

«Y lo mismo que el concepto de número, el de figura está tomando exclusivamente del mundo exterior y no ha brotado en la cabeza por obra del pensamiento puro».

Federico Engels

CAPÍTULO 4

Teoría de las gramáticas formales y de los dispositivos automáticos

§ 4.1. Gramáticas formales

Examinemos un sistema de sustituciones prefijado por el alfabeto $M = \{m_i / i = 1, \dots, p\}$ y por las *sustituciones básicas*

$$\alpha_i \rightarrow \beta_i, \quad (4.1)$$

donde α_i, β_i son las fórmulas (palabras) tal vez vacías en el alfabeto M .

Comprenderemos toda sustitución $\alpha_i \rightarrow \beta_i$ como una regla de deducción. Con frecuencia el sistema de sustituciones se denomina *semisistemas* de Thue, en honor del matemático noruego Aksel Thue. Empleando estos semisistemas Chomsky formó y desarrolló el aparato de gramáticas formales.

Definamos el concepto de la *gramática formal* que a continuación se llamará simplemente gramática. Examinemos un alfabeto finito $M = \{m_1, m_2, \dots, m_n\}$, cuyos elementos se denominarán *símbolos* (letras) y sucesiones finitas de símbolos, *palabras*.

Designemos todo el conjunto de palabras, cuyas longitudes no tienen ningunas restricciones mediante \mathcal{L} . Digamos que $\mathcal{L} \subset \mathcal{L}_0$ es un lenguaje en el alfabeto M .

Sea G una colección de reglas, con ayuda de las cuales en M se engendran todas las palabras pertenecientes al lenguaje \mathcal{L} y sólo ellas. Llámase *gramática del lenguaje* \mathcal{L} la colección de reglas G .

Denominaremos *equivalentes* a dos lenguajes si coinciden los conjuntos de palabras que integran estos lenguajes. Dos gramáticas G_1 y G_2 sobre \mathcal{L} se denominan *equivalentes*, si se engendran por éstas los lenguajes equivalentes.

Acordémonos decir que G es la *gramática del número finito de estados*, si las reglas de engendrar palabras de alfabeto $M = \{m_1, m_2, \dots, m_n\}$ se prefijan del modo siguiente. Existe un conjunto finito de estados $\{S_0, S_1, \dots, S_r\}$ y a cada $S_j (j = 1, 2, \dots, r)$ le pone en correspondencia un juego de pares de tipo (m_i, S_q) , donde $i \in \{1, 2, \dots, n\}$, $q \in \{0, 1, \dots, r\}$. Al estado S_0 le pone en correspondencia los pares de tipo (m_0, S_h) , donde $h \in \{1, 2, \dots, r\}$. El símbolo m_0 es un signo especial del blanco entre las palabras. La construcción de las palabras se realiza del modo siguiente: del estado S_0 pasa a cualquier estado S_q , uno de aquellos S_q que son segundos

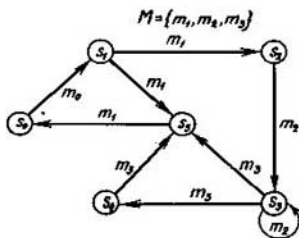


Fig. 4.1

miembros de los pares de tipo (m_0, S_q) , y se pone el signo del blanco en el inicio de la palabra. Partiendo de los pares, puestos en correspondencia al S_q elegido, se toma cualquier (m_i, S_i) . Esta elección determina el siguiente estado S_i y el primer símbolo de la palabra m_i . En adelante, el proceso de la construcción de la palabra se realiza de modo análogo. La palabra se termina pasando al estado final, que es, como regla, S_0 .

El lenguaje generado por una gramática con el número finito de estados se denomina *lenguaje con el número finito de estados*. Es cómodo representar la estructura de tales lenguajes en forma de un grafo, cuyos vértices se ponen en correspondencia a S_j y los arcos, en correspondencia a los pares (m_i, S_q) . En la fig. 4.1 se da un ejemplo de tal grafo. Utilizando la gramática, prefijada por este grafo, se engendra un lenguaje compuesto del siguiente conjunto de palabras: $\{m_1 m_1, m_1 m_2 m_3 m_1, m_1 m_3 m_3 m_1\}$. Se puede examinar la generación de las cadenas de los símbolos como resultado del trabajo de un dispositivo hipotético (fig. 4.2). A lo largo de una cinta infinita (a una o dos direcciones) dividida en células se mueve la cabeza de control (CC). Están prefijados un alfabeto exterior $M = \{m_0, m_1, m_2, \dots, m_n\}$, cuyos símbolos se denominan *letras*, un alfabeto interior $S = \{s_0, s_1, \dots, s_r\}$, cuyos símbolos se denominan *estados* y un alfabeto de traslaciones $E = \{D, I, N\}$. Todas las células se llenan con los símbolos de M , uno en cada célula. El símbolo m_0 juega el papel del símbolo vacío (si en una célula se halla m_0 , entonces «en esta célula no está escrito nada»). Se supone que toda la cinta infinita está llena siempre con los símbolos m_0 , excepto aquellas células, en las cuales están escritos cualesquier otros símbolos de M .

La cabeza de control puede mantenerse en varios estados que se caracterizan por los símbolos de S . El estado s_0 es específico. Si la CC se mantiene en el estado s_0 , «la máquina no hace ningún trabajo (está desconectada)». Se supone que al final del trabajo la máquina pasa siempre en el estado

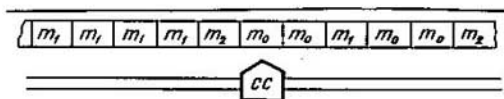


Fig. 4.2

s_0 . Durante el trabajo de la máquina la CC puede moverse a lo largo de la cinta en tiempos discretos. Se mueve ora una célula a la derecha (D), ora una célula a la izquierda (I). Puede ocurrir que en un tiempo dado del trabajo la CC no se mueve (N).

En todo tiempo del trabajo la CC realiza las siguientes operaciones:

1) lee el símbolo m_i que está en la célula «vista» por la CC en este tiempo; 2) en relación con el símbolo leído m_i y su estado s_j escribe el símbolo m_k en esta célula; 3) se mueve (o no) a lo largo de la cinta; 4) pasa al estado siguiente s_p .

Se puede representar todo el trabajo de la máquina utilizando una tabla funcional T , en cuyas células se encuentran las ternas de tipo $m_k s_p e_i$, donde $e_i \in E$ es un símbolo que determina la traslación. De este modo, la tabla funcional determina la aplicación $M \times S$ en $M \times S \times D$. El sentido enjundioso de la aplicación $(m_i, s_j) \rightarrow (m_k s_p e_i)$ consiste en que permaneciendo en el estado s_j y leyendo el símbolo m_i de la célula, la CC escribe el símbolo m_k en la célula dada de la cinta, pasa al estado s_p y realiza la traslación determinada por el símbolo e_i . Acordémonos que la tabla funcional está siempre hecha de modo que tiene lugar la aplicación $(m_i, s_0) \rightarrow (m_i, s_0, N)$. Esto significa que en el estado «desconectado» la máquina no funciona.

Hasta que la máquina empiece a funcionar es necesario llenar (si esto es indispensable) unas células de la cinta con los símbolos diferentes de m_0 , trasladar la CC a un estado distinto de s_0 y fijar la posición inicial de la CC respecto a la cinta. Después de esto la máquina funcionará en correspondencia con la tabla T . El funcionamiento de la máquina puede ser prefijado utilizando también un grafo, cuyos vértices se ponen en correspondencia biunívoca a los estados de este dispositivo, los arcos, en correspondencia con las traslaciones de un estado a otro. Con ello cada arco (s_j, s_p) está ponderado por el par $(m_i, m_k e_i)$. Según Chomsky, frecuentemente el estado se denomina *símbolo no terminal* (auxiliar) y el símbolo $m_i \in M$ se denomina *terminal*. El dispositivo hipotético descrito se denomina *máquina de Turing*.

Tesis de Post. *Un semisistema arbitrario de Thue puede representarse como una máquina de Turing y viceversa.*

En el capítulo 1 fue examinada la definición intuitiva del concepto de algoritmo. Empleando la máquina de Turing, precisemos este concepto.

Tesis de Turing. *Para cualquier algoritmo comprendido en sentido intuitivo, se puede construir una máquina de Turing, cuyo funcionamiento es equivalente a este algoritmo.*

El concepto de la máquina de Turing es una especificación estricta del concepto del algoritmo. El paso del concepto intuitivo del algoritmo al concepto exacto de la máquina de Turing posibilita llegar a la resolubilidad algorítmica (de máquina) de uno u otro problema.

En 1936, el científico norteamericano Church, examinando el problema de reconocer la deducibilidad en la lógica matemática, obtuvo uno de los primeros resultados negativos.

En el cálculo lógico para cualesquiera fórmulas dadas R y S determinemos, si existe o no una cadena deductiva que conduce de R a S .

A y B son las fórmulas *adyacentes*, si se puede transformar la fórmula A en la B , y viceversa, utilizando una sola vez la sustitución admisible. Una sucesión $(A_i, i = 1, 2, \dots, n)$ de fórmulas, entre las cuales las contiguas son adyacentes, se denomina *cadena deductiva* que conduce de A_1 a A_n . Como la resolución del problema de reconocer la deducibilidad se comprende un algoritmo que responde existe o no una cadena deductiva (para cualesquiera R y S).

Un problema es *irresoluble algorítmicamente*, si no existe ningún algoritmo (correspondiente máquina de Turing) para resolverlo. La máquina de Turing independiente puede ser representada como un programa de forma arbitraria para un ordenador con la memoria potencialmente infinita.

Teorema 4.1 (teorema de Church). *El problema de reconocimiento de la deducibilidad es algorítmicamente irresoluble.*

Siguiendo a Chomsky, introduzcamos las restricciones de las sustituciones $\alpha \rightarrow \beta$, observando al mismo tiempo, la correspondencia entre la gramática obtenida y el dispositivo automático.

Restricción 1. Si $\alpha \rightarrow \beta$ satisface la expresión (4.1), o sea, es una regla de deducción, entonces

$$(\exists a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n(m \leq n))((\alpha = a_1 a_2 \dots a_m) \& (\beta = b_1 b_2 \dots b_n)). \quad (4.2)$$

En lenguaje generado por la gramática que satisface (4.2) se realiza por la máquina de Turing.

Restricción 2. Si $\alpha \rightarrow \beta$ es una regla de deducción, entonces $(\exists \gamma_1, \gamma_2, a, \omega(\gamma_1, \gamma_2, \omega$ son las cadenas, a es un símbolo independiente; ω no es vacía).

$$((\alpha = \gamma_1 a \gamma_2) \& (\beta = \gamma_1 \omega \gamma_2)). \quad (4.3)$$

Las gramáticas que satisfacen la relación (4.3) se denominan *de contexto (vinculadas por el contexto)*.

Las gramáticas de contexto se realizan mediante los dispositivos tipo de autómeta de Myhill.

Sea (i, j, k, l, p) una de las reglas que determina el funcionamiento del dispositivo automático: si el bloque de mando permanece en el estado S_j y la cabeza exploradora está frente a la célula que contiene el símbolo m_i , entonces el bloque de mando puede pasar al estado S_k , mientras que la cinta avanza hasta l células a la izquierda y m_p sustituye el símbolo a examinar. El dispositivo que funciona según este principio se llama *autómeta de Myhill*.

Restricción 3. Si $\alpha \rightarrow \beta$ es una regla de deducción, entonces α es la letra no terminal y β no es vacía:

$$\beta \neq \emptyset \quad (4.4)$$

La gramática que satisface (4.4) se llama *sin contexto (libre del contexto)*.

De acuerdo a (4.4), cada regla de gramática afirma que cierto símbolo no terminal puede sustituirse por una cadena de símbolos independientemente del contexto.

Un lenguaje generado por una gramática sin contexto se realiza por un dispositivo automático de Myhill de tipo especial que utiliza la memoria jerárquica. Según Newell, Shaw y Simon, este dispositivo automático se denominará *autómata jerárquico*.

El autómata jerárquico es una composición de *un dispositivo automático de mando* y de tres *cajas*, cada una de las cuales representa de por sí una cinta infinita dirigida a un lado. En la cinta está escrita una palabra, cuya primera letra se encuentra en la primera célula, la segunda está en la segunda célula, etc. Leyendo la palabra el dispositivo automático percibe la primera letra, luego la borra y el resto de la palabra se desplaza hacia la primera célula. Cuando en la caja se graba una palabra de la longitud k , las primeras k células se libran como resultado de haber trasladado en k células la palabra anteriormente inscrita. La caja de entrada está vinculada con los canales de entrada del dispositivo automático de mando, la de salida, con los canales de salida, la interior se vincula tanto con los canales de entrada, como con los de salida del dispositivo automático de mando. El conjunto de estados interiores del dispositivo automático de mando se parte en dos subconjuntos A y B . Si el estado del dispositivo automático de mando pertenece al subconjunto A , se lee la información de las cajas de entrada e interior. Si su estado pertenece a B , $S_i \in B$, la lectura se hace sólo de la caja interior. Al mismo tiempo, el dispositivo automático pasa al estado siguiente y escribe palabras en las cajas interior y de entrada.

Una regla de la gramática sin contexto se denomina *lineal*, si tiene forma

$$A \rightarrow xBy; \quad (4.5)$$

lineal derecha, si

$$A \rightarrow xB, \quad (4.6)$$

y *lineal izquierda*, si

$$A \rightarrow Bx. \quad (4.7)$$

La regla de forma $A \rightarrow x$ llámase *conclusiva*. Según las restricciones determinadas por las reglas (4.5)—(4.7), la gramática sin contexto puede ser:

a) *lineal*, si cada regla suya no conclusiva es lineal, en particular, si es lineal izquierda o lineal derecha;

b) *unilateralmente lineal*, si cada regla suya no conclusiva es lineal izquierda o lineal derecha;

c) *metalineal*, si todas sus reglas no conclusivas ora son lineales, ora tienen forma $S \rightarrow \beta$ y, además, la gramática no posee reglas tipo $A \rightarrow \alpha S \beta$ para ningunos A, α, β , donde α, β no son vacías.

La gramática unilateralmente lineal se realiza por el *dispositivo automático finito* y engendra un lenguaje llamado *automático finito*.

Consideremos una gramática unilateralmente lineal, cuyas reglas son todas lineales derechas (para más precisión) o todas conclusivas. Sin perder la generalidad, se puede suponer que toda regla lineal tiene forma $A \rightarrow aB$ (donde B es un símbolo no inicial) y cada regla conclusiva de la gramática tiene forma $A \rightarrow a$.

Sean A_1, A_2, \dots, A_n los símbolos no terminales de la gramática, con la particularidad de que A_1 es el *símbolo inicial*. A la gramática le ponemos en correspondencia un dispositivo automático finito, cada uno de los estados interiores del cual corresponde biunívocamente a un símbolo no terminal de la gramática, y el símbolo de entrada corresponde a un símbolo terminal de la misma. Además, si $A_i \rightarrow aA_j$ es una regla de la gramática, la terna (a, A_i, A_j) determina el funcionamiento del dispositivo automático y se comprende como el paso del estado A_i al estado A_j , cuando se lee el símbolo de entrada a .

Para más generalidad, consideraremos que, pasando del estado S_i al estado S_j como resultado de la influencia de entrada a , el dispositivo automático elabora un símbolo b a su entrada. Entonces el dispositivo automático puede ser como una cuaterna (a, b, S_i, S_j) .

Si se fija el *estado inicial* S_0 , el dispositivo automático realiza un operador T

$$b = T(a, S_i, S_j).$$

A continuación lo denominaremos *de autómeta*.

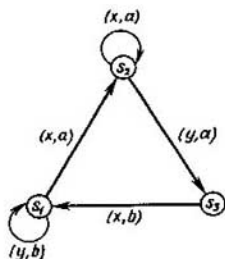


Fig. 4.3

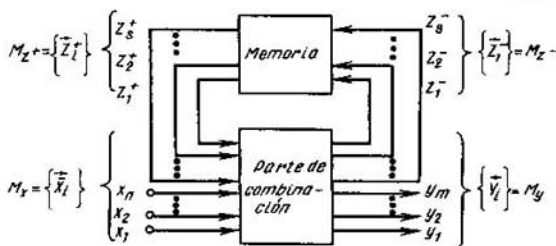


Fig. 4.4

El operador de autómatas T traspa una sucesión de símbolos de entrada (a_i) en una sucesión de salida (b_i) según el estado inicial y la gramática unilateralmente lineal realizable. Es cómodo representar el dispositivo automático en forma de la función T sobre un grafo $G = \langle V, U \rangle$, a cada vértice del cual le corresponde biunívocamente un estado del dispositivo automático. Si éste pasa del estado S_i al estado S_j bajo la influencia de entrada a , elaborando, al mismo tiempo, el símbolo de salida b , entonces los vértices correspondientes v_i y v_j se unen mediante el arco (v_i, v_j) ponderado por el par (a, b) . De tal modo, el campo de definición de esta función T es el grafo $G = \langle V, U \rangle$ construido por medio del procedimiento anteriormente examinado. El campo de valores son los símbolos de entrada y salida y los identificadores de los estados del dispositivo automático.

Por ejemplo, examinemos una gramática unilateralmente lineal con el alfabeto de símbolos terminales $M_T = \{x, y, a, b\}$, con el alfabeto de los símbolos no terminales $M_N = \{S_1, S_2, S_3\}$ y las siguientes reglas de deducción: $S_1 \rightarrow ybS_1$, $S_1 \rightarrow xaS_2$, $S_2 \rightarrow xaS_2$, $S_2 \rightarrow yaS_3$, $S_3 \rightarrow xbS_1$. Un dispositivo automático finito realiza esta gramática. Si ponemos este autómata en el estado inicial S_1 y a la entrada transmitimos una sucesión de símbolos terminales (y, x, y, x) , obtenemos a la salida la sucesión de los símbolos terminales (b, a, a, b) , cuando los símbolos no terminales forman la sucesión (S_1, S_1, S_2, S_3) . El operador de autómatas realizables T puede representarse en forma de la función correspondiente sobre el grafo $G = \langle V, U \rangle$ (fig. 4.3).

Si no examinemos el dispositivo automático como un dispositivo que realiza una gramática correspondiente, sino estudiemos su estructura, entonces hay que representar éste no en forma de la máquina de Turing, sino en forma del esquema en bloque representado en la fig. 4.4, donde M_x es el conjunto de símbolos terminales de entrada; M_y es el conjunto de símbolos terminales de salida; M_z es el conjunto de símbolos no terminales ($M_z = M_z^+, M_z^-$).

§ 4.2. Etapas principales del diseño de los dispositivos automáticos

Analicemos el problema de diseño del dispositivo automático. En la fig. 4.4 tenemos M_x , significa el *conjunto de vectores de entrada*

$$X = x_1^{\sigma_1} x_2^{\sigma_2} \dots x_n^{\sigma_n};$$

M_y es el *conjunto de vectores de salida*

$$Y = y_1^{\sigma_1} y_2^{\sigma_2} \dots y_m^{\sigma_m};$$

M_z^- es el conjunto de vectores que caracterizan los *canales de entrada de realimentación* (canales de memoria)

$$Z^- = (z_1^-)^{\sigma_1} (z_2^-)^{\sigma_2} \dots (z_r^-)^{\sigma_r},$$

M_z^+ es el conjunto de vectores que caracterizan los *canales de salida de realimentación*

$$Z^+ = (z_1^+)^{\sigma_1} (z_2^+)^{\sigma_2} \dots (z_s^+)^{\sigma_s}.$$

Si tenemos la lógica de k signos, cada uno de los canales puede estar en uno de sus k valores $\sigma \in (0, 1, \dots, k-1)$.

Acordémonos designar la variable α , igual a $\sigma \in (0, 1, 2, \dots, k-1)$ como α^σ . Entonces, para la gramática de autómatas, es más cómodo definir las reglas de deducción como sustitución $XZ^+ \rightarrow Z^- Y$, en la cual el valor inicial del vector Z^+ y sus valores sucesivos se obtienen al igualarlos al valor del vector Z^- calculado en el paso anterior, o sea,

$$Z^+(t=0) = Z_0^+, Z^+(t+\tau) = Z^-(t),$$

donde τ es una constante de tiempo.

Los vectores XZ^+ y $Z^- Y$ se obtienen añadiendo los vectores Z^+ e Y a la derecha de X y Z^- , respectivamente.

A continuación, los estados de los canales de realimentación se denominarán *estados interiores del dispositivo automático*, mientras que la constante de tiempo τ se llamará *tiempo de transición* de un estado interior en otro con la particularidad de que, según la destinación del dispositivo automático y su realización, τ puede ser constante para el dispositivo automático dado o depender de la variación del vector X . En el primer caso el dispositivo automático se denomina *sincrónico*, en el segundo, *asincrónico*.

