similarity. *Studia Logica* 58(1): 185–226.

[165] Kracht, M. 1996. Power and weakness of the modal display calculus. In *Proof theory of modal logic*, ed. H. Wansing, 93–121. Dordrecht: Kluwer.

[166] Kracht, M. 1999. *Tools and techniques in modal logic.* Amsterdam: Elsevier.

[167] Kracht, M., and F. Wolter. 1999. Normal monomodal logics can simulate all others. *Journal of Symbolic Logic* 64(1): 99–138.

[168] Krajicek, J. 1994. Lower bounds to the size of constant-depth propositional proofs. *Journal of Symbolic Logic* 59(1): 73–85.

[169] Kripke, S. 1959. A completeness theorem in modal logic. *Journal of Symbolic Logic* 24: 1–14.

[170] Kripke, S. 1963. Semantical analysis of modal logic I. *Zeitschrift f´ur Mathematische Logik und Grundlegen der Mathematik* 9: 67–96.

[171] Kripke, S. 1980. *Naming and necessity.* Cambridge: Harvard University Press.

[172] Lavendhomme, R., and T. Lucas. 2000. Sequent calculi and decision procedures for weak modal systems. *Studia Logica* 65: 121–145.

[173] Lemmon, E.J. 1957. New foundations for Lewis modal systems. *Journal of Symbolic Logic* 22: 176–186.

[174] Lemmon, E.J. 1965. *Beginning logic*. London: Nelson.

[175] Leszczyńska, D. 2008. The method of Socratic proofs for modal propositional logics: K5, S4.2, S4.3, S4M, S4F, S4R anf G. *Studia Logica* 89(3): 365–399.

[176] Lewis, D. 1968. Counterpart theory and quantified modal logic. *Journal of Philosophy* 65: 113–126.

[177] Lichtenstein, O., and A. Pnueli. 2000. Propositional temporal logics: Decidability and completeness. *Logic Journal of the IGPL* 8(1): 55–85.

[178] Lis, Z. 1960. Wynikanie semantyczne a wynikanie formalne. *Studia Logica* 10: 39–60.

[179] Loveland, D.W. 1978. *Automated theorem proving: A logical basis.* Amsterdam: North Holland.

[180] Łukasiewicz, J. 1951. *Aristotle's syllogistic from the standpoint of modern logic.* Oxford: Clarendon Press.

[181] Marciszewski, W., and R. Murawski. 1995. *Mechanization of reasoning in a historical perspective.* Amsterdam, Atlanta: Rodopi.

[182] Marx, M., S. Mikulas, and S. Schlobach. 1999. Tableau calculus for local cubic modal logic and it's implementation. *Logic Journal of the IGPL* 7(6): 755–778.

[183] Marx, M., S. Mikulas, and M. Reynolds. 2000. The mosaic method for temporal logics. In *Automated reasoning with analytic tableaux and related methods*, ed. R. Dyckhoff, Proc. of International Conference TABLEAUX 2000, Saint Andrews, Scotland, LNAI 1847, 324–340. New York: Springer.

[184] Martins, A.T., and L.R. Martins. 2008. Full classical S5 in natural deduction with weak normalization. *Annals of Pure and Applied Logic* 152: 132–147.

[185] Massacci, F. 1994. Strongly analytic tableaux for normal modal logics. In *Proc. CADE-12*, ed. A. Bundy, LNAI 814: 723–737. New York: Springer.

[186] Massacci, F. 1998. Single step tableaux for modal logics: Methodology, computations, algorithms. *Technical Report TR-04*, Dipartimento di Informatica e Sistemistica, Universita di Roma "La Sapienza".

[187] Mates, B. 1953. *Stoic logic*. Berkeley: University of California Press.

[188] McCarthy, J., and P. Hayes. 1969. Some philosophical problems from the standpoint of artificial inteligence. In *Machine Intelligence 4*, eds. B. Meltzer, and D. Michie. Edinburgh: Edinburgh University Press.

[189] Medeiros, M., and Da Paz N. 2006. A new S4 classical modal logic in natural deduction. *Journal of Symbolic Logic* 71(3): 799–809.

[190] Mints, G. 1970. Cut-free calculi of the S5 type. *Studies in Constructive Mathematics and Mathematical Logic* II: 79–82.

[191] Mints, G. 1997. Indexed systems of sequents and cut-elimination. *Journal of Philosophical Logic* 26: 671–696.