Para un $(Z^+)_0$ prefijado, la sucesión de los vectores de entrada X (*sucesión de entrada*) determina unívocamente la sucesión de los vectores de salida Y (*sucesión de salida*). Lo ilustremos analizando el siguiente ejemplo.

Sea que tenemos un dispositivo con el canal de entrada x , un canal de realimentación z y un canal de salida y que realiza la aplicación $XZ^+ \rightarrow Z^- Y$ prefijado en forma de una tabla (tabla 4.1),

Tabla 4.1

x	z^+	z^-	y
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	0

Tabla 4.2

Tiempo	x	z^+	z^-	y
0	1	0	0	1
τ	0	0	1	1
2τ	1	1	0	0
3τ	0	0	1	1
4τ	0	1	1	0
5τ	1	1	0	0

Determinemos la sucesión de salida, si la de entrada tiene forma 101001 y el valor inicial del vector Z^+ es igual a 0. En el instante inicial de tiempo, el vector XZ^+ , igual a 10, determina $Z^- Y = 01$ (la tercera fila de la tabla). Dentro de un intervalo de tiempo τ , $XZ^+ = 00$ determinará $Z^- Y = 11$ (la primera fila), etc. Anotemos la definición de la sucesión de salida en forma de la tabla 4.2.

Por consiguiente, $101001 \xrightarrow{Z_0^+ = 0} 110100$.

Según la aplicación de autómatas prefijada para $Z_0^+ = 1$ la sucesión de entrada 101001 se transforma en la sucesión de salida 010100.

De tal modo, la aplicación de autómatas $XZ^+ \rightarrow Z^- Y$ determina unívocamente la sucesión de salida (Y_i) por la sucesión de entrada dada (X_i) y el estado interior inicial Z_0^+ : $(X_i) \xrightarrow{Z_0^+} (Y_i)$.

Una de las características principales del dispositivo automático es la capacidad de su memoria. El número de estados interiores del dispositivo automático se denomina *capacidad de memoria del dispositivo automático*.

Como se sabe, la transformación de la información es resultado de haber realizado un algoritmo, con ello el *autómata operacional* realiza los pasos del algoritmo y el dispositivo *automático de mando*, el orden de cumplir los pasos. Los dispositivos automáticos operacional y de mando difieren por su destino y también por la capacidad de memoria. En el dispositivo automático operacional se transforma la información prefijada en forma de cierto conjunto de números inscritos en registros. La capacidad de memoria del dispositivo automático operacional es prácticamente infinito. Por ejemplo, el bloque de registros de un ordenador moderno, que integra el dispositivo automático operacional y consta de 22 registros binarios de 16 órdenes, tiene capacidad de memoria igual a $2^{352} > 10^{100}$ bitios. La capacidad de memoria de los dispositivos automáticos de mando suele tener de unas decenas a unas decenas de miles de bitios, o sea, no es grande en comparación con la capacidad de memoria de los dispositivos automáticos operacionales.

Examinemos la interacción de los dispositivos automáticos operacional y de mando (fig. 4.5). A la entrada del dispositivo automático operacional

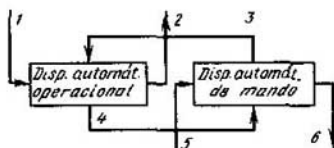


Fig. 4.5

(canal 1) llega una información que se transforma; de la salida del dispositivo automático operacional (canal 2) se toman los resultados de las transformaciones; por el canal 3 llegan las acciones de mando correspondientes al algoritmo que se realiza; por el canal 4 llegan a la entrada del dispositivo automático de mando los indicios que caracterizan la información que está por transformar; por el canal 5 llega la señal que determina la transformación a cumplir y su comienzo; por el canal 6 sale la señal fin de la operación. Los canales 1 y 2 se denominan *informativo*, los de 3 a 6 se denominan *de control*. Por ejemplo, examinemos como se cumple la operación de adición en el dispositivo aritmético (DA) de un ordenador. En este caso el dispositivo automático operacional es el dispositivo aritmético, o sea, los registros, el sumador y las comunicaciones entre ellos, el dispositivo automático de mando es el de mando del DA. Cuando se efectúa el algoritmo de adición, el primero y el segundo sumando sacados del dispositivo de memoria (DM) del ordenador se inscriben por el canal 1 en los correspondientes registros del DA; por el canal 5 del dispositivo central de mando (DCM) del ordenador llega el código de la operación («adición»), por el canal 3 del dispositivo de mando del DA, conforme al algoritmo de adición, se envían las señales de mando: el desplazamiento de registros, la excitación de las barras correspondientes en el sumador, la inscripción de la suma en el registro del resultado y otras señales. Por el canal 4 llegan los indicios (por ejemplo, el contenido de los órdenes de signo en el sumador que se utiliza para detectar la infracción de la normalización) que determinan la marcha del mando sucesivo. Por el canal 6 se transmite al DCM la señal de que la operación se ha realizado.

El ordenador es un transformador complejo de la información. Según V.M.Glushkov, es conveniente considerarlo como una *composición de pares de los dispositivos automáticos*, cada uno de los cuales comprende los dispositivos automáticos operacional y de mando. Además, cada dispositivo del ordenador (de entrada, de salida, el DM, el DCM) se representa, de modo análogo al dispositivo aritmético, como un par o varios pares de tales dispositivos automáticos. Tal representación del ordenador es cómoda para el análisis y la síntesis del ordenador y, también, es el desarrollo lógico de la estructura de los ordenadores modernos. Actualmente, en calidad de un método de aumentar la productividad del ordenador, se utiliza el régimen de programas múltiples, lo que, en particular, disminuye retardos pro-

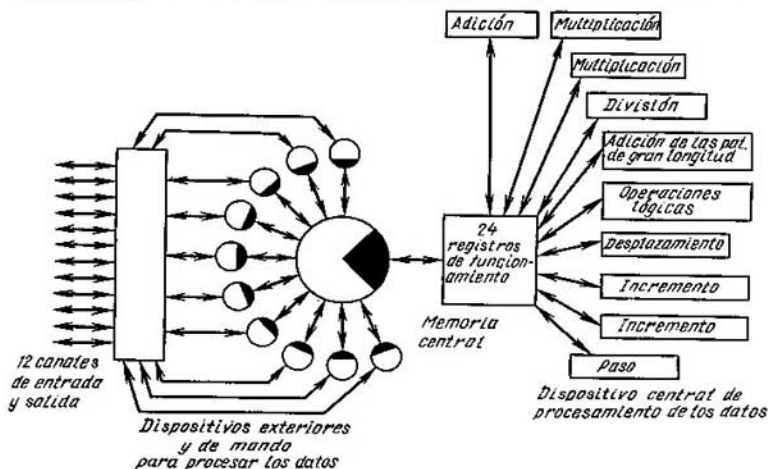


Fig. 4.6

vocados por la baja velocidad de los dispositivos exteriores y la falta de correspondencia entre las velocidades de funcionamiento de los dispositivos aritmético y exteriores. Para realizar el régimen de programas múltiples del ordenador, es indispensable cierta «autonomía» de unidades separadas, es decir, la posibilidad de almacenar y transformar la información y controlar esta transformación dentro de la misma unidad. En otras palabras, es necesario que toda unidad sea una composición de los dispositivos automáticos operacional y de mando. Para ilustrar todo lo dicho, demos el esquema en bloque del ordenador CDC-6600 (fig. 4.6), en el cual la información de entrada llega por 12 canales de entrada y salida y se da primeramente a los dispositivos periféricos de procesamiento de la información. Cada uno de estos dispositivos es una composición de los dispositivos automáticos operacional y de mando. Estos dispositivos de volumen pequeño pueden ora entregar los resultados transformados en el dispositivo de salida, ora transmitirlos, a la memoria central que es toda una jerarquía de dispositivos de memoria. La última operación se cumple con ayuda del sistema centralizado de mando que distribuye el «trabajo» entre los dispositivos especializados de procesamiento de la información que tienen alta velocidad de funcionamiento. Cada uno de estos dispositivos es una composición de los dispositivos automáticos operacional y de mando. Cada par de tales dispositivos realiza una o varias operaciones, por ejemplo, la operación de multiplicación.

Cuando el ordenador funciona en el régimen de programas múltiples, se realiza el principio del funcionamiento descentralizado de dispositivos separados. En el límite, el principio de descentralización pasa al principio de la descentralización absoluta o al principio de Holland que propuso la red completamente distribuida de los dispositivos de procesamiento de la información, cada uno de los cuales es una composición de los dispositivos automáticos operacional y de mando. Todos estos bloques funcionan independientemente. Esta estructura del ordenador realiza la concepción de los «programas de trabajo nadando en el mar de equipos».

Una de las etapas de desarrollo de esta tendencia es la concepción del *procesamiento distribuido de la información* que en el presente se realiza en forma de redes de ordenadores, centros de cálculos de uso colectivo, etc.

En el diseño de los ordenadores se puede destacar tres etapas principales: *sistémica, lógica y técnica*.

En la etapa de diseño sistémica se construye la composición de pares de los dispositivos automáticos operacionales y de mando (el esquema general en bloque del ordenador), se determinan la capacidad necesaria de memoria de los dispositivos automáticos, su interacción basada en la selección de sistema de instrucciones, lenguajes interiores y exteriores del ordenador, etc. En esta etapa, la información de partida es un conjunto de clases de problemas los que deben ser resueltos por el ordenador a diseñar y de sus parámetros (velocidad de funcionamiento, costo, dimensiones extremas, etc.).

Los dispositivos automáticos examinados en la etapa de diseño sistémica tienen gran capacidad de memoria que supera, como ya lo indicamos, 10^{100} bits. Por esta razón en el presente los problemas de la etapa sistémica se resuelven empleando la simulación en programas.

En la etapa sistémica, el proceso de resolver el problema por el método de simulación en programas suele tener los siguientes pasos:

- a) la composición de un modelo matemático que refleja las propiedades principales del dispositivo en simulación o del ordenador en total;
- b) la elaboración de un algoritmo simulador, su registración en un lenguaje destinado para describir los modelos de los ordenadores;
- c) la realización del algoritmo simulador en un ordenador;
- d) el análisis de los resultados y la corrección del modelo del dispositivo o del ordenador.

La simulación en programas permite determinar las características principales del ordenador a diseñar y los puntos «flacos» de su estructura.

En la etapa de diseño lógico de los ordenadores se sintetizan directamente los esquemas lógicos (funcionales) de todos los bloques del ordenador. Para esta etapa, como la información de partida se presentan los algoritmos de funcionamiento de los bloques.

En la etapa de *diseño técnico* se construyen, basándose en los esquemas lógicos, los esquemas principales de montaje y se prepara la documentación técnica para fabricar los ordenadores.

El *diseño lógico* consiste en la síntesis de los dispositivos automáticos operacionales y también de los de mando. Debido a que la capacidad de memoria de los dispositivos automáticos operacionales es prácticamente infinita, en el presente estos últimos se sintetizan con ayuda de la simulación en programas. Para realizar la síntesis formalizada de los dispositivos automáticos operacionales es necesario desarrollar la teoría de los dispositivos automáticos infinitos, mientras que para formalizar la síntesis de autómatas de mando es posible aplicar la teoría de los dispositivos automáticos finitos.

Según esta teoría, en la proyección de los dispositivos automáticos de mando distinguiremos dos etapas principales: la *construcción del operador de autómatas* y la *síntesis estructural del dispositivo automático*.

I. Etapa de la construcción del operador de autómatas. En esta etapa se construye el *sistema de funciones de salida* $Y = f(X, Z^+)$ y el *sistema de funciones de excitación* $Z^- = \varphi(X, Z^+)$.

Una *aplicación de autómatas* escrita en forma de un sistema de funciones de salida y funciones de excitación se denominará *operador de autómatas*.

A su vez, esta etapa se compone de tres subetapas: *algorítmica*, *abstracta* y *de codificación (distribución) de estados interiores de autómatas*.

En la etapa de la proyección algorítmica un operador dado A se formaliza como un algoritmo, partiendo de las exigencias planteadas (la simplicidad de cumplimiento de las operaciones, la velocidad de funcionamiento, la reducción máxima de gastos para los aparatos, etc.). La búsqueda del algoritmo óptimo por el operador dado A se puede representar en forma de un árbol, cada vértice pendiente del cual corresponde a un algoritmo determinado, es decir, a una composición determinada de los dispositivos automáticos operacional y de mando. Para hallar el algoritmo óptimo, tenemos que realizar el sondeo de todos los vértices pendientes, cuyo número no es sabido y se determina por el grafo de desarrollo de las transformaciones que se examinan.

Por ejemplo, sea que es necesario sintetizar un dispositivo aritmético que realiza la operación cuadrádica de sumación. En función del algoritmo elegido para cumplir este trabajo, obtenemos un esquema en bloque determinado del dispositivo a sintetizar que se caracteriza por la velocidad de funcionamiento y gastos para los aparatos. En este caso se puede proponer, por ejemplo, tres variantes del algoritmo.

Variante 1. Sumamos dos primeros números, adicionamos el tercer número a la suma obtenida y otra vez adicionamos el último número a la suma obtenida. Esta variante del algoritmo se caracteriza por el tiempo

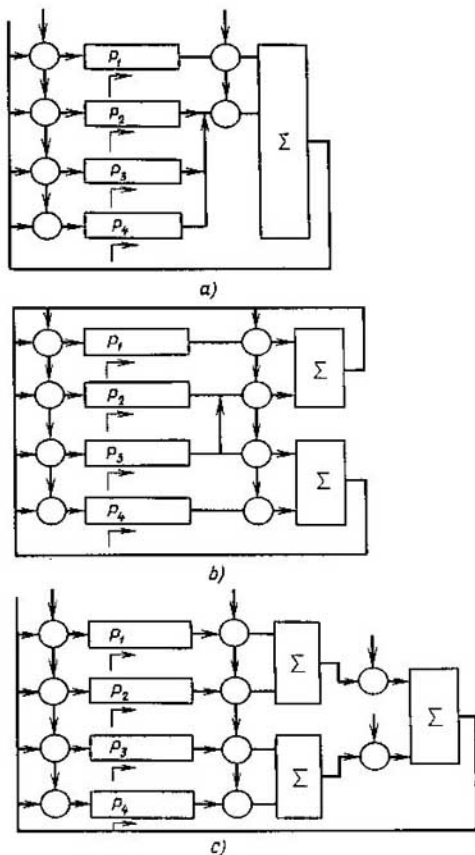


Fig. 4.7

de cumplimiento de la operación τ_1 y los gastos para los aparatos en forma de un sumador, cuatro registros, bloque de mando y canales de comunicación necesarios. A este algoritmo le corresponde el esquema en bloque representado en la fig. 4.7, a.

Variante 2. Sumamos simultáneamente el primer número y el segundo, el tercero y el cuarto; inscribimos los resultados en el primero y el tercer registro, respectivamente, y luego los sumamos en el primer sumador. El resultado final lo inscribimos en el primer registro. Esta variante de cumpli-

miento del trabajo A se caracteriza por el tiempo de cumplimiento τ_2 y los gastos para los aparatos representados en el esquema en bloque de esta variante (fig. 4.7, *b*).

Variante 3. Sumamos dos primeros números en un sumador, dos segundos en el segundo sumador, las sumas obtenidas, en el tercer sumador. Inscribimos el resultado en el primer registro. Esta variante se caracteriza por el tiempo de cumplimiento de la operación τ_3 y los gastos para los aparatos dados en la fig. 4.7, *c*.

Una u otra variante del algoritmo que realiza el trabajo A se escoge partiendo de las restricciones concretas para los gastos de aparatos y el tiempo de cumplimiento de las operaciones dadas.

Cada vértice pendiente del árbol de búsqueda se estima por el tiempo de cumplimiento de una transformación dada y la complejidad de los aparatos (la complejidad de los dispositivos automáticos operacional y de mando). La complejidad del dispositivo automático operacional se aprecia por el cálculo inmediato de los gastos de aparatos. Utilizando el concepto de derivada del modelo, se puede apreciar la complejidad del dispositivo automático de mando.

En la etapa abstracta se resuelve el problema de minimizar la capacidad de memoria del dispositivo automático. En la etapa de codificación (distribución) de los estados interiores del autómata a cada uno de ellos se les pone en correspondencia un código, es decir, un conjunto determinado de estados (valores) de elementos de la memoria.

Al cumplir las etapas consideradas componemos los sistemas de funciones de salida y de funciones de excitación del dispositivo automático, o sea, construimos el operador de autómata y luego pasamos a la síntesis estructural.

II. Etapa de la síntesis estructural del autómata. Esta etapa consiste en que de los elementos dados se construye el esquema lógico (funcional) del dispositivo automático que realiza el operador de autómata obtenido. A continuación consideremos la síntesis estructural con más detalles.

§ 4.3. Fundamentos aritméticos de los dispositivos automáticos operacionales

El proceso de cálculo, que realiza un programa en el ordenador representa en sí las transformaciones de números en paralelo y en serie. Cada número es una sucesión de cifras.

La palabra «cifra» proviene de la palabra árabe «sefr» traducida al latín. Ésta, a su vez, es traducción de la palabra sánscrita «suña» (alfabeto «devanagari») que significa «lugar vacío» («orden»), en el cual se pone un signo numérico al dar relaciones cuantitativas.

Llámase *sistema de numeración* o *numeración* una colección de procedi-

mientos y reglas para denotar y denominar los números. El sistema de numeración de las reglas de inscripción codificada de equivalentes cuantitativos permitiendo obtener unívocamente para toda cantidad su inscripción en código y por toda inscripción en código, su respectivo equivalente cuantitativo. Un conjunto de signos elementales que se utilizan para codificar se denominan *cifras* del sistema de numeración.

Sistemas de numeración son *posicionales* y *no posicionales*. En los sistemas no posicionales a cada cifra se le ha puesto unívocamente en correspondencia un equivalente cuantitativo estándar, mientras que el equivalente cuantitativo del código del número se calcula como una función de los equivalentes cuantitativos de las cifras que integran la inscripción de este código. Un ejemplo de tal sistema es el sistema de numeración, en la que se utiliza sólo una cifra, por ejemplo, 1. A esta cifra se le ha puesto en correspondencia el equivalente cuantitativo igual a la unidad. Entonces el código III III significa el equivalente cuantitativo seis; aquí la función para calcular el equivalente cuantitativo del código es la función de adición. En los sistemas posicionales un equivalente cuantitativo se pone en correspondencia no unívoca a cada cifra dependiendo de su posición en el código del número. Examinaremos solamente inscripciones linealmente ordenadas. En la inscripción escojamos el comienzo de la lectura (orden nulo). Los órdenes a la izquierda de éste enumeremos con 1, 2, ... y a la derecha, con $-1, -2, \dots$. A cada cifra a_i situada en el orden de número j se le pone en correspondencia un equivalente cuantitativo $\varphi(a_i, j)$. La función φ ora es igual para todos los órdenes, ora su forma cambia de orden a orden. A continuación se examinarán solamente los sistemas posicionales de una función φ igual para todos los órdenes. Cualquier sistema posicional se prefija por tres componentes: $\langle A, \varphi, F \rangle$. Aquí A es el *conjunto de las cifras del sistema*; φ es la función que para las cifras en cada orden determina su *equivalente cuantitativo*; F es la función que por los equivalentes cuantitativos de la inscripción del número determina el *equivalente cuantitativo del propio número*.

Según la forma de la función F elijamos dos tipos de sistemas de numeración: *aditivos* y *multiplicativos*. En los sistemas del primer tipo F es la función de *adición*, en los del segundo F es la función de *multiplicación*. No examinaremos otros tipos de F .

Si para cualesquiera cifras A y cualquier j tiene lugar la igualdad $\varphi(a_i, j) = S^j \cdot \varphi(a_i, 0)$, se trata del sistema de numeración *de la base S* . Si $\varphi(a_i, j) = p_j \varphi(a_i, 0)$ y p_j no coinciden para distintos j , el sistema es *de tipo de valores ponderables* y p_j son *pesos* de los órdenes.

Si en un sistema de la base S el conjunto de cifras comprende $0, 1, \dots, S-1$, el sistema tiene el *conjunto lógico de cifras* (*sistema lógico de numeración*); si $S = m + k + 1$ y el conjunto de cifras es $\{-m, -m+1, \dots, -1, 0, 1, \dots, k\}$, entonces para $m = k$ el sistema tiene el *conjunto simétri-*

co de cifras (sistema simétrico de numeración), cuando $k > m$ o $m > k$ el sistema tiene el conjunto de cifras asimétrico en sentido positivo o negativo, respectivamente, (sistema asimétrico de numeración).

Con la mayor frecuencia se usan sistemas lógicos de numeración de la base natural. En un sistema de la base S cualquier número x puede representarse en la forma siguiente:

$$x = \sum_{i=-\infty}^{\infty} (a_i)S^i, \quad (4.8)$$

donde (a_i) es el equivalente cuantitativo de la cifra a_i en el orden nulo.

En un sistema lógico de numeración de la base natural S , el desplazamiento de un número entero x en un orden a la derecha significa la división en números enteros de x por S . El resto obtenido es una cifra situada antes del desplazamiento en el orden nulo. El algoritmo de traspasar números enteros de un sistema de numeración de la base natural R a otro de la base natural Q consiste en lo siguiente. Un número x escrito en el sistema de numeración de la base R se divide por Q según las reglas de división en este sistema hasta obtener el resto. Si el cociente de la división no es igual a 0, pasa a ser divisible y el proceso de la división por Q continúa hasta que el cociente obtenido sea igual a 0. Los restos escribimos en el orden inverso de obtenerlos («del último al primero») y así tenemos la denotación del número x en el sistema de numeración de la base Q .

En el sistema lógico de numeración de la base natural S , el desplazamiento del número fraccionario x en un orden a la derecha significa la multiplicación de x por S . La parte entera obtenida, tal vez igual a 0, es una cifra situada en el orden -1 antes del desplazamiento. De aquí, el algoritmo de traspasar números fraccionarios (fracciones R -arias) de un sistema de numeración de la base R a otro de la base Q consiste en lo siguiente. La denotación fraccionaria de un número x se multiplica por Q según las reglas de multiplicación en el sistema de numeración de la base R . En el producto obtenido se separa la parte entera (tal vez, nula). La parte fraccionaria del producto se multiplica otra vez por Q , luego se separa la parte entera del producto, etc. multiplicando k veces con el traspaso de exactitud hasta Q^{-k} . Para obtener la denotación del número x en el sistema de numeración de la base Q (con exactitud de Q^{-k}), después de la coma se escriben todas las partes enteras de los productos según el orden de su obtención.

Examinemos un ejemplo de emplear estos algoritmos para traspasar el número decimal 11,87 al sistema quinario. Dividimos el número 11 por 5 en números enteros: $11 = 2 \cdot 5 + 1$, $2 = 0 \cdot 5 + 2$. En el sistema quinario el número 11 se denota como 21. Se puede traspasar la parte fraccionaria de 0,87 con exactitud hasta de 5^{-2} : $0,87 \cdot 5 = 4,35$; $0,35 \cdot 5 = 1,75$. Por consiguiente, en el sistema quinario el número 0,87 se denota como 0,41. En definitiva, tenemos $\langle 11,87 \rangle_{10} = \langle 21,41 \rangle_5$.

En los sistemas lógicos de numeración que adquirieron la máxima propagación, se puede escribir los números (si la base es natural) solamente

de un signo. En estos sistemas para representar números del otro signo se utiliza el signo especial «+» o «-». Es más conveniente codificar el signo del número con ayuda de cifras utilizadas para escribir el número. Para esto, en la denotación del número se escoge un orden especial llamado *de signo*. La cifra situada en él no codifica el equivalente cuantitativo y no participa en cálculos del equivalente por medio de la función F .

Codificamos el signo «+» con el cero y el signo «-» con la cifra $(S - 1)$ en el orden nulo cuando representamos números x , $|x| < 1$. Para esto determinemos el código directo de un número x , $|x| < 1$ del modo siguiente:

$$[x]_D = \begin{cases} 0. x_1 x_2 \dots x_n, & \text{cuando } x \geq 0 \\ S - 1. x_1 x_2 \dots x_n, & \text{cuando } x \leq 0. \end{cases}$$

Teniendo en cuenta que $|x| < 1$, esta correlación puede escribirse en la forma

$$[x]_D = \begin{cases} x, & \text{cuando } x \geq 0, \\ S - 1 + |x|, & \text{cuando } x \leq 0. \end{cases}$$

En el presente se usa ampliamente la forma *semilogarítmica* de representación de los números o la representación *de punto flotante*.

En un sistema de numeración de la base natural S , cualquier número x puede expresarse (multiformemente) como $S^{p(x)}m(x)$, donde $|m(x)| < 1$. Denominaremos $m(x)$ y $p(x)$ respectivamente *mantisa* y *orden* del número x en el sistema de la base S . Si para un S fijado damos $p(x)$ y $m(x)$, el número x se determina unívocamente. El par $p(x)$ y $m(x)$ está almacenado en la memoria del ordenador. Para representar unívocamente los números en forma semilogarítmica habitualmente es necesario que la mantisa satisfaga la desigualdad $S^{-1} \leq m(x) < 1$. La mantisa de este tipo se denomina *normalizada*.

Representando el número para la máquina en el orden más izquierdo se codifica el signo del orden, después en n_1 órdenes se anota el valor del orden, después van el orden para el signo de la mantisa y n_2 órdenes para escribir el valor de la mantisa.

Examinemos la realización de las operaciones de adición, sustracción, multiplicación y división en las aritméticas posicionales con el conjunto lógico de cifras. Para cualquier base S , las operaciones de adición y multiplicación se determinan por las tablas de cumplimiento de estas operaciones en un orden y por las reglas de formar traslados a los órdenes superiores. Además, es necesario que en la operación participen los órdenes correspondientes de los sumandos. Por eso, cuando se realiza la adición es también necesario justificar los órdenes de los sumandos (desplazar las mantisas de modo que se sumen los órdenes de números iguales).

Para realizar la adición algebraica, los órdenes se justifican cuando se aumenta un orden inferior hasta un superior. Para que no cambie en este

caso el equivalente decimal del número, es necesario que cada aumento del orden en una unidad se compense por el desplazamiento de la mantisa a un orden a la derecha.

El resultado obtenido toma el orden justificado. Normalizando la mantisa del resultado es necesario desplazarla a la derecha (en un orden) o a la izquierda hasta que se cumplan las condiciones de normalización. Para que al mismo tiempo se conserve el equivalente cuantitativo del número, es necesario aumentar (al desplazar la mantisa a la derecha) o disminuir (al desplazar la mantisa a la izquierda) el orden del resultado en el número de unidades que coincide con el número de desplazamiento.

Durante la sustracción de dos números surgen las complicaciones vinculadas con que se toman prestadas unidades de órdenes superiores. Esta operación se realiza mal en los ordenadores modernos para los sistemas de numeración con un conjunto lógico de cifras y una base natural. Para los sistemas con una base negativa o con una base natural y un conjunto simétrico (asimétrico) de cifras es fácil realizar esta operación: el sustraendo se invierte (o sea, en vez de denotar el número x se denota el número $-x$ en este sistema) y se suman los códigos obtenidos. Para los sistemas con una base natural y un conjunto lógico de cifras, las operaciones de la adición algebraica se realizan por medio de los *códigos complementario e inverso* de estos números.

Cambemos la operación de sustracción $y - x$ por la operación de adición $y + (S - x)$ después de que sigue la disminución del resultado en S :

$$y - x = y + (S - x) - S. \quad (4.9a)$$

Introduzcamos el concepto *del código complementario* de un número x :

$$[x]_c = \begin{cases} x, & x \geq 0, \\ S + x, & x < 0. \end{cases}$$

La operación de sustracción puede cambiarse por la de adición también a base de la siguiente correlación:

$$y - x = y + (S - S^{-n} - x) - S + S^{-n} \quad (4.9c)$$

De aquí obtenemos la definición *del código inverso* del número x :

$$[x]_I = \begin{cases} x, & \text{cuando } x \geq 0, \\ S + x - S^{-n}, & \text{cuando } x \leq 0 \end{cases} \quad (4.9d)$$

De las correlaciones (4.9b) y (4.9d) obtenemos

$$[x]_c = [x]_I + S^{-n}, \quad x < 0. \quad (4.9e)$$

Según la fórmula (4.9b) el código complementario del número negativo $x \leftrightarrow x_1 x_2 \dots x_n$, $|x| < 1$, tiene la siguiente forma:

$$[x]_c = (S - 1) \cdot (S - 1 - x_1)(S - 1 - x_2) \dots (S - x_n).$$

Según la correlación (4.9e) el código inverso de este número tiene la forma

$$[x]_1 = (S - 1). (S - 1 - 1x_1)(S - 1 - x_2) \dots (S - 1 - x_n).$$

De esta manera tenemos las siguientes *reglas de formación de los códigos complementario e inverso para los números negativos*.

1. Para obtener el código inverso de un número negativo es necesario en cada orden de la denotación S -aria del número cambiar la cifra de este orden por la que la complementa hasta $S - 1$. En el orden de signo hay que escribir la cifra $S - 1$.

2. Para obtener el código complementario de un número negativo, es necesario sumar la unidad al orden inferior de su código inverso.

Basándose en las fórmulas (4.9a) y (4.9c) tenemos, respectivamente, las siguientes *reglas de la adición algebraica*.

1. Para realizar la adición algebraica de dos números x e y de signo arbitrario en el sistema de numeración de la base natural S y un conjunto lógico de cifras es suficiente escribir estos números en el código complementario, sumar los códigos obtenidos según las reglas de adición de números en el sistema de la base S y omitir la unidad de traslado del orden de signo, si ella aparece. El resultado obtenido es el código complementario de la suma algebraica de los números x e y .

2. Para realizar la adición algebraica de los números x e y de signo arbitrario en el sistema de numeración de la base natural S y un conjunto natural de cifras, es suficiente escribir estos números en el código inverso, sumar los códigos obtenidos según las reglas de adición de los números en el sistema de la base S y añadir una unidad en el orden inferior de la expresión obtenida, si al sumar aparece la unidad de traslado del orden de signo. El resultado obtenido es el código inverso de la suma verdadera (algebraica) de los números x e y .

Para multiplicar, es necesario sumar órdenes de los factores, multiplicar mantisas según las reglas de multiplicación de los números, normalizar el resultado y, si la mantisa del resultado se ha desplazado, cambiar respectivamente el orden del producto. Cuando se realiza la división, del orden del dividendo se sustrae el orden del divisor. La división de mantisas se sustituye por sustraer el divisor del dividendo hasta obtener la diferencia negativa como resultado de la sustracción de turno. Luego suman el divisor y la diferencia negativa (esta operación se llama *restablecimiento del resto*) y en calidad de cifra del cociente se escribe el número de sustracciones del divisor sin tener en cuenta la última sustracción. Luego el resto se hace dividendo y el papel de divisor lo juega el divisor antiguo desplazado en un orden a la derecha, etc. La mantisa del cociente se normaliza y, respectivamente, cambia el orden del cociente.

Como ya hemos señalado, un sistema de numeración es de valor ponderable, si para todo orden de número j se puede indicar un número p_j tal

que el equivalente cuantitativo correspondiente a una cifra a_i escrita en este orden es igual a $p_j \cdot (a_i)$, donde (a_i) es un equivalente cuantitativo correspondiente a la misma cifra en el orden nulo. A la correlación (4.8) le es análoga la siguiente expresión:

$$x = \sum_{j=-\infty}^{\infty} (a_i) \cdot p_j.$$

En la técnica de cálculo tienen interés los sistemas bidecimales de numeración, en los cuales cada cifra decimal se codifica con cuatro cifras binarias y, por consiguiente, cada orden de la inscripción decimal se sustituye por cuatro órdenes. Si a estos cuatro órdenes les corresponden unos pesos, tiene lugar un sistema de valor ponderable para estos cuatro órdenes binarios.

Al utilizar los sistemas bidecimales en los ordenadores, es deseable que la codificación (es multiforme puesto que para codificar diez cifras se puede usar cualesquiera de 16 tétradas de ceros y unidades) satisfaga unas restricciones. Citemos cinco exigencias principales formuladas por Rutishauser.

1. *Unicidad.* Es necesaria la correspondencia unívoca entre las cifras y las tétradas. Si esta exigencia no se cumple es imposible codificar y decodificar los números. En otras palabras, hace falta codificar distintas cifras decimales con las tétradas diferentes.

2. *Ordenación.* A cifras decimales mayores les deben corresponder tétradas mayores (según el equivalente cuantitativo). Es necesario cumplir esta exigencia cuando se comparan los números codificados.

3. *Paridad.* A las cifras decimales pares les deben corresponder las tétradas pares (que tienen cero (unidad) en el orden derecho extremo), mientras que a las cifras impares, las tétradas impares.

4. *Propiedad complementaria.* Si la suma de cifras del sistema decimal es igual a nueve, a estas cifras les hay que poner en correspondencia las tétradas mutuamente invertidas (es decir, obtenidas una de otra sustituyendo unidades por ceros y viceversa). Es necesario cumplir esta exigencia para introducir el código complementario o inverso en el sistema bidecimal.

5. *Valor ponderable.* Deben existir cuatro pesos p_1, p_2, p_3 y p_4 tales que, si a una cifra decimal x se le ha puesto en correspondencia una tétrada $\alpha_1\alpha_2\alpha_3\alpha_4$, tiene lugar la igualdad $x = \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3 + \alpha_4 p_4$.

Una codificación que satisface todas las cinco exigencias se denomina *perfecta*.

Acordémonos designar la tétrada puesta en correspondencia a una cifra decimal x mediante $q(x)$. Consideremos las reglas de adición para los números escritos en el sistema bidecimal. Sean tomadas cifras x e y en el sistema decimal. Entonces $x + y$ es ora una cifra nueva de este sistema (si $x + y < 10$), ora el resultado de sumar estas cifras es la cifra correspondiente a $x + y - 10$ y aparece la unidad de traslado al orden siguiente. Entonces,

en el sistema bidecimal, las reglas de adición tienen forma

$$q(x) + q(y) = \begin{cases} q(x + y), & \text{cuando } x + y < 10, \\ q(x + y - 10) + 16, & \text{cuando } x + y \geq 10, \end{cases} \quad (4.10)$$

Aquí la adición de 16 corresponde al traslado de la unidad al orden siguiente (el traslado al orden derecho de la tétada que está a la izquierda de la dada). Según se desprende de la correlación (4.10), es necesario hacer correcciones, sumando los números en tal sistemas.

Por ejemplo, sea que toda cifra decimal se codifica con su inscripción en el sistema binario empleando cuatro órdenes binarios. En este caso la cifra 5 se codificará con la tétada 0101. Tal procedimiento de codificación se denomina *código de sustitución directa* (8421). En este caso, teniendo en cuenta que para el código de sustitución directa $q(x) = x$, la correlación (4.10) puede escribirse en la forma

$$q(x) + q(y) = \begin{cases} x + y, & \text{cuando } x + y < 10, \\ x + y + 6, & \text{cuando } x + y \geq 10. \end{cases}$$

De este modo, sumando en el código de sustitución directa, es necesario en todo par de tétadas sumar según las reglas de la adición binaria, teniendo en cuenta el traslado entre las tétadas, si este último surge. Después de esto, a los órdenes, donde la suma de las cifras codificadas supera 10, hay que añadir la corrección 0110. Lo ilustremos con el siguiente ejemplo. Sea que haga falta hallar la suma de los números 205 y 768. Cumplamos las operaciones necesarias:

$$\begin{array}{r} + \quad 205 - 0010 \quad 0000 \quad 0101 \\ \quad 768 - 0111 \quad 0110 \quad 1000 \\ \hline \quad \quad 1001 \quad 0110 \quad 1101 \\ + \quad \quad 0000 \quad 0000 \quad 0110 \\ \hline 973 - 1001 \quad 0111 \quad 0011 \end{array} \begin{array}{l} \text{codificación,} \\ \\ \text{primera sumación} \\ \\ \text{resultado.} \end{array}$$

El código de sustitución directa satisface todas las exigencias de Rutherford excepto la cuarta. La infracción de esta exigencia no permite introducir el código complementario o inverso, lo que, a su vez, no permite sustituir la sustracción por la operación de adición. Para cumplir la propiedad complementaria, codifiquemos toda cifra decimal x con la tétada $q(x)$, igual a $x + 3$. El código obtenido se denomina *código por exceso de tres* o *código de Stibitz*. En este caso la correlación (4.10) toma la siguiente forma:

$$q(x) + q(y) = \begin{cases} x + y + 3, & \text{cuando } x + y < 10, \\ (x + y - 10) + 3 + 16, & \text{cuando } x + y \geq 10. \end{cases}$$

De tal modo, en el código por exceso de tres se necesita la corrección $+3(0011)$, cuando $x + y \geq 10$ y la corrección $-3(-0011:1100)$ en el código

inverso y 1101 en el complementario), cuando $x + y < 10$. Pongamos por ejemplo la sumación de los números -471 y 607 utilizando el código complementario:

$$\begin{array}{r}
 -471 : 1000 \quad 0101 \quad 1100 \\
 + 607 : 1001 \quad 0011 \quad 1010 \\
 \hline
 10001 \quad 1001 \quad 0110 \\
 + 0001 \quad 1001 \quad 0110 \\
 \hline
 0011 \quad 1101 \quad 0011 \\
 136 - 0100 \quad 0110 \quad 1001
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{codificación,} \\ \text{primera sumación,} \\ \text{omitimos el traslado del orden} \\ \text{izquierdo,} \\ \text{correcciones,} \\ \text{obtención del resultado después de ha-} \\ \text{cer correcciones.} \end{array}$$

La codificación por exceso de tres no posee del valor ponderable. La única codificación que tiene todas las cinco propiedades es la *codificación de Aiken-Emeriax*. Sus pesos son 2 4 2 1.

En calidad de ejercicio, demostrar que la codificación de pesos 2 4 2 1 es perfecta:

$$\begin{array}{l}
 0 - 0000, 1 - 0001, 2 - 0010, 3 - 0011, 4 - 0100, \\
 5 - 1011, 6 - 1100, 7 - 1101, 8 - 1110, 9 - 1111.
 \end{array}$$

Analizando los códigos, notemos que

$$q(x) = \begin{cases} x, & \text{cuando } x < 5, \\ x + 6, & \text{cuando } x \geq 5; \end{cases}$$

para determinar las reglas de adición de las tétradas en este código, es necesario considerar los siguientes casos:

1. $x \leq 5, y \leq 5, x + y \leq 5,$
 $q(x) + q(y) = x + y; q(x + y) = x + y; \Delta = 0.$
2. $x < 5, y < 5, 5 \leq x + y < 10,$
 $q(x) + q(y) = x + y; q(x + y) = x + y + 6; \Delta = 6.$
3. $x < 5, y \geq 5, 5 \leq x + y < 10,$
 $q(x) + q(y) = x + (y + 6), q(x + y) = x + y + 6; \Delta = 0.$
4. $x < 5, y \geq 5, 10 \leq x + y < 15,$
 $q(x) + q(y) = x + (y + 6), q(x + y) + 6$ (traslado,
véase (4.10)) $= x + y + 6, \Delta = 0.$
5. $x \geq 5, y \geq 5, 10 \leq x + y < 15,$
 $q(x) + q(y) = (x + 6) + (y + 6), q(x + y) + 6 = (x + y) + 6;$
 $\Delta = -6.$
6. $x \geq 5, y \geq 5, x + y \geq 15,$
 $q(x) + q(y) = (x + 6) + (y + 6), q(x + y) + 6 = (x + y + 6) + 6;$
 $\Delta = 0.$

Por lo tanto, si sumamos $x + y$ en el código de Aiken—Emeriax necesitamos la corrección +6 en las tétradas, donde $x < 5$, $y < 5$, $5 \leq x + y < 10$ y la corrección -6, donde $x \geq 5$, $y \geq 5$, $10 \leq x + y < 15$.

En este código sumamos el par ya examinado anteriormente de los números -471 y 607 empleando, además, el código complementario. Tenemos

- 471	: 1011	0010	1111	}	codificación,
+ 607	: 1100	0000	1101		
	0111	0011	1100		sumación,
	0111	0011	1100		omitimos el traslado del orden izquierdo,
	1010	0000	0000		correcciones,
+ caso 5	caso 1	caso 8			
136	: 0001	0011	1100.		

Un sistema de numeración que permite hacer cálculos en cada orden independientemente de los resultados obtenidos en otros órdenes es el código en restos. Llámase *conjunto de módulos* de un código en restos un conjunto I de los números naturales mutuamente simples: q_1, q_2, \dots, q_i .