[192] Mouri, M. 2001. Theorem provers with countermodels and xpe. *Bulletin of the Section of Logic* 30(2): 79–86.

[193] Negri, S., and J. von Plato. 2001. *Structural proof theory*, Cambridge: Cambridge University Press.

[194] Negri, S. 2005. Proof analysis in modal logic. *Journal of Philosophical Logic* 34: 507–544.

[195] Nishimura, H. 1980. A study of some tense logics by Gentzen's sequential method. *Publications of the Research Institute for Mathematical Sciences, Kyoto University* 16: 343–353.

[196] De Nivelle, H., R.A. Schmidt, and U. Hustadt. 2000. Resolution-based methods for modal logics. *Logic Journal of the IGPL* 8(3): 265–292.

[197] Ohnishi, M., and K. Matsumoto. 1957. Gentzen method in modal calculi I. *Osaka Mathematical Journal* 9: 113–130.

[198] Ohnishi, M., and K. Matsumoto. 1959. Gentzen method in modal calculi II. *Osaka Mathematical Journal* 11: 115–120.

[199] Øhrstrøm, P., and P. Hasle. 1993. A.N. Prior rediscovery of tense logic. *Erkenntnis* 39: 23–50.

[200] Ono, H., and A. Nakamura. 1980. On the size of refutation Kripke models for some linear modal and tense logics. *Studia Logica* 39: 325–333.

[201] Orłowska, E. 1987. Relational interpretation of modal logics. *Technical Report* ICSPas.

[202] Passy, S., and T. Tinchev. 1985. PDL with data constants. *Information Processing Letters* 20: 35–41.

[203] Passy, S., and T. Tinchev. 1991. An essay in CDL. *Information and Computation* 93: 263–332.

[204] Pelletier, F.J. 1998. Automated natural deduction in THINKER. *Studia Logica* 60: 3–43.

[205] Pelletier, F.J. 1999. A brief history of natural deduction. *History and Philosophy of Logic* 20: 1–31.

[206] Perzanowski, J. 1973. The deduction theorems for the modal propositional calculi formalised after the manner of Lemmon I. *Reports on Mathematical Logic* 1: 1–12.

[207] Perzanowski, J. 1976. The deduction theorems for the modal propositional calculi formalised after the manner of Lemmon II. *Reports on Mathematical Logic* 1: 1–12.

[208] Perzanowski, J. 1989. Logiki modalne a filozofia. In *Jak filozofować*, ed. J. Perzanowski, 262–346. Warszawa: PWN.

[209] Perzanowski, J. 1999. Combination semantics for intensional logics I. *Logique et Analyse* 165–166: 181–203.

[210] Perzanowski, J. 2004. Towards combination metaphysics. *Reports on Mathematical Logic* 38: 93–116.

[211] Plaisted, D.A., and Y. Zhu. 1997. *The efficiency of theorem proving strategies: A comparative and assymptotic analysis.* Braunschweig: Vieweg Verlag.

[212] Plato von, J. 2005. Normal derivability in modal logic. *Mathematical Logic Quaterly* 54(6): 632–638.

[213] Plato von, J. 2008. Gentzen's prof of normalization for ND. *The bulletin of symbolic logic* 14(2): 240–257.

[214] Poggiolesi, F. 2008. *Sequent calculi for modal logics*, Ph.D Thesis, Florence.

[215] Pogorzelski, W.A. 1973. *Klasyczny rachunek zdań.* Warszawa: PWN.

[216] Pollock, J.L. 1992. Interest driven suppositional reasoning. *Journal of Automated Reasoning* 6: 419–462.

[217] Pollock, J.L. Natural deduction. Available on http://oscarhomme.socsci.arizona.edu/ftp/publications.html

[218] Pollock, J.L. *Logic: An introduction to the formal study of reasonong.* Available on *http://oscarhomme.socsci.arizona.edu/ftp/publications.html*

[219] Portoraro, D.F. 1998. Strategic construction of Fitch-style proofs. *Studia Logica* 60: 45–67.

[220] Prawitz, D. 1965. *Natural deduction.* Stockholm: Almqvist and Wiksell.

[221] Prawitz, H., D. Prawitz, and N. Voghera. 1960. A mechanical proof procedure and its realization in an electronic computer. *Journal of the Association for Computing Machinery* 7: 102–128.

[222] Priest, G. 2001. *An introduction to non-classical logic.* Cambridge: Cambridge University Press.

[223] Prior, A. 1957. *Time and modality.* Oxford: Oxford University Press.