Designemos el resto de la división del número x por q_i mediante

$\frac{x}{q_i}$. Se denomina *código en restos* de un número x conforme a un res $\frac{x}{q_i}$ conjunto de módulos q_1, q_2, \dots, q_i una expresión de tipo $\text{res } \frac{x}{q_1} \text{ res } \frac{x}{q_2} \dots \text{ res } \frac{x}{q_i}$. Si $\text{res } \frac{x}{q_i} = \alpha$ y $\text{res } \frac{y}{q_i} = \beta$, se tiene $x = nq_i + \alpha$ e $y = mq_i + \beta$. Sumemos estas igualdades. Resulta $x + y = (n + m)q_i + (\alpha + \beta)$. Ahora multipliquemos respectivamente los miembros derechos e izquierdos de las igualdades iniciales $xy = (mnq_i + n\beta + m\alpha)q_i + \alpha\beta$. Si $\alpha + \beta < q_i$ y $\alpha\beta < q_i$, se puede afirmar que, sumando y multiplicando dos números, sus restos de la división por q_i se suman o se multiplican también. Si $\alpha + \beta > q_i$ o $\alpha\beta > q_i$, dividiendo cada una de ellas por q_i y determinando el cociente entero obtenemos que la afirmación sobre la adición y la multiplicación de los restos coincide con la afirmación anterior, si consideramos que después de cumplir estas operaciones se realiza la división por el módulo del orden dado y la selección del resto verdadero. La operación semejante puede tener lugar para cada q_i independientemente, por eso el código en restos permite sumar y multiplicar por órdenes, lo que, a su vez, permite aumentar la velocidad de cumplir las operaciones en los ordenadores.

Por ejemplo, sea que existe el sistema de tres módulos: 7, 8, 9. Entonces, para los números $x = 11$ e $y = 6$, los códigos en restos son 432 y 666, respectivamente. Sumemos y multipliquemos por órdenes los códigos de x e y ; entonces obtenemos 10 9 8 y 24 18 12. En cada orden, dividamos el

resultado por el módulo de este orden y saquemos el resto verdadero; obtenemos 318 y 323, respectivamente. La suma y el producto (iguales a 17 y 66) de x e y tienen códigos en restos 318 y 323.

Para obtener la correspondencia biunívoca de los números en el código en restos debemos tener en cuenta que si el producto de todos los módulos q_i utilizados para codificar es igual a N , en el código en restos se puede codificar biunívocamente con estos módulos sólo N números distintos (de 0 a $N - 1$, de N a $2N - 1$, etc.). El número N se denomina *potencia del sistema de módulos*.

§ 4.4. Etapa algorítmica del diseño

El diseño del operador de autómatas consiste en construir una aplicación de autómatas según una descripción verbal prefijada que incluye el objetivo del diseño, la destinación del dispositivo automático que se sintetiza y las propiedades de funcionamiento del objeto controlado que junto al dispositivo automático diseñado realiza el objetivo planteado.

Por regla general, la aplicación de autómatas se prefija en forma de un grafo de transiciones. El *grafo de transiciones* es un grafo $G = \langle V(X, Y) \rangle$, cada vértice del cual corresponde biunívocamente a un estado interior de dispositivo automático; si éste pasa de un estado S_i al estado S_j , sus vértices respectivos v_i y v_j se unen mediante un arco $(v_i, v_j) \in U$ ponderado por un par de vectores de (X, Y) , mediante los cuales se realiza este paso.

En la etapa algorítmica, la información no formalmente prefijada se transforma en un sistema formal como un operador de autómatas. Al mismo tiempo se emplea el principio de analogías. Formalmente este paso puede basarse en aplicar las gramáticas, cuyas reglas de sustitución formalizan las propiedades dadas del objeto controlado. Resulta que se engendran las composiciones de los grafos de transiciones que satisfacen la descripción verbal prefijada. Muchas veces esta composición es una jerarquía de dos niveles. El primer nivel comprende los grafos de transiciones, cuya reacción es una acción directa de mando sobre el objeto controlado. El segundo nivel es el grafo de transiciones, cuya reacción corresponde a la excitación de los vértices iniciales de los grafos del primer nivel.

Examinemos la formación de los grafos de transiciones para los dispositivos automáticos que realizan las unidades de la técnica de cálculos y de la automática industrial.

Sea prefijado un dispositivo automático operacional en forma de un dispositivo aritmético de acción sucesiva representado en la fig. 4.8. Este dispositivo contiene cuatro registros $R1, R2, R3, R4$ y un sumador de un orden de tipo combinado que se comunican entre sí mediante los canales y las válvulas controlados. En el dispositivo los números de cuatro órdenes se representan en el código binario 8421.

Sinteticemos el dispositivo automático que controla la calculación de la mantisa del cociente. Antes de empezar a cumplir la operación, el dividendo a está en el registro $R1$, el

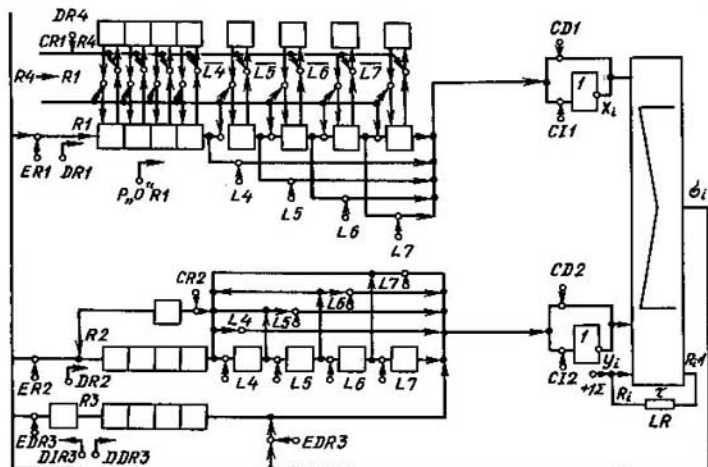


Fig. 4.8

divisor b , en el registro $R2$, los números a y b están normalizados y el cociente se forma en el registro $R3$.

Durante el cumplimiento de la operación de división sustituimos la sustracción por la adición en el código complementario y empleamos el siguiente algoritmo.

1. En el código complementario sustraemos el contenido del registro $R2(b)$ del contenido del registro $R1(a)$, copiando simultáneamente el contenido del registro $R1$ en el registro $R4$.

2. Si $a - b \geq 0$ cumplimos el punto 3, en caso contrario, el punto 4.

3. Inscríbimos la diferencia $a - b$ en el registro $R1$, ponemos el registro $R4$ en «cero», inscribimos 1 en el registro $R3$, desplazamos el divisor en un orden a la derecha y el registro del resultado a la izquierda, pasamos al punto 5.

4. Inscríbimos el contenido del registro $R4$ en el registro $R1$ y 0 en el registro $R3$, desplazamos el divisor en un orden a la derecha y el registro del resultado a la izquierda, pasamos al punto 5.

5. Si el punto 1 se ha cumplido menos que cinco veces, pasamos al punto 1, en caso contrario al punto 6.

6. El fin (en caso general, el control se transmite al bloque de normalización del cociente). Para simplificar, consideremos que el código del nulo está inscrito de antemano en el registro $R3$.

Si conocemos el dispositivo automático operacional y el algoritmo que se realiza, componemos el diagrama temporal de funcionamiento del dispositivo automático de control que es el algoritmo para cumplir esta operación en términos de puntos de control, *microoperaciones*. A cada microoperación le corresponde un canal de salida del dispositivo automático de mando. De aquí el número de canales de salida del dispositivo automático que se sintetiza es igual al número de todas las microoperaciones.

El conjunto de microoperaciones del dispositivo considerado es el siguiente: *DR1* es el desplazamiento del registro *R1*; *DR2* es el desplazamiento del registro *R2*; *DDR3* es el desplazamiento derecho del registro *R3*; *DIR3* es el desplazamiento izquierdo del registro *R3*; *DR4* es el desplazamiento del registro *R4* (la presencia del registro *R4* permite omitir el restablecimiento del resto); *CD1* es el código directo del contenido del registro *R1*; *CD2* es el código directo del contenido del registro *R2*; *CI1* es el código inverso del contenido del registro *R1*; *CI2* es el código inverso del contenido del registro *R2*; *ER1* es la entrada del registro *R1*; *ER2* es la entrada del registro *R2*; *P«0» R1* es la puesta en «ceros» del registro *R1*; *EIR3* es la entrada del registro *R3* a la izquierda; *EDR3* es la entrada del registro *R3* a la derecha, *CR1* es el ciclo del registro *R1*; *CR2* es el ciclo del registro *R2*, $+1 \Sigma$ es la entrega de la unidad en la cadena de traslado del sumador; *L4* es el funcionamiento con los vectores de longitud 4; *L5* es el funcionamiento con los vectores de longitud 5; *L6* es el funcionamiento con los vectores de longitud 6; *L7* es el funcionamiento con los vectores de longitud 7; *R4* → *R1* es la transmisión del contenido del registro *R4* al registro *R1*.

La fila del diagrama temporal corresponde biunívocamente a una microoperación.

El conjunto de microoperaciones que se cumplen simultáneamente se denomina *microinstrucción*, el conjunto de sucesiones de microinstrucciones que corresponde a una operación para cumplir se denomina *microprograma*. Cada microprograma corresponde biunívocamente a un valor del vector de entrada *X*. Por lo tanto, el número de operaciones que se cumplen, el número de canales de entrada es igual a $\lceil \log_2 | \{x_i\} | \rceil$ donde $\lceil \rceil$ es el signo del número entero más próximo.

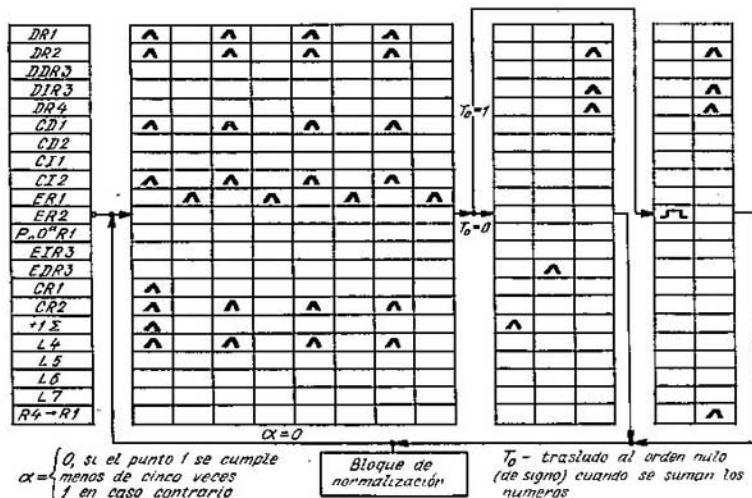


Fig. 4.9

Para sintetizar un dispositivo automático de mando que realice un algoritmo prefijado, se componen con anticipación diagramas temporales detallando cada punto del algoritmo. El diagrama temporal correspondiente a la calculación del orden nulo del cociente en el dispositivo automático operacional (fig. 4.8) según el algoritmo de división mencionado está representado en la fig. 4.9. Los diagramas temporales correspondientes a las calculaciones de los órdenes consecuentes del cociente se diferencian de los diagramas representados en la fig. 4.9 sólo en el primer bloque, a saber: calculando i -ésimo orden del cociente en vez de cumplir la microoperación $L4$ cuatro veces se cumple respectivamente la microoperación $L(4 + i)$ $4 + i$ veces, es decir, calculando el primer orden se cumple la excitación de $L5$ cinco veces, etc. Por consiguiente, para utilizar el primer bloque temporal hace falta «sintonizarlo» antes de calcular el orden correspondiente del cociente.

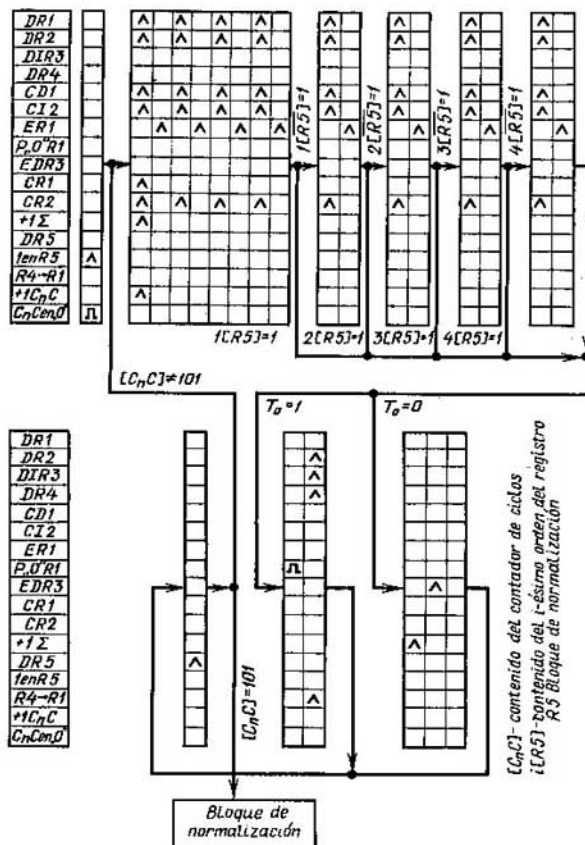


Fig. 4.10

La sintonización del primer bloque temporal consiste en lo siguiente: L_j se examina como un parámetro de entrada de este bloque y antes de calcular el orden nulo se adjudica a j el valor de 4 y después de calcular todo orden del cociente el valor de j se aumenta en 1. Sintonicemos el primer bloque temporal introduciendo un registro complementario de cuatro órdenes $R5$, antes de empezar a cumplir la operación de división se inscribe 1 en este registro y al cabo de calcular cada orden del cociente el registro $R5$ se desplaza en un orden. La salida del i -ésimo orden del registro $R5$ se transmite a la válvula $L(|i| \neq 3)$.

Para hallar el número de órdenes calculados del cociente, introduzcamos un contador de tres órdenes (C_nC), cuyo estado 101 indica que cinco órdenes del cociente están calculados y hace falta transmitir el mando al bloque de normalización.

Basándose en el análisis realizado, es conveniente cambiar el conjunto de microoperaciones, a saber: en vez de las microoperaciones $L4$, $L5$, $L6$ y $L7$ introducir las microoperaciones $DR5$ (desplazamiento del registro $R5$) y 1 en $R5$ (inscripción de la unidad en el registro $R5$) y, también, introducir complementariamente las microoperaciones $+1 C_nC$ (adición de la unidad al contador de ciclos C_nC) y C_nC en «O» (puesta del contador de ciclos en «O»).

De las transformaciones consideradas resulta que el diagrama dado en la fig. 4.9 se transforma en el diagrama temporal representado en la fig. 4.10.

El diagrama temporal no da la aplicación de autómatas. En efecto, a un valor del vector de entrada que determina la operación para cumplir le corresponde varios valores del vector de salida (de las microinstrucciones), es decir, el diagrama temporal prefija una transformación no unívoca del vector de entrada en el de salida. Por consiguiente, esta transformación no es aplicación de autómatas. Para obtenerla es necesario introducir la memoria en el dispositivo automático que se sintetiza. Con ello, el

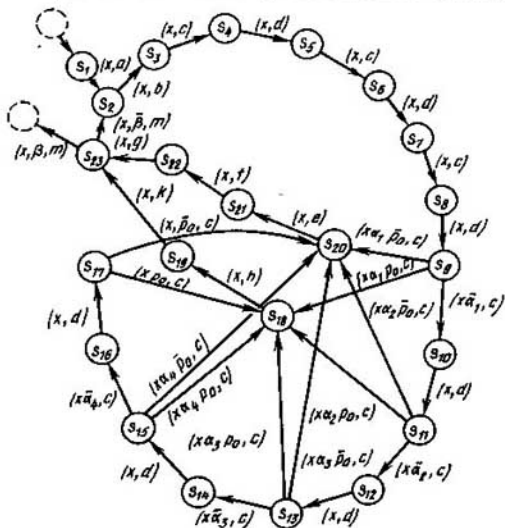


Fig. 4.11

volumen de la memoria, igual al número de todas las microinstrucciones, es premeditadamente suficiente. El conjunto del vector de entrada y del estado de la memoria determina unívocamente la microinstrucción.

En correspondencia con el diagrama temporal construyamos el grafo de transiciones del dispositivo automático. Su estado interior se pone en correspondencia biunívoca a una columna del diagrama temporal.

El grafo de las transiciones del dispositivo automático que realiza el diagrama temporal (fig. 4.10) se representa en la fig. 4.11. En esta figura, las microinstrucciones se denotan por letras latinas y tienen siguiente forma:

$$\begin{aligned} a &= \{1 \text{ en } R5, C_n C \text{ en } \langle O \rangle\}; \\ b &= \{DR1, DR2, CD1, C12, CR1, CR2, +1\Sigma; +1C_n C\}; \\ c &= \{ER1\}; d = \{DR1, DR2, CD1, C12, CR2\}; \\ e &= \{+1\Sigma\}; f = \{EDR3\}; g = \{DR2, DIR3, DR4\}; \\ h &= \{P_n O \text{ en } R1\}; k = \{DR2, DIR3, DR4, R4 \rightarrow R1\}; m = \{DR5\}. \end{aligned}$$

Consideremos que si la unidad ($x = 1$) llega a la entrada del dispositivo automático de mando, éste controla la división.

Las variables lógicas (etiquetas) que determinan el mando las designaremos mediante α_i , contenido del i -ésimo orden del registro $R5$; β es el estado convencional del contador $C_n C$;

$$\beta = \begin{cases} 1, & \text{si el estado del contador es } 101, \\ 0 & \text{en caso contrario;} \end{cases}$$

T_0 es el traslado al orden nulo cuando se suman los números.

Examinemos un sistema grande de la automática industrial, sistema para quemar el combustible sólido en una capa densa. El sistema comprende cuatro hogares semigasógenos, instalaciones de alimentación de combustible y de aire, así como la instalación de desescoriado. En la fig. 4.12 se representa el esquema del hogar semigasógeno, donde 1 es el transportador; 2 es el arado; 3 es la tolva de entrada; 4 es el combustible; 5 es el alimentador; 6 es el cargador; 7 es el semigas; 8 es la alimentación de aire; 9 es la cámara de trabajo, 10 es la escoria; 11 es la capa del combustible recién cargado; 12 es la zona de reducción; 13 es la zona de combustión; 14 es la zona de escoria; 15 es el emparillado.

Los hogares semigasógenos son un tipo de los hornos industriales y forman parte del servicio de gas de una fábrica. La automatización del mando de los procesos de combustión en los hogares es una de las tareas de automatización de todo el proceso tecnológico. La completitud de gasificación la determina el porcentaje de bióxido carbónico en el semigas. Si el porcentaje aumenta, el calor de combustión del semigas disminuye y la temperatura del hogar aumenta. Antes de salir del hogar, el semigas se mezcla con el aire secundario y se suministra a la cámara de trabajo. El hogar se alimenta con combustible por medio de un sistema compuesto de un transportador, un lanzador de tipo arado, una tolva de entrada, un alimentador y un cargador. La escoria se quita empleando sacudidoras, una válvula de arranque y vagones. El aire destinado para quemar el combustible se suministra utilizando la válvula de alimentación del aire primario.

El mando del proceso de gasificación se realiza manteniendo un espesor determinado de la capa del combustible que se quema (de 200 a 400 mm para antracita, de 400 a 800 mm para hulla y lingito), es decir, se determina por la intensidad de alimentación de combustible, del aire primario y de extracción de escoria de la zona de combustión.

El sistema de mando de combustión tiene los siguientes canales (fig. 4.13) que son microoperaciones en el diseño del dispositivo automático de mando: $K1$ es el arranque y la parada del proceso de gasificación; $K2$ es la indicación del funcionamiento del proceso de gasifica-

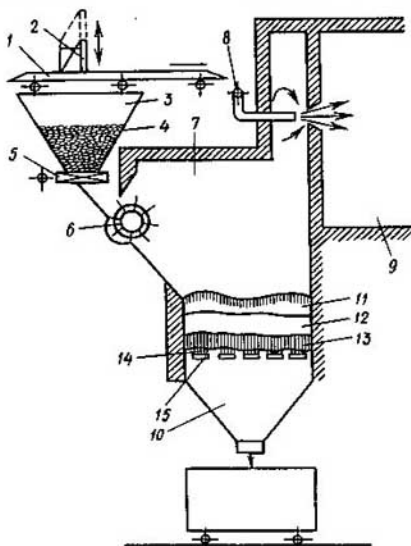


Fig. 4.12

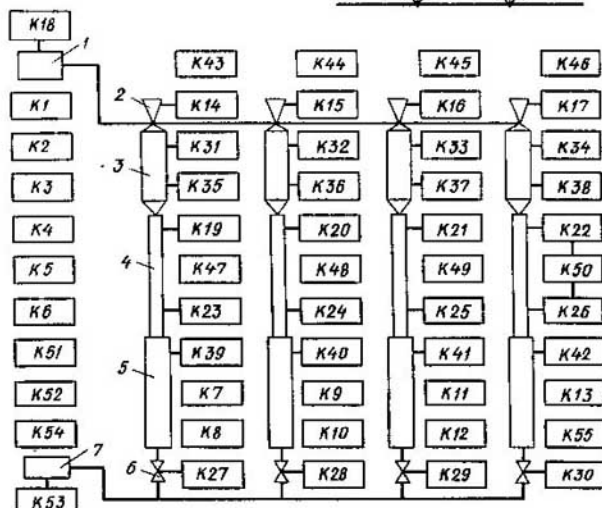


Fig. 4.13

ción; $K3$ es el arranque y la parada de llenado de las tolvas de entrada; $K4$ es la indicación de llenado de las tolvas de entrada; $K5$ es el arranque y la parada de alimentación de los hogares con combustible; $K6$ es la indicación de alimentación de los hogares con combustible; $K7$ es la detención de gasificación en el primer hogar; $K8$ es la indicación del estado del canal $K7$; $K9$ es la detención de gasificación en el segundo tolva; $K10$ es la indicación del estado del canal $K9$; $K11$ es la detención de gasificación en la tercera hogar; $K12$ es la indicación del estado del canal $K11$; $K13$ es la detención de gasificación en el cuarto hogar; $K55$ es la indicación del estado del canal $K13$; $K14$ es el arado del primer hogar; $K15$ es el arado de la segunda tolva; $K16$ es el arado de la tercera tolva; $K17$ es el arado de la cuarta tolva; $K18$ es el transportador; $K19...K22$ son los alimentadores de la primera, ..., la cuarta tolva, respectivamente; $K23...K26$ son los cargadores; $K27...K30$ son las válvulas de alimentación con aire secundario; $K31...K34$ son los captadores del nivel superior en las tolvas de entrada; $K39...K42$ son los captadores de alimentación con combustible; $K43...K46$ es la indicación de avería de la instalación para llenar las tolvas de entrada; $K47...K50$ es la indicación de avería de la instalación para alimentar hornos con combustible; $K51$ y $K52$ es el arranque y la parada del proceso de control, respectivamente; $K53$ es la indicación de control; $K54$ son las instalaciones iniciales de los elementos ejecutivos.

En la fig. 4.13: 1 es el transportador; 2 es el arado; 3 es la tolva de entrada; 4 es el sistema de alimentación de combustible; 5 es el hogar; 6 es la válvula de alimentación de aire; 7 es la válvula común.

El conjunto enumerado de canales se incluye en el portador de un modelo Ψ_a que formaliza el funcionamiento del objeto controlado. Además de este conjunto, el portador del modelo Ψ_a puede incluir un conjunto de elementos complementarios que son identificadores de los canales interiores, de entrada y salida introducidos para lograr el control eficaz de autómatas. Partiendo de esta información obtenida del tecnólogo en el portador se establecen relaciones de causa-efecto tipo $A \rightarrow B$ que determinan la signatura del modelo Ψ_a . Es obvio, que este modelo es simétrico y puede representarse en forma del mografo $G^M(\Psi_a)$. Se puede también expresar las relaciones de causa-efecto en forma de diagramas temporales. El paso del mografo o diagramas temporales al grafo de transiciones es análogo al paso de diagramas temporales (que pueden representarse en forma del mografo) al grafo de transiciones de un dispositivo de control de microprograma.

Así, pues, en esta etapa de diseño de los dispositivos automáticos se realiza el paso de la aplicación $A \rightarrow B$ a una aplicación de autómatas $XS^+ \rightarrow S^- Y$, donde X es el vector de entrada, Y es el vector de salida, o reacción del dispositivo automático, S^+ es el identificador del estado interior, en el cual pasa el dispositivo automático en el momento examinado, S^- es el identificador del estado interior, al cual el dispositivo automático pasa del estado S^+ bajo la influencia del vector de entrada X .

§ 4.5. Diseño abstracto del dispositivo automático

La capacidad de memoria $\{S_i\}$ de un dispositivo automático introducida en la etapa de diseño algorítmica puede ser excesiva, lo que se puede eliminar encolando estados equivalentes. La transformación dada se refiere a la etapa de diseño abstracto.

Los estados se denominan *equivalentes* cuando el dispositivo automático, encontrándose en éstas, elabora una misma sucesión de salida para cualquier sucesión de entrada.

De esta definición se desprende que si, encolando los vértices correspondientes, se sustituye cada clase de estados equivalentes por un estado, el grafo obtenido de transiciones representará la misma aplicación $X \rightarrow Y$ que el grafo inicial.

Huffman propuso el método de *minimización abstracta del dispositivo automático* basado en el encolamiento de estados equivalentes. Este método consiste en formación sucesiva de clases de estados equivalentes con ayuda de tablas de salidas y de transiciones. Examinemos el método de Huffman en un ejemplo.

Ejemplo 4.1. Sea que después de realizar la etapa algorítmica de síntesis se ha obtenido el grafo de transiciones representado en la fig. 4.14, a.

Construimos la *tabla de salidas* que es una tabla bidimensional (tabla 4.3), a cada fila de la cual le corresponde biunívocamente un valor del vector de entrada X ; a cada columna, un estado interior del dispositivo automático, y en la intersección de i -ésima fila con j -ésima columna se halla un valor del vector de salida Y que se elabora a la salida cuando el dispositivo automático está en el j -ésimo estado interior y a la entrada se da el i -ésimo vector.

Tabla 4.3

X_i	S_j						
	S_1	S_2	S_3	S_4	S_5	S_6	S_7
0	1	1	0	0	1	0	0
1	0	0	1	1	0	1	1

Si, en la tabla de salida, a dos estados interiores del dispositivo automático les corresponden distintos valores de las columnas, los estados no son equivalentes, puesto que en ellos las aplicaciones $X \rightarrow Y$ son distintas. De antemano partamos todo el conjunto de estados en las clases de *estados convencionalmente equivalentes*, a saber: en una misma clase se incluyen los estados, a los cuales les corresponden los valores iguales de las columnas en la tabla de salidas: $K_1 = \{S_1, S_2, S_3\}$; $K_2 = \{S_3, S_4, S_6, S_7\}$.

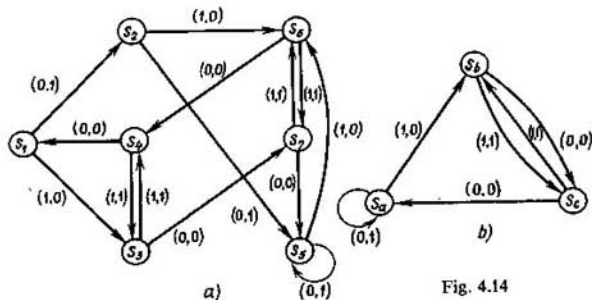


Fig. 4.14

Si la tabla de salidas se examina como una matriz de incidencia de un modelo, los estados interiores se hallan en una misma clase cuando en su respectiva matriz de frecuencia de las relaciones las frecuencias propias y mutuas de los estados interiores son iguales una a otra:

$$\frac{\partial G^M}{\partial S} (S_a, S_b) = \frac{f_{aa} - 2f_{ab} + f_{bb}}{f_{ab}} = 0.$$

Para que los estados interiores del dispositivo automático sean equivalentes es insuficiente tener una misma correspondencia $X \rightarrow Y$ solamente en estos estados; es necesario que para cualquier otra transición posible de estos estados la aplicación $X \rightarrow Y$ sea la misma.

Para verificar este dato construimos la *tabla de transiciones* (tabla 4.4). En esta tabla, a cada fila y columna les corresponden los mismos valores que en la tabla de salidas y en la intersección de i -ésima fila con j -ésima columna se encuentra una clase de estados convencionalmente equivalentes, bajo la influencia X_i el dispositivo automático pasa del estado S_j a esta clase.

Tabla 4.4

X_i	S_j						
	S_1	S_2	S_3	S_4	S_5	S_6	S_7
0	K_1	K_1	K_2	K_1	K_1	K_2	K_1
1	K_2	K_2	K_2	K_2	K_2	K_2	K_2

Si la clase de estados convencionalmente equivalentes escogida en el paso anterior no es clase de estados equivalentes, a sus estados les corresponden valores distintos de las columnas; esto significa que, para las siguientes transiciones, las aplicaciones $X \rightarrow Y$ son diferentes para los estados dados.

Cada clase K_i se parte en nuevas clases de estados convencionalmente equivalentes con la particularidad de que una misma clase comprenda todos los estados de la clase K_i con los números iguales de columnas: $K_1 = \{S_1, S_2, S_5\}$, $K_2^1 = \{S_3, S_6\}$, $K_2^2 = \{S_4, S_7\}$.

Componemos la matriz de frecuencia de las relaciones según esta tabla de transiciones determinando, al mismo tiempo, la multiplicación como $K_i \times K_i = 1$, $K_i \times K_j = 0$ ($i \neq j$). Entonces cuando se parten las clases obtenidas en el paso anterior los estados interiores se encuentran en una misma clase, si las frecuencias propias y mutuas de estos estados son iguales una a otra, o sea,

$$\frac{\partial G^M}{\partial S} (S_a, S_b) = 0.$$

Al formar las clases K_1 , K_2^1 , K_2^2 volvemos a construir la tabla de transiciones, etc. hasta que cada clase de estados convencionalmente equivalentes formada en el paso anterior sea constante.

Construyamos la tabla de transiciones (tabla 4.5) teniendo en cuenta la partición de la clase K_2

Tabla 4.5

X_i	S_j						
	S_1	S_2	S_3	S_4	S_5	S_6	S_7
0	K_1	K_1	K_2''	K_1	K_1	K_2''	K_1
1	K_2'	K_2'	K_2''	K_2'	K_2'	K_2''	K_2'

Del análisis de esta tabla deducimos que todos los estados de cada una de las clases formadas «se portan como amigos»: bajo la influencia de X pasan a la misma clase que en la salida permite obtener un mismo valor de Y para todos los estados de la clase al realizar una transición. Por consiguiente, todos los estados de una clase se portan como un estado que los sustituye.

Sustituyamos las clases K_1 , K_2' , K_2'' por los estados interiores S_a , S_b , S_c , respectivamente. Como resultado obtenemos el grafo minimizado de transiciones (fig. 4.14, b) que prefija la misma aplicación $X \rightarrow Y$ que el grafo inicial de transiciones (fig. 4.14, a). Para ilustrarlo realicemos tres experimentos.

Sea dada la sucesión temporal en forma $X(t) = 011010$ a la entrada del dispositivo automático. En los casos primero y segundo el dispositivo automático está prefijado por el grafo inicial de transiciones (fig. 4.14, a) respectivamente a los estados iniciales S_1 y S_2 . En el tercer caso el dispositivo automático está prefijado por el grafo minimizado (fig. 4.14, b) con el estado inicial S_a . Determinemos sucesiones temporales de salida $Y(t)$ para cada caso.

Reduzcamos los resultados de todos los tres experimentos en la tabla (tabla 4.6).

Tabla 4.6

t		t_1	t_2	t_3	t_4	t_5	t_6
	$X(t)$	0	1	1	0	1	0
$S_n = S_1$	$S(t+1)$	S_2	S_6	S_7	S_5	S_6	S_4
	$Y(t)$	1	0	1	0	0	0
$S_n = S_2$	$S(t+1)$	S_5	S_6	S_7	S_5	S_6	S_4
	$Y(t)$	1	0	1	0	0	0
$S_n = S_a$	$S(t+1)$	S_c	S_b	S_c	S_c	S_b	S_c
	$Y(t)$	1	0	1	0	0	0

Si las dimensiones de la tabla de salidas y transiciones son grandes para la minimización necesaria del grafo de transiciones por medio del ordenador, es conveniente emplear matrices de frecuencia de las relaciones correspondientes a estas tablas, lo que simplifica los cálculos. Con ello, se minimizan los grafos de transiciones basándose en la afirmación siguiente.

Los estados interiores S_a y S_b son equivalentes si, y sólo si, las derivadas $\frac{\partial G}{\partial S}(S_a, S_b)$ de los grafos modelos, correspondientes a las tablas de salidas

y transiciones, son iguales a cero

$$\frac{\partial G}{\partial S} (S_a, S_b) = \frac{2f_{aa} - 2f_{ab} + f_{bb}}{f_{ab}} = 0$$

en todo paso de la partición del conjunto de estados interiores en clases.

La minimización de la capacidad de la memoria, por ejemplo, cuatro veces, ahorra sólo dos elementos de la memoria. Un medio más actual y potente para optimizar toda la estructura del dispositivo automático es la *descomposición abstracta de los dispositivos automáticos*, especialmente descomposición paralela de los dispositivos automáticos. Por una parte, la necesidad de buscarla se debe a las exigencias modernas que deben satisfacer la acción rápida del control (que condiciona el uso del control paralelo de objeto). Por otra parte, el aumento de la fiabilidad del control automático exige la partición del dispositivo automático en una serie de los dispositivos automáticos de dimensiones menores que no son ligados entre sí funcionalmente y garantizan la realización del operador obtenido de autómeta.

La búsqueda de la descomposición paralela de los dispositivos automáticos abstractos se reduce a la *descomposición* del grafo de transiciones en el *producto cartesiano parcial* de los grafos más simples conforme al número de vértices. Descomponer el grafo G en el producto cartesiano parcial significa hallar los grafos G_i , $i = 1, \dots, n$, tales que $G \subset G_1 \times G_2 \times \dots \times G_n$.

Para una clase de los dispositivos automáticos de *mando de microprogramas*, importante en la técnica de cálculos, se puede resolver el problema de construir la descomposición paralela utilizando las propiedades típicas de los dispositivos automáticos de esta clase.

Al examinar los dispositivos automáticos de *microprogramas* referiremos a los elementos de los cuales se toma la información que caracteriza la marcha de cálculos a las cadenas de reacción incluidas en el bloque de memoria del dispositivo automático. Los dispositivos automáticos de microprogramas poseen una propiedad específica: el vector de entrada X no cambia desde el comienzo hasta el fin del funcionamiento del dispositivo automático hasta que se cumpla la operación prefijada. Empleando esta propiedad detallamos el concepto del dispositivo automático de microprogramas.

Introduzcamos con anticipación algunos conceptos. Se denomina *zung* G_z un grafo ponderado en el que existe al menos un vértice, a través del cual pasan todos los circuitos del grafo, y no existe ningún camino que no sea parte de tal circuito.

Un grafo se denomina *reducible al zung*, si transforma en el zung cuando se introducen en él no más de un vértice y sus arcos incidentes.

Un dispositivo automático se denomina *de microprogramas*, si su grafo de transiciones se reduce al zung con el vértice inicial que corresponde al

comienzo y al fin del funcionamiento del dispositivo automático para cualquier vector de entrada X que determina la operación realizada y no varía en estados intermedios. En caso general el dispositivo automático de microprogramas realiza varias operaciones a cada una de las cuales le corresponde un microprograma. Durante la construcción del grafo de transiciones que realiza estos microprogramas encolaremos los estados compatibles.

Estados interiores S_i y S_j se denominan *compatibles* $S_i \equiv S_j$, si cualquier vector de entrada que llega al dispositivo automático en estado S_i no coincide y no integra ningún vector de entrada que llega al dispositivo automático en estado S_j : $X_{S_i} \not\subset X_{S_j}$.

Un subgrafo $G' = \langle V', U' \rangle$ de un grafo $G = \langle V, U \rangle$ se llama *bincente*, si a toda su vértice le son incidentes dos, y sólo dos, grupos de arcos paralelos que entran y salen, tal vez excepto un elemento minimal $v^+ \in V'$ y un elemento maximal $v^- \in V'$ del conjunto V' con la particularidad de que en el vértice v^+ pueden entrar y del vértice v^- pueden salir varios

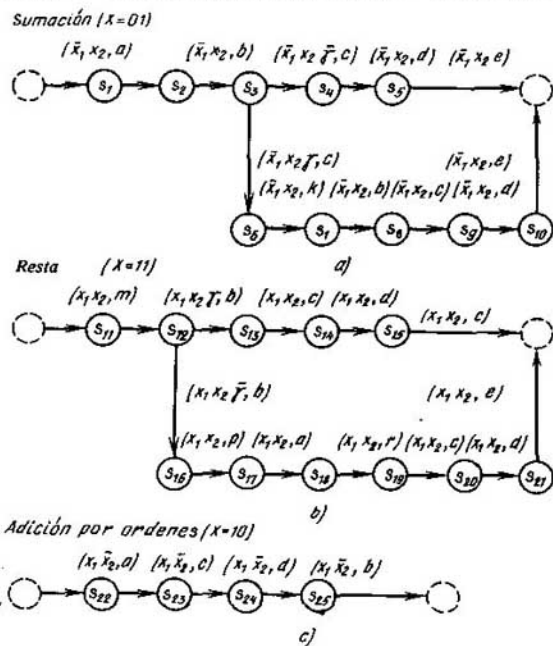


Fig. 4.15

arcos no paralelos. Las sucesiones de microinstrucciones que ponderan el subgrafo bincidente se denominan *microrrayo* y el número de microinstrucciones que integran la sucesión se llama su *longitud*.

Al sintetizar el dispositivo automático de mando que realiza varios microprogramas encolamos estados interiores compatibles teniendo en cuenta dos restricciones. En primer lugar, debemos encolar empezando por los estados finales correspondientes a operaciones distintas. En segundo lugar, encolamos los estados compatibles que se elaboran en la salida del dispositivo automático mediante una misma microinstrucción.

Examinemos el dispositivo automático de microprogramas para controlar el procesador del ordenador que cumple las operaciones «Sumación» ($a + b$), «Sustracción» ($a - b$) y «Adición por órdenes». En la fig. 4.15 se representan los grafos de transiciones construidos por diagramas temporales. En este dibujo las microinstrucciones se designan por letras latinas: $a = \{BD\}$; $b = \{CR\}$; $c = \{S\Sigma\}$; $d = \{C_nB, BC, B \rightarrow C\}$; $e = \{CC, R\}$; $k = \{BD, C_nB, BA, +1T30D\}$; $m = \{CA, BD\}$; $p = \{CA, CB\}$; $r = \{CR, C_nB\}$.

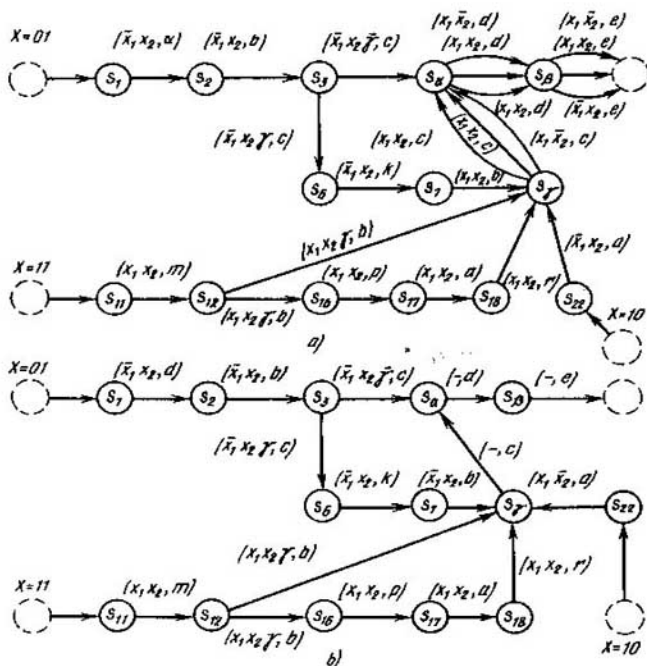


Fig. 4.16

Las microoperaciones, que son los elementos de microinstrucciones, se descifran del modo siguiente: $\bar{C}A$ es el complemento del registro A ; BD es el borrado del registro D ; CR es el comienzo del recorrido; $\bar{C}B$ es el complemento del registro B ; C_nB es el cálculo del registro B ; $S\Sigma$ es la salida de la suma Σ ; BA es el borrado del registro A ; $+1730D$ es «+1» en el trigger 30 del registro D ; BC es el borrado del registro C ; $B \rightarrow C$ es la transmisión del registro B al registro C ; C_nC es el cálculo del registro C ; R es la respuesta sobre el cumplimiento de la operación.

El resultado de los cálculos que se realizan se caracteriza por una variable lógica TOD , valor del trigger de cero del registro D (en la fig. 4.15 y, a continuación, TOD se denota por γ).

Encolemos los estados compatibles. Como resultado obtenemos el grafo unido de transiciones G (fig. 4.16, a). Sobre este grafo en los estados S_γ , S_u y S_p completamos la definición de la aplicación $X \rightarrow Y$ añadiendo $(\bar{x}_1, \bar{x}_2, c)$, $(\bar{x}_1, \bar{x}_2, d)$ y $(\bar{x}_1, \bar{x}_2, e)$, respectivamente, (fig. 4.16, b). Después de haber encolado los estados compatibles contraigamos los subgrafos bincidentes. Como *contracción* de subgrafos bincidentes se comprende la sustitución de este subgrafo por un vértice, al cual pondera el microrrayo correspondiente. Los microrrayos tienen siguiente forma: $A = abc$; $B = de$; $D = kb$, $E = mb$; $M = par$.