[224] Prior, A. 1967. *Past, present and future.* Oxford: Clarendon/Oxford University Press.

[225] Prior, A. 1968. *Papers on time and tense.* Oxford: Clarendon/Oxford University Press.

[226] Quine W. Van O. 1950. *Methods of logic.* New York: Colt.

[227] Raggio, A. 1965. Gentzen's hauptsatz for the systems NI and NK. *Logique et Analyse* 8: 91–100.

[228] Rasiowa, H., and R. Sikorski. 1963. *The mathematics of metamathematics*. Warszawa: PWN.

[229] Rautenberg, W. 1983. Modal tableau calculi and interpolation. *Journal of Philosophical Logic* 12: 403–423.

[230] Renteria, C., and E. Hausler. 2002. Natural deduction for CTL. *Bulletin of the section of logic* 31(4): 231–240.

[231] Rescher, N., and A. Urquhart. 1971. *Temporal logic*. New York: Springer-Verlag.

[232] Restall, G. *Proof theory and philosophy*. Available on *http://consequently.org/writing/ptp*

[233] Ridder, J. 1955. Die Gentzenschen Schlussverfahre in Modalen Aussagenlogiken. *Indagationes Mathematicae* 17: 163–276.

[234] Rieger, L. 1967. *Algebraic methods of mathematical logic*. Prague: Academia.

[235] Robinson, J.A. 1965. A machine oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery* 12: 23–41.

[236] Rousseau, G. 1967. Sequents in many valued logic. *Fundamenta Mathematicae*, LX 1: 22–23.

[237] Russo, A. 1995. Modal labelled deductive systems. Dept. of Computing, Imperial College, London, Technical Report 95(7).

[238] Sambin, G., and S. Valentini. 1980. A modal sequent calculus for a fragment of arithmetic. *Studia Logica* 39: 245–256.

[239] Sambin, G., and S. Valentini. 1982. The modal logic of provability. The sequential approach. *Journal of Philosophical Logic* 11: 311–342.

[240] Sato, M. 1977. A study of Kripke-type models for some modal logics by Gentzen's sequential method. *Publications of the Research Institute for Mathematical Sciences*, Kyoto University 13: 381–468.

[241] Satre T.W. 1972. Natural deduction rules for modal logics. *Notre Dame Journal of Formal Logic* 13: 461–475.

[242] Schmidt, R.A. 2006. Developing modal tableaux and resolution methods via first-order resolution. In *Advances in modal logic 6*, ed. M. deRijke, 107–135. Dordrecht: Kluwer.

[243] Schmidt, R.A., and U. Hustadt. First-order methods for modal logics. *Volume in memoriam of Harold Ganzinger, Lecture Notes in AI*, Appear in eds. A. Podelski, A. Voronkov, and R. Wilhelm. Springer.

[244] Schütte, K. 1977. *Proof theory.* Berlin: Springer.

[245] Scott, D. 1970. Advices on modal logic. In *Philosophical problems in logic*, ed. K. Lambert, 143–173. Dordrecht: Reidel.

[246] Segerberg, K. 1971. *An essay in classical modal logic I–III.* Uppsala: Filosofiska Studier nr 13, Uppsala Universitet.

[247] Seldin, J. 1989. Normalization and excluded middle I. *Studia Logica*, 48: 193–217.

[248] Seligman, J. 1991. A cut-free sequent calculus for elementary situated reasoning. *Technical Report HCRC/RP-22 HCRC*, Edinburgh.

[249] Seligman, J. 1992. Situated consequence in elementary situation theory. *Logic Group Preprint IULG-92–16*, Indiana University.

[250] Seligman, J. 1997. The logic of correct description. In *Advances in Intensional Logic*, ed. M. de Rijke, 107–135. Dordrecht: Kluwer.

[251] Seligman, J. 2001. Internalization: The case of hybrid logics. *Journal of Logic and Computation* 11: 671–689.

[252] Shimura, T. 1991. Cut-free systems for the modal logic S4.3 and S4.3GRZ. *Reports on Mathematical Logic* 25: 57–73.

[253] Shvarts, G.F. 1989. Gentzen style systems for K45 and K45D. *Logic at Botic '89*, 245–256. Berlin: Springer.

[254] Sieg, W., and J. Byrnes. 1998. Normal natural deduction proofs (in classical logic). *Studia Logica* 60: 67–106.