Como resultado de la contracción de los subgrafos bincidentes obtenemos el grafo del dispositivo automático de mando (fig. 4.17, a) con la determinación interrumpida: varias microinstrucciones corresponden a un estado que, a su vez, corresponde al vértice, al cual está contraído el subgrafo bincidente.

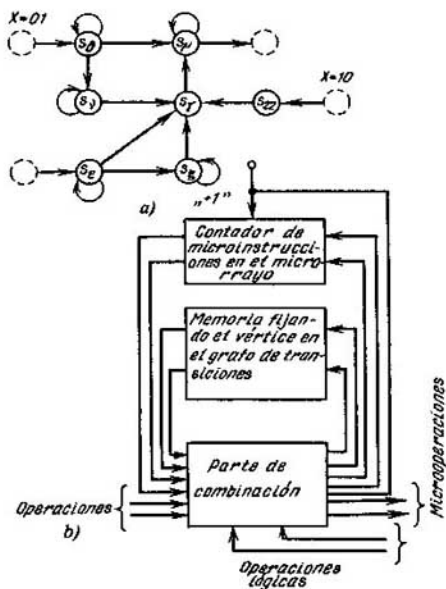


Fig. 4.17

El grafo inicial de transiciones (fig. 4.16, *b*) se obtiene como el producto cartesiano parcial del grafo G (fig. 4.17, *a*) y un grafo bincidente, cuyos fin y origen se unen por un arco y el número de vértices es igual a la longitud maximal de microrrayos considerados (en este caso a tres). El grafo bincidente, cuyos fin y origen se unen por un arco se realiza como un contador.

De tal modo conservamos el carácter determinado del dispositivo automático y la equivalencia de su funcionamiento utilizando en la realimentación del dispositivo automático el contador de microinstrucciones en el microrrayo que aumenta su valor hasta el número igual a la longitud maximal del microrrayo $l_{m\acute{a}x}$ (fig. 4.17, *b*).

Pasando al vértice v_G , que corresponde a un subgrafo bincidente contraído G_i , en el contador se pone el número igual a $l_{m\acute{a}x} - l_i$, donde l_i es la longitud del microrrayo que pondera este subgrafo. Con ello, los l_i estados del contador se ponen en correspondencia biunívoca a las microinstrucciones correspondientes al vértice v_G . Después de cada transición en el subgrafo bincidente se suma 1 al contenido del contador. Por lo tanto, el conjunto del código del vértice v_G , y el contenido del contador determina biunívocamente la microinstrucción en ejecución. El relleno en exceso del contador indica que el dispositivo automático ha salido del estado correspondiente al vértice v_G .

Las transiciones organizadas de tal modo permiten no excitar las salidas de la parte de combinación del dispositivo automático que van a la realimentación cuando se cumple el microrrayo, puesto que el estado que memoriza el vértice v_G , no cambia y el contador pasa automáticamente de un estado a otro al adicionar 1 o con ayuda de la microoperación $+1C_n\mu$ (al contador de microinstrucciones se suma la unidad) que amplía el conjunto de microoperaciones. Durante la utilización del último procedimiento se excita sólo una salida orientada a la reacción.

Realicemos transformaciones sucesivas del grafo de transiciones concentrando estados no encadenados. Dos estados interiores (dos vértices) del grafo de transiciones se denominan *no encadenados*, si no integran ningún camino simple del grafo de transiciones cuando funciona el dispositivo automático sintetizado.

Para realizar la transformación propuesta del grafo de transiciones construyamos el grafo G_3 del modo siguiente. A cada vértice del grafo de transiciones le pongamos en correspondencia biunívoca un vértice del grafo G_3 ; un par de vértices $v_a, v_b (v_a \neq v_b)$ del grafo G_3 es adyacente, si en el grafo de transiciones existe un camino simple que pasa por los vértices correspondientes a v_a y v_b cuando funciona el dispositivo automático sintetizado. Denominaremos *grafo de encadenamiento* el grafo G_3 construido de tal modo.

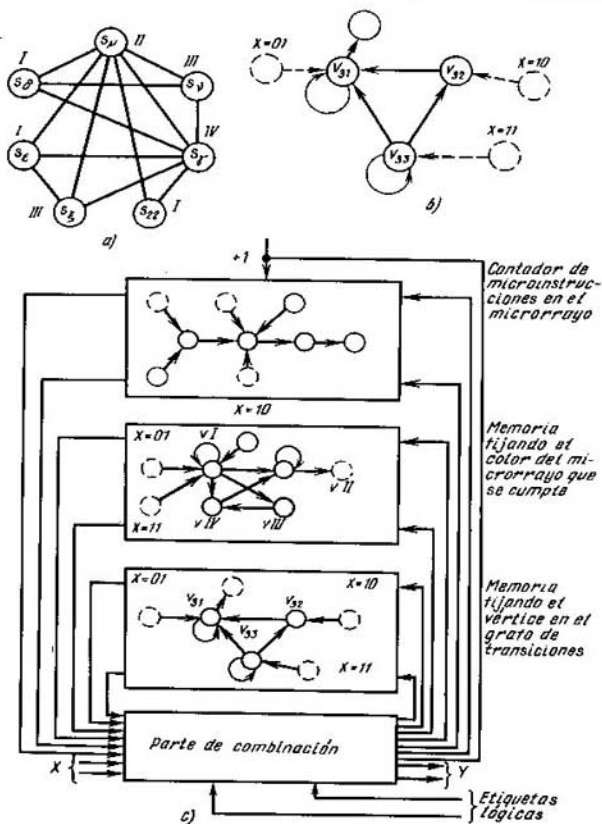


Fig. 4.18

Empleando la operación de colorear los vértices del grafo partamos todo el conjunto de vértices del grafo de encadenamiento en subconjuntos, cada uno de los cuales comprende vértices correspondientes a los estados interiores no encadenados del dispositivo automático que sintetizamos.

En la fig. 4.18, a se dan el grafo de encadenamiento y su coloración para el dispositivo automático representado en la fig. 4.17, a. Al colorear el grafo de encadenamiento tenemos los subconjuntos siguientes de estados,

cada uno de los cuales consta de los estados no encadenados entre sí:

$$V_I = \{S_0, S_{22}, S_\varepsilon\}, V_{II} = \{S_w, S_\xi\}, V_{III} = \{S_\mu\}, V_{IV} = \{S_\gamma\}.$$

De cada subconjunto $V_i (i = I, \dots, IV)$ tomamos por un elemento, por ejemplo, $S_0, S_\gamma, S_\varepsilon$, de éstos formemos el subconjunto $V_{31} = \{S_0, S_\mu, S_\gamma, S_\varepsilon\}$. Luego de cada subconjunto V_i volvemos a tomar por un elemento entre los que se quedan y a formar un subconjunto de éstos. Elegimos hasta que cada uno de los subconjuntos V_i se transforme en un conjunto vacío. Los subconjuntos formados como resultado constan de estados encadenados. En este caso tienen siguiente forma:

$$V_{31} = \{S_0, S_\mu, S_\gamma, S_\varepsilon\}, V_{32} = \{S_{22}, S_\xi\}, V_{33} = \{S_\varepsilon\}.$$

Cada subconjunto compuesto de estados encadenados se sustituye por un estado, cuyo peso es un conjunto de microrrayos correspondientes a los vértices que se unen (en caso particular el microrrayo puede comprender una microinstrucción). Esta sustitución se denominará *concentración de estados encadenados*. Después de concentrar los estados encadenados del grafo de transiciones (fig. 4.17, a) obtenemos el grafo de transiciones G_n (fig. 4.18, b).

Las concentraciones de estados encadenados interrumpen también la determinación del dispositivo automático. Conservamos la determinación y la equivalencia del dispositivo automático que se sintetiza utilizando en la realimentación un elemento de memoria que fije el color correspondiente al microrrayo que se cumple en el momento dado (fig. 4.18, c). Entonces el código del vértice en el grafo final de transiciones, el color y el contenido del contador de microinstrucciones del microrrayo en su conjunto determinan biunívocamente la microinstrucción que se cumple. Después de esta transformación, el grafo de transiciones puede representarse como el producto cartesiano parcial de los grafos correspondientes (fig. 4.18, c), lo que simplifica el proceso de la codificación de estados interiores y disminuye de hecho los gastos para aparatos cuando se sintetiza el esquema de excitación de canales a reacción inversos del dispositivo automático.

Si en el grafo de transiciones correspondiente a los microprogramas que se realizan no hay circuitos, es conveniente descomponerlo sólo en dos grafos excluyendo el grafo que indica el cambio de colores. Al mismo tiempo, se realiza la contracción de los subgrafos bincidentes en cada grafo de transiciones correspondiente a la operación que se realiza y, después, se encolan los estados compatibles.

Si el grafo de transiciones es un árbol, en la realimentación del dispositivo automático se queda solamente el contador. Además, el contador realiza el grafo de transiciones, si se memorizan o presentan valores de las variables lógicas que caracterizan los cálculos. Al estado interior inicial del dispositivo automático se le pone en correspondencia un código inicial en el contador, por ejemplo, el código 0. Los estados interiores, a los que el dispositivo automático pasa del estado S_i con el código A , se codifican como $A + 1$.

Los códigos cambian por medio de adicionar la unidad al contador a cada transición. Entonces, si existe la ramificación, ciertos estados interiores tendrán códigos iguales. Además, la determinación del dispositivo automático no se interrumpe debido al almacenamiento de los índices lógicos que caracterizan los cálculos. Las indicaciones del contador y los valores de las variables lógicas determinan unívocamente la microinstrucción necesaria. Esta afirmación es válida, ya que el grafo de transiciones es un árbol, o sea, no contiene ciclos. Al finalizar las operaciones se realiza la extinción de contadores.

Representando el grafo de transiciones correspondiente al dispositivo automático de mando de una operación en forma del producto cartesiano parcial, realizamos las siguientes transformaciones:

- 1) encolamiento de estados pseudoequivalentes;
- 2) contracción de los subgrafos bincidentes, además como resultado de cambiar la sucesión de aplicación de estas transformaciones, surgen algunas variantes equivalentes.

Examinemos el grafo de transiciones para controlar la operación de división (véase la fig. 4.12). Concentrando los subgrafos bincidentes, obtenemos el grafo de transiciones G (fig. 4.19, *a*). A los vértices de este grafo les corresponden los siguientes microrrayos: $S_1 - a$; $S_2 - bcdbcddc$; $S_b - dc$; $S_d - dc$; $S_e - dc$; $S_f - hk$; $S_g - efg$; $S_{23} - m$. Por consiguiente, para restablecer el funcionamiento equivalente hace falta tener el contador de ocho estados.

Se puede encolar los vértices correspondientes a los estados S_f y S_g del grafo de transiciones al haber memorizado el indicio p_0 ; la determinación del dispositivo automático con

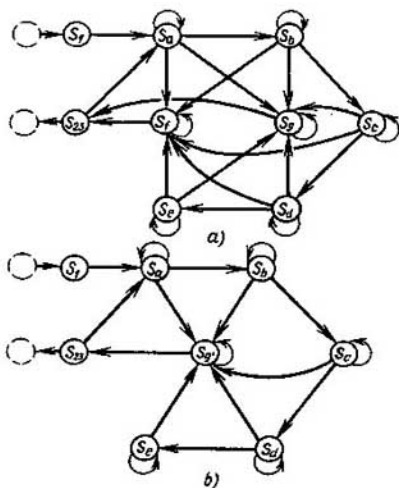


Fig. 4.19

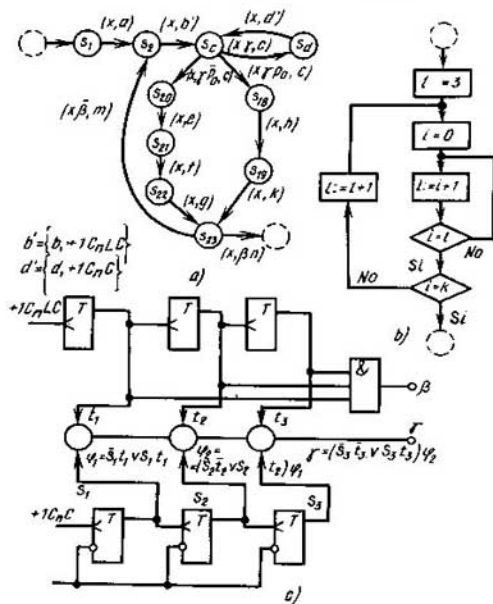


Fig. 4.20

ello, no se interrumpe. En definitiva, tenemos el grafo de transiciones G , representado en la fig. 4.19, b. Los estados del contador, el indicio p_0 y el código del vértice del grafo de transiciones (fig. 4.19, b) determinan unívocamente la microinstrucción que se cumple.

Los estados interiores se denominan *pseudoequivalentes*, si una misma microinstrucción se realiza en ellos. Para el grafo de transiciones G_T (véase la fig. 4.11) tenemos los siguientes conjuntos de estados pseudoequivalentes: $M_c = \{S_3, S_5, S_7, S_9, S_{11}, S_{13}, S_{15}, S_{17}\}$; $M_d = \{S_4, S_6, S_8, S_{10}, S_{12}, S_{14}, S_{16}\}$.

Al encolar los estados pseudoequivalentes (fig. 4.20, a) se interrumpe la determinación del dispositivo automático. Para restablecer la determinación y equivalencia del funcionamiento del dispositivo automático introducamos el contador organizando ciclo de longitud variable. En la fig. 4.20, b se ofrece el programa de funcionamiento de contadores; en el dispositivo automático a examinar $k = 7$. El contador de longitud del ciclo $C_n LC$ realiza el ciclo exterior. El contador del número de ciclos $C_n C$ realiza el ciclo interior. En el bloque que incluye estos contadores (fig. 4.20, c) se calcula el indicio γ que restablece la determinación del dispositivo automático. En este mismo bloque se calcula el indicio β del fin de la operación (como señal de relleno del contador $C_n LC$ con la particularidad de que su estado inicial es igual a dos).

Después de restablecer la determinación del dispositivo automático encolamos los subgrafos bincidentes (fig. 4.21, a, b). Para restablecer la determinación del funcionamiento del dispositivo automático una vez concentrados los subgrafos bincidentes, introduzcamos otros dos contadores que tienen tres y dos estados, respectivamente.

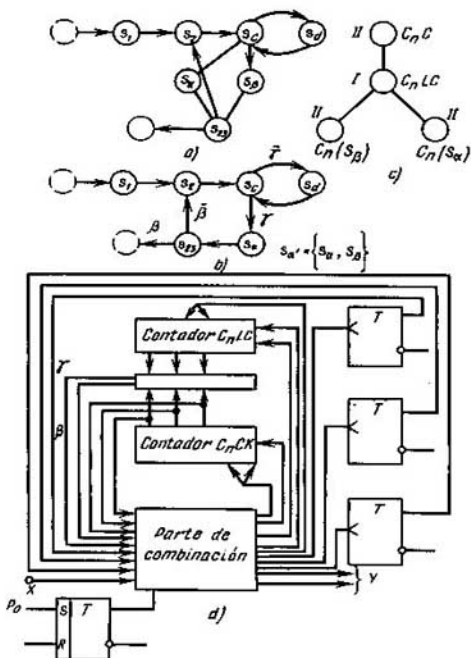


Fig. 4.21

Al restablecer la determinación del dispositivo automático introduciendo contadores, surge el problema de minimización del número de contadores que se reduce al problema de coloración del grafo de borrado G_b . A cada contador se le pone en correspondencia biunívoca un vértice del grafo G_b ; dos vértices del grafo G_b son adyacentes, si los subgrafos del funcionamiento de contadores correspondientes tienen al menos un vértice común (el subgrafo G_α es el subgrafo del funcionamiento del contador α , si para realizar transiciones correctas en este subgrafo es necesario saber el estado del contador α).

En el caso examinar a los contadores $C_n C$, $C_n LC$, $C_n(S_A)$ y $C_n(S_B)$ les corresponden subgrafos, cuyos portadores tienen, respectivamente, forma siguiente: $\{s_3, s_4\}$, $\{s_2, s_3, s_4, s_5, s_6, s_{23}\}$, $\{s_A\}$ y $\{s_B\}$. El grafo de borrado G_b para el caso considerado está representado en la fig. 4.21, c.

Es obvio, que dos «contadores», que corresponden a los vértices no adyacentes en el grafo de borrado, pueden ser examinados como un conta-

dor físico, ya que no existe ningún estado, en el cual sea imprescindible saber los estados de los dos contadores para que el dispositivo automático funciones correctamente. De este modo, la minimización del número de contadores se reduce a la coloración de los vértices del grafo de borrado.

En virtud de la coloración minimal de vértices del grafo de borrado (fig. 4.21, c), en este caso es suficiente tener dos contadores: C_nLC y C_nCK que se conecta en los momentos del funcionamiento de los contadores C_nC , $C_n(S_a)$ y $C_n(S_b)$ (fig. 4.21, d).

El empleo de los contadores en el circuito de realización de los dispositivos automáticos de mando posibilita utilizar nudos estándares del ordenador ejecutados en forma de microcircuitos integrados.

Como resultado de la búsqueda de la descomposición óptima de los dispositivos automáticos obtenemos algunas variantes equivalentes, entre los cuales escogemos la variante final que tiene valor minimal $p(G^M)$:

$$p(G^M) = \frac{1}{|U|} \sum_{k=1}^{|U|} \left(\frac{1}{l(l-1)} \sum_{i=1}^{l-1} \sum_{j=l+1}^l \frac{f_i - 2f_{ij} + f_j}{f_b} \right)_k, \quad (4.11)$$

donde $|U|$ es el número de palabras en el mografo, cada una de las cuales corresponde a una transición de autómata y se representa como XS^+S^-Y ; l es el número de letras (termas) que forman la palabra XS^+S^-Y ; por el signo exterior Σ se halla una expresión que es valor medio de la derivada $\partial G^M / \partial S$ calculada sobre los pares de letras que forman la palabra XS^+S^-Y . A esta variante la corresponde una realización estructural más sencilla.

§ 4.6. Codificación de los estados interiores

En la etapa de *codificación de los estados interiores*, una aplicación $XS^+ \rightarrow S^-Y$ obtenida en la etapa del diseño abstracto se transforma en una aplicación $XZ^+ \rightarrow Z^-Y$, donde S^+ , S^- son los identificadores de estados interiores considerados en los momentos del tiempo t y $t + \tau$, respectivamente; Z^+ , Z^- son códigos de estos estados, cuyos elementos son letras del alfabeto estructural que se sintetiza; son 0 y 1 para la lógica de Boole.

Se puede codificar estados interiores del dispositivo automático partiendo de las exigencias de reducir gastos para aparatos o bien de aumentar la fiabilidad del funcionamiento del dispositivo automático, o bien de satisfacer simultáneamente las dos exigencias.

Analicemos el método de codificación que satisface la primera exigencia. El número de procedimientos de codificar N vértices del grafo de transiciones $G = \langle V, U \rangle$ aumenta, mientras que aumenta $|V|$:

$$N = \frac{i(2^{\lceil \log_2 |V| \rceil} - 1)!}{i(2^{\lceil \log_2 |V| \rceil} - |V|)! i \lceil \log_2 |V| \rceil!}, \quad (4.12)$$

donde $[]$ es el número entero próximo. Cada procedimiento de codificación determina sus gastos de aparatos para realizar el dispositivo automático.

Hartmanis, Stearns propusieron el más interesante método *de codificación con empleo de particiones sustituyentes*. Este método se basa en disminuir la dependencia funcional de las funciones de excitación. Lamentablemente, cuando se codifican los estados del dispositivo automático de gran capacidad de memoria este método necesita mucho trabajo, lo que no permite utilizarlo para codificar estados interiores de los dispositivos automáticos de mando.

Examinemos el *método de frecuencias y matrices para codificar* los estados. Sea que como resultado de la síntesis abstracta fue construido un mografo que determina la aplicación $XS^+ \rightarrow S^- Y$. Después de la codificación obtenemos un mografo que prefija la aplicación $XZ^+ \rightarrow Z^- Y$, es decir, en el mografo inicial sustituimos los vértices correspondientes a los identificadores de los estados interiores S_i por subgrafos completos correspondientes a los códigos Z_i de los estados interiores. Este método de codificación puede realizarse empleando el siguiente algoritmo de sintetizar el árbol codificante.

1. Construimos una matriz bidimensional $Q = [q_{ij}]$, a cada fila de la cual le corresponde biunívocamente una microoperación o un valor del canal de entrada (elementos del vector Y o X), a cada columna, un estado interior

$$q_{ij} = \begin{cases} 1, & \text{si la terma primaria correspondiente a la } i\text{-ésima fila} \\ & \text{que se incluye en cada par de vectores } (X, Y)(X \rightarrow Y) \\ & \text{que ponderan los arcos salientes del vértice } j; \\ 0 & \text{en caso contrario.} \end{cases}$$

Arbitrariamente a cada estado interior (una columna de la matriz) le ponemos en correspondencia biunívoca un vértice maximal del árbol codificante que se sintetiza. Ponemos $Q = \tilde{Q}$.

2. Por la matriz \tilde{Q} encontramos la matriz de frecuencia de las relaciones

$$\tilde{F} = \tilde{Q}^T \tilde{Q}.$$

3. Calculamos valores de la derivada del modelo prefijado por la matriz \tilde{Q} .

4. Escogemos un par de estados interiores con el valor calculado minimal de la derivada.

5. Eliminamos el par escogido de estados y en el árbol codificante le ponemos en correspondencia un vértice, origen de los arcos, cuyos extremos son los estados escogidos. Al vértice construido le ponemos en correspondencia biunívoca una columna de la matriz Q_c que se construye. Esta columna es igual al producto vectorial de las columnas de la matriz \tilde{Q} que corresponden a los vértices escogidos.

6. Comprobamos, si se quedan los estados interiores no considerados, para los cuales se calcularon valores de las derivadas. Si «sí», pasamos al punto 4, en caso contrario, al punto 7.

7. Comprobamos, si está formada la matriz Q_c . Si «sí», consideramos $Q_c = \hat{Q}$ y pasamos al punto 2, en caso contrario, al punto 8.

8. Ponemos 0 y 1 en correspondencia a cada dos arcos que salen de un vértice del árbol construido partiendo de los arcos salientes de la raíz del árbol.

El camino que une la raíz del árbol y un elemento maximal está ponderada por el código que se pone en correspondencia al estado interior que corresponde a este elemento maximal del árbol.

9. Fin.

Ilustremos el algoritmo propuesto en el siguiente ejemplo.

Ejemplo 4.2. Sea que después de la síntesis abstracta fue obtenido el grafo de transiciones G (fig. 4.22, a). Empleando el algoritmo propuesto codifiquemos estados interiores del dispositivo automático. Proponemos construir individualmente la matriz Q correspondiente al grafo dado. La matriz de frecuencia de relaciones correspondiente a la matriz Q tiene siguiente forma:

$$F = \begin{array}{c} \begin{array}{|c|c|c|c|c|c|} \hline S_1 & S_2 & S_3 & S_4 & S_5 & S_6 \\ \hline 2 & 0 & 2 & 1 & 0 & 0 \\ \hline 0 & 3 & 0 & 1 & 3 & 1 \\ \hline 2 & 0 & 5 & 2 & 1 & 1 \\ \hline 1 & 1 & 2 & 4 & 1 & 3 \\ \hline 0 & 3 & 1 & 1 & 5 & 1 \\ \hline 0 & 1 & 1 & 3 & 1 & 3 \\ \hline \end{array} \begin{array}{l} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{array} \end{array}$$

Calculamos valores de la derivada para cada par de estados:

$$\frac{\partial G^M}{\partial S}(S_1, S_3) = \frac{f_{11} - 2f_{13} + f_{33}}{f_{13}} = \frac{2 - 2 \cdot 2 + 5}{2} = 1,5;$$

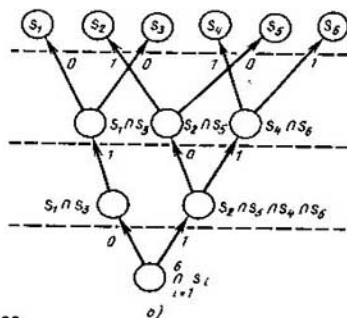
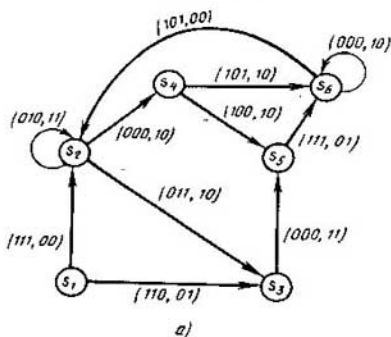


Fig. 4.22

$$\frac{\partial G^M}{\partial S}(S_1, S_4) = \frac{f_{11} - 2f_{14} + f_{44}}{f_{14}} = \frac{2 - 2 \cdot 1 + 4}{1} = 4;$$

$$\frac{\partial G^M}{\partial S}(S_2, S_4) = 5; \quad \frac{\partial G^M}{\partial S}(S_3, S_6) = 6; \quad \frac{\partial G^M}{\partial S}(S_2, S_5) = 0,67;$$

$$\frac{\partial G^M}{\partial S}(S_4, S_5) = 7; \quad \frac{\partial G^M}{\partial S}(S_2, S_6) = 4; \quad \frac{\partial G^M}{\partial S}(S_4, S_6) = 0,33;$$

$$\frac{\partial G^M}{\partial S}(S_3, S_4) = 2,5; \quad \frac{\partial G^M}{\partial S}(S_5, S_6) = 6; \quad \frac{\partial G^M}{\partial S}(S_3, S_5) = 8.$$

Los demás valores de la derivada son iguales a ∞ . La derivada $\frac{\partial G^M}{\partial S}(S_i, S_j)$ tiene valor minimal sobre el par (S_4, S_6) . A este par de estados le ponemos en correspondencia la «intersección» de los vértices correspondientes del árbol en construcción. Según el algoritmo juntamos en pares de estados S_2, S_5 y S_1, S_3 . Como resultado construimos el siguiente nivel del árbol.

La matriz Q_c correspondiente a los vértices del nivel construido tiene forma:

$$Q_c = \begin{array}{c} \begin{array}{ccc} S_1 \cap S_3 & S_2 \cap S_5 & S_4 \cap S_6 \end{array} \\ \left\| \begin{array}{ccc} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{array} \right\| \begin{array}{l} x_1 \\ \bar{x}_1 \\ x_2 \\ \bar{x}_2 \\ x_3 \\ \bar{x}_3 \\ y_1 \\ \bar{y}_1 \\ y_2 \\ \bar{y}_2 \end{array} \end{array}$$

La matriz de frecuencia de relaciones correspondiente a la matriz Q_c tiene siguiente forma:

$$F_c = \left\| \begin{array}{ccc} 2 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 1 & 3 \end{array} \right\|$$

La derivada del modelo $\Psi(Q_c)$ sobre los pares de estados es

$$\frac{\partial G}{\partial S}(S_2 \cap S_5, S_4 \cap S_6) = \frac{3 - 2 \cdot 1 + 3}{1} = 6.$$

Los demás valores de la derivada son iguales a ∞ . El árbol codificante buscado está representado en la fig. 4.22, b. Según el árbol codificante construido tenemos siguientes códigos de los estados interiores del dispositivo automático: S_1 —010, S_2 —100, S_3 —011, S_4 —110, S_5 —101, S_6 —111.

Examinemos la codificación de estados interiores del dispositivo automático partiendo de la satisfacción de las exigencias de fiabilidad. El funcionamiento del dispositivo automático puede interrumpirse debido al retardo desigual en el esquema real que realiza el dispositivo automático en

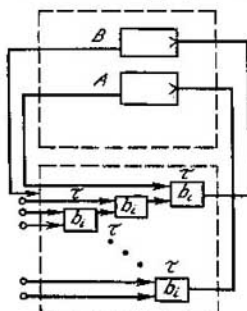


Fig. 4.23

virtud del efecto de carreras, cuya esencia puede ser ilustrada en el siguiente ejemplo (fig. 4.23). Sea que en el momento examinado dos triggers A y B se conmutan (la primera condición), con la particularidad de que la función de excitación φ de uno de éstos (por ejemplo, B) contiene en calidad de una variable el valor del trigger A , $\varphi_B = \varphi(\dots, z_A^+, \dots)$ (la segunda condición) y el tiempo de retardo en los esquemas de excitación de los triggers A y B se determina por la siguiente desigualdad: $\Delta_{BA} > t_A$ (Δ_{BA} es el tiempo, durante el cual el valor «viejo» del trigger A existe en el esquema de excitación del trigger B ; t_A es el tiempo de retardo de la señal de excitación del trigger A) (la tercera condición).

Cumpliendo estas tres condiciones el valor del trigger no será calculado correctamente (ya que para el cálculo correcto es necesario que el trigger A conserve su valor «viejo» por lo menos durante 2τ (después de empezar la transición del dispositivo automático), pero el trigger A «renueva» su valor dentro de τ). Esto es el llamado *efecto de carreras*. Al haber dicho efecto el dispositivo automático no pasa al estado indicado para la transición dada, lo que interrumpe la correspondencia de autómeta.

Se puede eliminar las carreras interrumpiendo una de las tres condiciones de éstas. Para interrumpir la tercera condición, o sea, para cumplir la desigualdad $\Delta_{BA} < t_A$ es necesario tener esquemas de excitación de los triggers A y B para determinar Δ_{BA} y t_A . Se puede obtenerlos sólo en la etapa de la síntesis estructural, para cuya realización son necesarios los resultados de codificación de los estados interiores. Por consiguiente, eliminando lógicamente las carreras es imposible interrumpir la tercera condición. Técnicamente se puede realizarlo introduciendo la segunda cascada de elementos de reacción o bien por medio del cumplimiento simultáneo del esquema del dispositivo automático.

La segunda condición de carreras ($\varphi_B = \varphi(\dots, z_A^+, \dots)$) puede ser interrumpida por medio de desatar funcionalmente los elementos de memoria, por ejemplo, empleando particiones sustituyentes.

La primera condición (conmutación de dos, y más elementos de memoria durante una transición del dispositivo automático) se interrumpe, si durante cualquier transición del dispositivo automático se conmuta sólo un elemento de memoria. Esto significa que a cada estado interior del dispositivo automático se le pone en correspondencia un código tal que si existe la transición del estado S_i al estado S_j ($S_i \neq S_j$), sus códigos correspondientes se diferencian solamente en un orden. En otras palabras, durante la realización del dispositivo automático las carreras ausentan, si se puede colocar el grafo correspondiente de transiciones en un hipercubo n -dimensional de tal modo que las transiciones del estado S_i al estado S_j ($S_i \neq S_j$) se realicen sólo por las aristas del hipercubo y los códigos de vértices correspondientes del hipercubo se hayan puesto en correspondencia a los vértices del grafo de transiciones. Esta codificación se denomina *vecina*.

Digamos que para un grafo de transiciones es posible la codificación vecina por los códigos de longitud n , si éste es encajable en un cubo n -dimensional. Expongamos la condición de encajabilidad del grafo de transiciones en el hipercubo.

Del análisis del hipercubo se desprende que si el número cromático $h(G)$ de un grafo de transiciones G (sin contar los arcos que son los lazos) es más que dos, para este grafo de transiciones no existe la codificación vecina. Por lo tanto, para que el grafo de transiciones G tenga codificación vecina hace falta que no contenga ciclos de longitud impar. Si los contiene, es necesario eliminar todos los ciclos de longitud impar introduciendo estados interiores complementarios, en los cuales no se realiza la aplicación $X \rightarrow Y$. Estos estados se denominan estados interiores *inestables*.

Examinemos un algoritmo exacto de la codificación vecina basado en contar figuras prohibidas que caracterizan el carácter cubicable del grafo codificado.

Ejemplo 4.3. Sea dado un grafo de transiciones del dispositivo automático (fig. 4.24) que se representa sin los lazos, arcos múltiples y la orientación (ya que todo esto no es sustancial para resolver el problema de la codificación vecina). En correspondencia con las condiciones deducidas anteriormente para que sea posible la codificación vecina, es necesario y suficiente que el grafo no contenga subgrafos críticos prohibidos. Al diseñar el dispositivo automático es necesario transformar estos grafos de tal modo que todos ellos se hagan cubi-

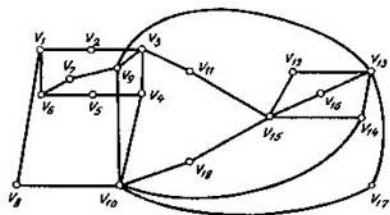


Fig. 4.24

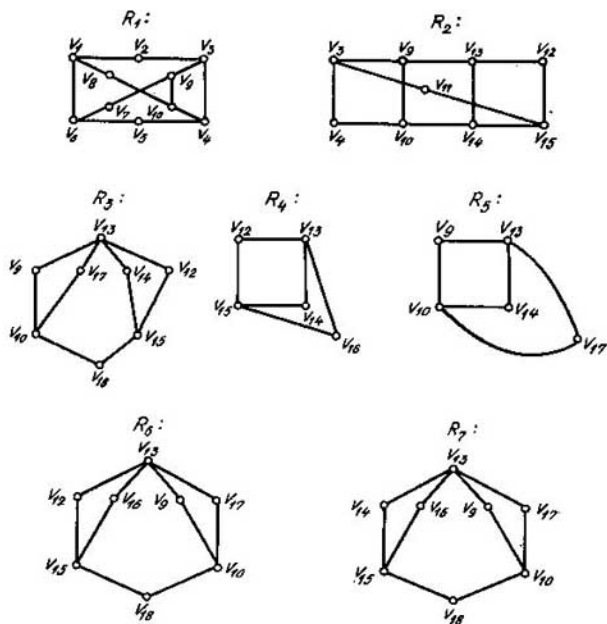


Fig. 4.25

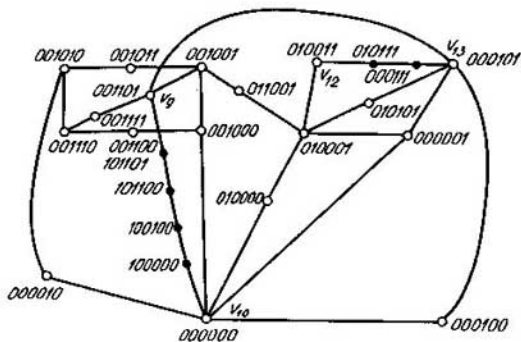


Fig. 4.26

cables. La transformación dada no interrumpe el funcionamiento del dispositivo automático sólo en el caso, cuando cada grafo nuevo es homeomorfo al anterior. De hecho la transformación se reduce a añadir los vértices nuevos a unas aristas del cada grafo crítico inicial que es el subgrafo del grafo dado de transiciones. De este modo introducimos los estados interiores inestables del dispositivo automático. En el diseño óptimo el número de estados inestables introducidos debe ser mínimo.

La tabla bidimensional (tabla 4.7) de distribución de aristas por las figuras prohibidas (fig. 4.25) se cubre por dos filas $\{l_{13}, l_{21}\}$. Añadiendo entre los vértices v_9, v_{10} y v_{12}, v_{13} los que corresponden a los estados inestables, encajamos el grafo obtenido en el espacio booleano (fig. 4.26).

Tabla 4.7

	R_1	R_2	R_3	R_4	R_5	R_6	R_7
$l_1 = \{v_1, v_2\}$	1	0	0	0	0	0	0
$l_2 = \{v_2, v_3\}$	1	0	0	0	0	0	0
$l_3 = \{v_3, v_4\}$	1	1	0	0	0	0	0
$l_4 = \{v_4, v_5\}$	1	0	0	0	0	0	0
$l_5 = \{v_5, v_6\}$	1	0	0	0	0	0	0
$l_6 = \{v_1, v_6\}$	1	0	0	0	0	0	0
$l_7 = \{v_6, v_7\}$	1	0	0	0	0	0	0
$l_8 = \{v_7, v_9\}$	1	0	0	0	0	0	0
$l_9 = \{v_7, v_9\}$	1	0	0	0	0	0	0
$l_{10} = \{v_3, v_9\}$	1	1	0	0	0	0	0
$l_{11} = \{v_1, v_8\}$	1	0	0	0	0	0	0
$l_{12} = \{v_8, v_{10}\}$	1	0	0	0	0	0	0
$l_{13} = \{v_4, v_{10}\}$	1	1	0	0	0	0	0
$l_{14} = \{v_9, v_{10}\}$	1	1	1	0	1	1	1
$l_{15} = \{v_9, v_{13}\}$	0	1	1	0	1	1	1
$l_{16} = \{v_{10}, v_{18}\}$	0	0	1	0	0	1	1
$l_{17} = \{v_{10}, v_{14}\}$	0	1	0	0	1	0	0
$l_{18} = \{v_{10}, v_{17}\}$	0	0	1	0	1	1	1
$l_{19} = \{v_{15}, v_{18}\}$	0	0	1	0	0	1	1
$l_{20} = \{v_3, v_{11}\}$	0	1	0	0	0	0	0
$l_{21} = \{v_{11}, v_{15}\}$	0	1	0	0	0	0	0
$l_{22} = \{v_{12}, v_{13}\}$	0	1	1	1	0	1	0
$l_{23} = \{v_{12}, v_{15}\}$	0	1	1	1	0	1	0
$l_{24} = \{v_{13}, v_{14}\}$	0	1	1	1	1	0	1
$l_{25} = \{v_{14}, v_{15}\}$	0	1	1	1	0	0	1
$l_{26} = \{v_{15}, v_{16}\}$	0	0	0	1	0	1	1
$l_{27} = \{v_{13}, v_{10}\}$	0	0	0	1	0	1	1
$l_{28} = \{v_{13}, v_{17}\}$	0	0	1	0	1	1	1

Al introducir los vértices complementarios en el grafo no se interrumpen las figuras permitidas (ciclos de longitud par).

§ 4.7. Diseño estructural de los dispositivos automáticos

Examinemos el diseño del esquema lógico como síntesis del grafo booleano correspondiente. Partamos todo el conjunto de bases en dos clases: *topológicas* y *funcionales*.

En las bases topológicas para la síntesis se utilizan funciones, cuya completitud puede lograrse por medio de la correspondiente simulación física de variables binarias y por la determinada agrupación de los elementos prefijados. Todos los elementos de conmutación, llaves de corriente, criotrones, interruptores de alumbrado, espacitores, deplitores, unitrones y otros elementos de válvula, que en la técnica de cálculos se utilizan como elementos de acción rápida, forman bases de este tipo.

En las bases funcionales, los elementos, independientemente de su agrupación y la simulación física de variables binarias, realizan las funciones que forman un sistema completo de funciones booleanas. Por ejemplo, se refieren a estas bases los elementos que realizan las funciones de Sheffer, de Webb, de implicación, etc.

Partamos, a su vez, el conjunto de bases topológicas en cuatro clases. La primera y la segunda clase las forman elementos que dejan pasar la señal de información en un sentido cuando el valor de la señal de control coincide con la letra que pondera este elemento. Pero, en los elementos de la primera clase, la señal pasada se difunde por los canales de conductividad unilateral en los elementos de la segunda clase, por los canales de conductividad bilateral.

La primera clase comprende elementos de válvula-diodo, en caso del procesamiento óptico de la información con el empleo de la técnica de fibras, es decir, los interruptores de alumbrado; la segunda clase comprende elementos de válvula (por ejemplo, espacistor-triodo).

En las entradas de un esquema compuesto de elementos que se examinan y en las de un esquema construido de los elementos de cualquier base topológica se usa la representación parafísica de la información binaria.

La tercera y la cuarta clase las forman elementos que dejan pasar la señal de información en los dos sentidos cuando coinciden el valor de la señal de control y la letra que pondera este elemento.

En los elementos de la tercera clase la señal pasada se propaga por los canales de conductividad unilateral y en los elementos de la cuarta clase, por los de conductividad bilateral.

Entre los elementos de la tercera clase figuran, por ejemplo, los tuneltrones y los elementos de contacto-diodo; entre los de la cuarta, elementos conmutadores de conductividad bilateral (criotrones y contactos).

De este modo, todo el conjunto de bases topológicas está dividido en cuatro clases según dos características: conductividad de elemento y conductancia del canal de conexión.

Como resultado de la optimización abstracta y la codificación de estos interiores del dispositivo automático, un sistema de funciones de excitación y de salida prefija el operador de autómatas. La realización en esquemas, óptima por su complejidad, de estas funciones es un proceso de trabajo denso, lo que está condicionado por el carácter combinatorio del

problema. Uno de los caminos eficaces para disminuir la densidad de trabajo de la realización en esquemas del operador de autómatas es reducir dimensiones de los problemas empleando la descomposición. Para las funciones que se obtienen como resultado de la descomposición, se puede construir los esquemas óptimos. La descomposición óptima está orientada a lograr la óptimidad global de la resolución del problema de realizar el operador de autómatas en esquema.