[255] Sieg W., and S. Cittadini. 2005. Normal natural deduction proofs (in nonclassical logics). In *Mechanizing mathematical reasoning, LNAI 2605*, eds. D. Hutter, and W. Stephan, 169–191. Springer-Verlag.

[256] Siemens, D.F. 1977. Fitch-style rules for many modal logics. *Notre Dame Journal of Formal Logic* 18: 631–636.

[257] Simpson, A. 1994. *The Proof Theory and Semantics of Intuitionistic Modal Logic*, PhD thesis, University of Edinburgh.

[258] Skura, T. 1992. Refutation rules for three modal logics. *Bulletin of the Section of Logic* 21(1): 31–32.

[259] Smullyan, R. 1965. Analytic natural deduction. *The Journal of Symbolic Logic* 30(2): 123–139.

[260] Smullyan, R. 1966. Trees and nest structures. *The Journal of Symbolic Logic* 31(3): 303–321.

[261] Smullyan, R. 1968. *First-order logic*. Berlin: Springer.

[262] Stachniak, Z., and P. O'Hearn. 1990. Resolution in the domain of strongly finite logics. *Fundamenta Informaticae* 13: 333–351.

[263] Stalmarck, G. 1991. Normalizations theorems for full first order classical natural deduction. *The Journal of Symbolic Logic* 56: 129–149.

[264] Stouppa, P. 2004. *The design of modal proof theories: The case of S5*, Master Thesis, Dresden.

[265] Stouppa, P. A deep inference system for the modal logic S5. Appear in *Studia Logica*.

[266] Suppes, P. 1957. *Introduction to logic*. Princeton: Van Nostrand.

[267] Suszko, R. 1948. W sprawie logiki bez aksjomatów. *Kwartalnik Filozoficzny* 17(3/4): 199–205.

[268] Szabo, M.E. 1969. *The collected papers of Gerhard Gentzen*. Amsterdam: North-Holland.

[269] Takano, M. 1992. Subformula property as a substitute for cut-elimination in modal propositional logics. *Mathematica Japonica* 37(6): 1129–1145.

[270] Takano, M. 2001. A modified subformula property for the modal logics K5 and K5D. *Bulletin of the Section of Logic* 30(2): 115–123.

[271] Tapscott, B.L. 1987. A simplified natural deduction approach to certain modal systems. *Notre Dame Journal of Formal Logic* 23(3): 371–384.

[272] Tarski, A. 1930. Fundamentale Begriffe der Methodologie der deduktiven Wissenschaften. *Monatschefte für Mathematik und Physik*, 37: 361–404.

[273] Ten Cate, B. 2004. *Model theory for extended modal languages*. PhD thesis, University of Amsterdam.

[274] Thomason, R. 1970. *Symbolic logic*. New York: Macmillan.

[275] Thomason, R.H. 1970. A Fitch-style formulation of conditional logic. *Logique et Analyse* 13: 397–412.

[276] Troelstra, A.S., and H. Schwichtenberg. 1996. *Basic proof theory*. Oxford: Oxford University Press.

[277] Tzakova, M. 1999. Tableau calculi for hybrid logics. In *Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs*, ed. N. Murray, 278–292. USA: LNAI 1617.

[278] Wallen, L.A. 1990. *Automated proof search in non-classical logics*. Cambridge: MIT Press.

[279] Wang, H. 1960. Toward mechanical mathematics. *IBM Journal of Research and Development* 4: 2–22.

[280] Wansing, H. 1998. *Displaying modal logics*. Dordrecht: Kluwer Academic Publishers.

[281] Wansing, H. 2002. Sequent systems for modal logics. In *Handbook of philosophical logic*, eds. D. Gabbay, and F. Guenthner, vol IV, 89–133. Dordrecht: Reidel Publishing Company.

[282] Wisdome, W.A. 1964. Possibility elimination in natural deduction. *Notre Dame Journal of Formal Logic* 5: 295–298.

[283] Woleński, J. 1985. *Filozoficzna szkoła Lwowsko-Warszawska*. Warszawa: PWN.

[284] Wolper, P. 1985. The tableau method for temporal logic: An overview. *Logique et Analyse* 28(110/111): 119–136.

[285] Wójcicki, R. *Theory of logical calculi.* Dordrecht: Kluwer.

[286] Vickers, S. 1988. *Topology via logic.* Cambridge: Cambridge University Press.

[287] Vigano, L. 2000. *Labelled non-classical logics.* Dordrecht: Kluwer.

[288] Zeman, J.J. 1973. *Modal logic.* Oxford: Oxford University Press.

Index

D

E