La descomposición de la función $f(X)$, $X = \{x_1, \dots, x_n\}$ es su representación en forma de superposición de varias funciones de menor dimensión: $f(X) = F(\varphi_1(X_1), \varphi_2(X_2)), \dots, \varphi_k(X_k)$, donde $X_i \subseteq X$. La función F se denomina *exterior*, las funciones $\varphi_1, \dots, \varphi_k$ se llaman *conjugadas*. El criterio de calidad de la descomposición es el mínimo del número de las funciones conjugadas.

La descomposición debe satisfacer las siguientes exigencias prácticas:

- 1) tener en cuenta las particularidades de las funciones que se encuentran en la práctica de la proyección;
- 2) aumentar la fiabilidad de los dispositivos automáticos en diseño a cuenta de bajar la probabilidad de surgimiento del riesgo.

La primera exigencia supone que se puede aplicar bien el método a las funciones débilmente definidas, que aparecen con la mayor frecuencia en la práctica, es decir, las funciones cuya potencia en un campo definido es mucho menor que la del campo indefinido. La segunda exigencia impone una restricción sobre la longitud de los caminos en la realización en esquema de la función, los caminos que se determinan por la profundidad, a que entran en la función los correspondientes argumentos. Por ejemplo, la probabilidad de que surja el riesgo para la función $f(X) = F(\varphi_1(\varphi_{11}(x_1, \dots), \varphi_{12}(\varphi_{121}(x_2, \dots), \varphi_{122}(x_3, \dots), \dots), \varphi_{13}(\dots), \varphi_2(\varphi_{21}(x_4, \dots), \dots, x_5), \dots, x_6))$ es proporcional a la suma de módulos de las diferencias de las profundidades a que los argumentos entran en las subfunciones. Para la función dada, la profundidad de entrada de la variable x_6 es igual a cero, de la variable x_3 , a tres. La validez de esta afirmación se basa en las siguientes suposiciones: los sucesos en los que pueda surgir el riesgo en las funciones conjugadas son independientes; la probabilidad de que surja el riesgo en la función es proporcional al módulo de diferencia de las longitudes de los caminos por las que pasan las señales, es decir, en la representación de la función $F(\varphi_1(x_1, \dots), x_2)$ la probabilidad de que surja el riesgo es dos veces menor que en la representación de la función $F(\varphi_1(\varphi_2(x_3, \dots), x_1), x_2)$. Por lo tanto, la probabilidad de que surja el riesgo para la función $f(X)$ es proporcional al desbalance de longitudes de los caminos d_i en el esquema que realiza $f(X)$, $d_i = \sum |l_{med} - l_i|$, donde l_{med} es la longitud media del camino, l_i es la longitud del i -ésimo camino. El desbalance d_i puede valorarse por el desbalance de las profundidades de entradas d_v , que se obtiene de d_i sustituyendo l_i por ν_i , l_{med} por ν_{med} , donde ν_{med} es la profundidad media de entrada

de una variable en una función, ν_i es la profundidad de entrada de la variable x_i en la función.

La segunda de las exigencias anteriormente aducidas se cumple automáticamente en la descomposición $f(X)$ tipo

$$F(\varphi_1(X_A), \dots, \varphi_p(X_A), \xi_1(X_B), \dots, \xi_s(X_B)), \quad (4.13)$$

donde $X_A \cup X_B = X$, $X_A \cap X_B = \emptyset$.

Examinemos el método de la descomposición estructural y funcional tipo (4.13) orientado a las funciones débilmente definidas. Se basa en la representación de la función por un K -grafo. Llámase K -grafo un grafo de dos partes $G^K = \langle V^A, V^B, U_1, U_0 \rangle$, en el cual los conjuntos V^A y V^B determinan el conjunto de vértices G^K , mientras que U_1 y U_0 son los conjuntos que representan aristas de dos tipos que unen los vértices de V^A y V^B .

De la partición del conjunto X en X_A y X_B se desprende la partición de cada juego m_i del campo unitario de definición de la función $f(X)$ en dos subjuegos m_i^A y m_i^B que comprenden las variables entrantes en X_A y X_B , respectivamente. Los campos unitario y nulo se determinan mediante las colecciones de los pares de subjuegos (m_i^A, m_i^B) . De tal modo, la función $f(X)$ con la partición prefijada de X en X_A y X_B se representa por el K -grafo $G^K = \langle V^A, V^B, U_1, U_0 \rangle$, donde V^A es el conjunto de subjuegos m_i^A , V^B es el conjunto de subjuegos m_i^B , U_1, U_0 son las relaciones binarias que determinan los campos unitario y nulo de la función.

Examinemos la función booleana débilmente definida $f(X)$, $X = \{x_1, x_2, x_3, x_4\}$. La función $f(X)$ toma el valor 1 sobre el conjunto de juegos $M_1 = \{1010, 0111, 1101, 1100\}$; toma el valor 0 sobre el conjunto de juegos $M_0 = \{0001, 1000, 1011\}$. Sean $X_A = \{x_1, x_2\}$, $X_B = \{x_3, x_4\}$. El K -grafo G^K que representa $f(X)$ con la partición dada X aparece reflejado en la fig. 4.27, *a*. Las aristas tachadas forman el conjunto U_0 , las no tachadas, el conjunto U_1 .

Definamos la operación de encolamiento de los vértices en el K -grafo. Dos vértices de una clase v_1 y v_2 pueden encolarse, si no existe el vértice w de otra clase ligada con un vértice por la arista y con otro, por la arista tachada. Como resultado del encolamiento los vértices se sustituyen por uno y sus entornos se agrupan. Por ejemplo, después de encolar los vértices correspondientes a los juegos $\bar{x}_1 x_2$ y $\bar{x}_1 \bar{x}_2$, el K -grafo tiene forma dada en la fig. 4.27, *b*. Notemos que el encolamiento de los vértices de una clase cambia las condiciones del encolamiento de los de otra clase. Encolaremos los vértices del K -grafo hasta que sea posible. El K -grafo resultante está representado en la fig. 4.27, *c*.

Teorema 4.2. Una función booleana $f(X) = f(x_1, x_2, \dots, x_n)$ es representable en forma $f(X) = F(\varphi_1(X_A), \dots, \varphi_p(X_A), \xi_1(X_B), \dots, \xi_s(X_B))$, $X_A \cap X_B = \emptyset$, $X_A \cup X_B = X$, $p < |X_A|$, $s < |X_B|$ si, y sólo si, existe una sucesión de operaciones de encolamiento que reduce el K -grafo $G^K = \langle V^A,$

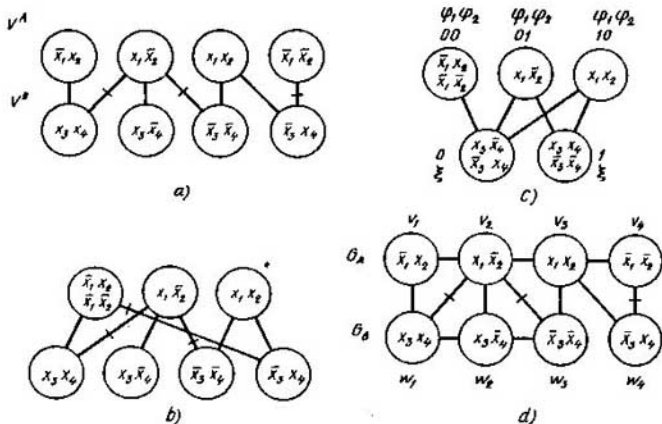


Fig. 4.27

V^B, U_1, U_0 , que representa la función, al K -grafo $G^K = \langle V_1^A, V_1^B, U_1, U_0 \rangle$ que tiene $|V_1^A| \leq 2^p, |V_1^B| \leq 2^s$.

El criterio de calidad de la descomposición es el mínimo del número de funciones φ_i y ξ_i que se alcanza minimizando el número de vértices en las clases del K -grafo después de encolar. El proceso de encolar los vértices es equivalente a la coloración de los grafos de incompatibilidad $G_A = \langle V^A, U^A \rangle$ y $G_B = \langle V^B, U^B \rangle$. Los portadores de estos grafos son las clases de los vértices V^A y V^B . Dos vértices v_1 y v_2 de un grafo de incompatibilidad son adyacentes, si uno de ellos se une con un vértice de otra clase mediante una arista y el otro, mediante una arista tachada. Los grafos de incompatibilidad para el K -grafo representado en la fig. 4.27, *a* se ofrecen en la fig. 4.27, *d*.

Si fijamos el mismo color de dos vértices del grafo de incompatibilidad, esto es equivalente a encolarlos y por esta razón, en caso general, lleva a introducir nuevas aristas en otro grafo de incompatibilidad. Si los vértices v_1 y v_2 de un grafo G_i ($i = A, B$) pasan a ser de un mismo color, en G_j ($j = A, B, j \neq i$) se introducen las aristas (w_1, w_2) tales que en la configuración de G_A y G_B los v_1 y w_1 se unen mediante una arista, v_2 y w_2 , mediante una arista tachada. De aquí se deduce que en caso general la coloración de configuraciones de los grafos de incompatibilidad no se determina por su coloración independiente. El proceso de encolamiento es equivalente a la coloración conexa de los grafos de incompatibilidad. Como *coloración conexa* de la configuración de grafos G_A y G_B se comprende una partición de los vértices de los grafos en clases (y atribución de un mismo color a

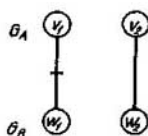


Fig. 4.28

los vértices de cada clase) tal que en cada clase, los vértices no son adyacentes dos a dos y, además, en G_A , los vértices de cualquier clase no pueden unirse simultáneamente mediante las aristas y aristas tachadas con los vértices de una clase G_B . Por lo tanto, la descomposición de una función booleana débilmente definida se reduce a la coloración conexas de los grafos de incompatibilidad.

Aunque en caso general la coloración independiente de cada uno de los grafos no determina la coloración conexas es muy interesante encontrar las condiciones, en los cuales la coloración independiente de los grafos de incompatibilidad da la coloración conexas, puesto que en este caso el problema de descomposición se simplifica considerablemente.

Teorema 4.3. *La coloración independiente de los grafos G_A y G_B determina su coloración conexas si, y sólo si, la configuración de G_A y G_B no contiene la configuración de G_{AB} en calidad de subgrafo (fig. 4.28).*

□ *Necesidad.* Sea que la coloración conexas de los grafos G_A y G_B se reduce a su coloración independiente. Esto significa que la coloración conexas de cualquier subgrafo de la configuración se determina también por la coloración independiente de los grafos que la integran. La coloración conexas de la configuración de G_{AB} no se determina por la configuración independiente, por eso la configuración de G_A y G_B no puede contener G_{AB} en calidad de subgrafo.

Suficiencia. Sea que la coloración conexas de los grafos G_A , G_B no se determina por su coloración independiente. Sea también que fijando los colores iguales de vértices no adyacentes v_1, v_2 del grafo G_A llegamos a introducir la arista (w_1, w_2) en el grafo G_B . Esto es posible sólo en el caso cuando uno de los vértices v_1, v_2 se une con uno de los vértices w_1, w_2 mediante la arista y el otro se vincula con el restante mediante la arista tachada. Otras aristas entre v_1, v_2 y w_1, w_2 no pueden existir ya que en caso contrario serían adyacentes a v_1, v_2 o w_1, w_2 . Pero la configuración descrita de v_1, v_2 y w_1, w_2 es la configuración de G_{AB} con exactitud de hasta la redesignación. Por lo tanto, si G_{AB} no se presenta en la configuración de G_A, G_B como un subgrafo generado, la coloración independiente de G_A y G_B determina su coloración conexas. ■

Si sabemos la configuración de G_{AB} , podemos proponer el aparato constructivo para reducir la coloración conexas a la independiente. Para cada subgrafo de la configuración, isomorfo a G_{AB} , con los vértices v_1, v_2 ,

w_1, w_1 introduzcamos los vértices v'_1, v'_2, w'_1, w'_2 . Unamos v'_1 con los vértices de $\Gamma(v_1)$; v'_2 , con los de $\Gamma(v_2)$; w'_1 , con los de $\Gamma(w_1)$; w'_2 , con los de $\Gamma(w_2)$. Introduzcamos también las aristas (v'_1, v'_2) y (w'_1, w'_2) . La configuración obtenida la denominaremos *extensión* y denotaremos G_A^*, G_B^* . Hallemos la coloración independiente de G_A^* y G_B^* . Ahora realicemos la operación de reducción: para toda configuración de G_{AB} eliminemos los vértices v_1, v_2 y w_1, w_2 , si el vértice v_1 es del mismo color que v_2 y w_1 es del mismo color que w_2 . Pero, si por lo menos un par de v_1, v_2 y w_1, w_2 no es concoloreado, eliminemos v'_1, v'_2, w'_1, w'_2 . Resulta la coloración conexas de la configuración de G_A, G_B .

Ejemplo 4.4. La extensión de G_A, G_B para la configuración de G_A, G_B dada en la fig. 4.27, *a* y la coloración independiente de G_A, G_B están representadas en la fig. 4.29. Como resultado de la extensión de G_A y G_B obtenemos el encolamiento del K -grafo ofrecido en la fig. 4.27, *c*.

Codifiquemos los vértices de cada clase. La longitud del código de ésta es igual a $\lceil \log_2 s \rceil$, donde s es el número de vértices en la clase. Cada código determina un juego definido sobre nuevas variables, obtenidas como resultado de la codificación: φ_i , para una clase ξ_i , para otra. De este modo, el K -grafo determina una nueva función F , exterior en la descomposición (4.13). La descomposición de la función considerada se determina del modo siguiente:

$$f(x_1, x_2, x_3, x_4) = F(\varphi_1(x_1, x_2), \varphi_2(x_1, x_2), \xi(x_3, x_4)).$$

La función $F(\varphi_1, \varphi_2, \xi)$ toma el valor unitario sobre el conjunto de juegos $M_1 = \{\bar{\varphi}_1\bar{\varphi}_2\bar{\xi}, \bar{\varphi}_1\varphi_2\bar{\xi}, \bar{\varphi}_1\varphi_2\xi\}$ y el valor nulo sobre el conjunto de juegos $M_0 = \{\bar{\varphi}_1\varphi_2\xi, \bar{\varphi}_1\bar{\varphi}_2\xi\}$. Con ello, $\varphi_1\varphi_2$ y ξ son las funciones débilmente definidas. Para ellas escribamos los conjuntos de juegos unitarios y nulos:

$$M_1(\varphi_1) = \{x_1x_2\}, M_0(\varphi_1) = \{\bar{x}_1x_2, \bar{x}_1\bar{x}_2, x_1\bar{x}_2\};$$

$$M_1(\varphi_2) = \{x_1\bar{x}_2\}, M_0(\varphi_2) = \{\bar{x}_1x_2, \bar{x}_1\bar{x}_2, x_1x_2\};$$

$$M_1(\xi) = \{x_3x_4, \bar{x}_3\bar{x}_4\}, M_0(\xi) = \{x_3\bar{x}_4, \bar{x}_3x_4\};$$

La signatura de las álgebras determina *leyes de composición*, cuyas propiedades se prefijan por las identidades de las álgebras. Las leyes de compo-

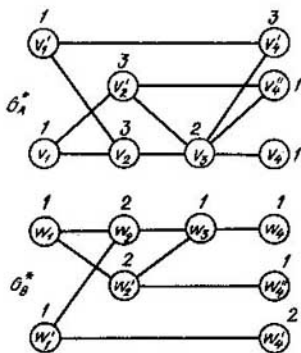


Fig. 4.29

sición permiten determinar el funcionamiento de un todo único por el funcionamiento de sus partes. Durante la síntesis de los sistemas el funcionamiento de un todo único está prefijado y es necesario determinar el funcionamiento de sus partes, o sea, construir una estructura que realiza el funcionamiento prefijado. En la síntesis no tienen interés las leyes de composición, sino las *de descomposición*.

Examinemos un modelo

$$\Psi = \langle M, P'_1, P'_2, \dots, P'_n \rangle,$$

donde

$$P'_i(m_1, m_2, \dots, m_k) = \begin{cases} 1, & \text{cuando } m_k = f_i(m_1, m_2, \dots, m_{k-1}), \\ 0 & \text{en caso contrario.} \end{cases}$$

Análogamente al concepto del *álgebra*

$$A \langle M, f_1, f_2, \dots, f_n \rangle,$$

introduzcamos el concepto de la *coálgebra*

$$K = \langle M, x'_1, x'_2, \dots, x'_n \rangle,$$

donde M es el portador de la coálgebra, x'_1, x'_2, \dots, x'_n es su *signatura*. El procedimiento de determinar por m_k un conjunto $\{m_1, m_2, \dots, m_{k-1}\}$, tal que $m_k = f_i(m_1, m_2, \dots, m_{k-1})$, se denomina *co-operación* $x'_i(m_k)$,

$$x'_i(m_k) = \{m_1, m_2, \dots, m_{k-1}\}.$$

Es obvio que, si el resultado de operaciones f_i es estrictamente unívoco, el de la co-operación x'_i no lo es. El álgebra determina las leyes de composición, la coálgebra, las de descomposición.

Así como existen, por ejemplo, las álgebras de Boole, las de Webb, el álgebra implicativa, el álgebra de Zhegalkin es lógico, desarrollando la teoría de las coálgebras, esperar que aparezca la coálgebra de Boole, la de Webb, la implicativa, la de Zhegalkin.

En el presente está elaborada solamente la coálgebra de grafos que es isomorfa a la coálgebra de Boole, cuyo portador está prefijado por los diagramas de Hasse o por los grafos estructurales.

En un grafo estructural, cada vértice está ponderado por una *terma primaria* x_i^0 y el camino corresponde biunívocamente a un intervalo maximal (implicante simple) de una función booleana $f(x_1, x_2, \dots, x_n)$.

Llámase *coálgebra de los grafos* una colección de la forma

$$K = \langle M, x^{\vee}, x^{-} \rangle,$$

donde el portador M es el conjunto de todos los grafos estructurales posibles, la *signatura* es la *co-operación de disyunción* x^{\vee} y la *co-operación de negación* x^{-} de los grafos estructurales. A continuación, estas dos co-operaciones se denominarán *operación de descomposición* x^{\vee} y *operación de inversión* x^{-} de los grafos, respectivamente.

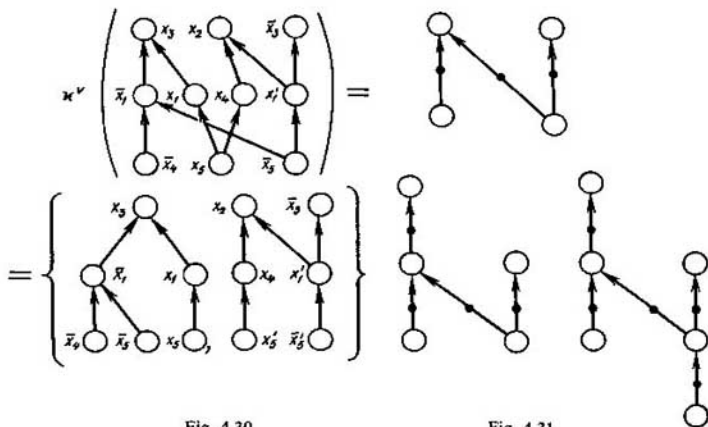


Fig. 4.30

Fig. 4.31

Operación de descomposición x^- de los grafos. Cumpliendo esta operación se indican las direcciones, en las cuales se transcurre la descomposición del grafo H . La dirección se prefija mediante los vértices que son los elementos maximales de los subgrafos del grafo H .

La operación de descomposición $x^-(H)$ del grafo H en direcciones V_i ($i = 1, \dots, k$) es la formación respectiva de los subgrafos H_i ($i = 1, \dots, k$) compuestos de todos los vértices $\{v_j/j = 1, \dots, l\}$ y los arcos que los unen, para los cuales en el grafo H existe un camino que une $v_a \in \{v_j/j = 1, \dots, l\}$ y $v_b \in V_i$.

Por ejemplo, el resultado de la operación de descomposición del grafo estructural H en las direcciones $\{x_3\}$ y $\{x_2, \bar{x}_3\}$ está efectuado en la fig. 4.30. El signo (') marca las letras repetidas, para que éstas identifiquen los vértices del grafo.

La operación de inversión de x^- grafos. El grafo estructural representado en forma de la superposición de estructuras tipos π y σ se denomina grafo tipo $\pi\sigma$.

La operación de inversión $x^-(H)$ de un grafo estructural H es la reducción del grafo dado a un grafo tipo $\pi\sigma$ utilizando su descomposición, la sustitución de cada subgrafo tipo σ por un subgrafo tipo π y de cada subgrafo tipo π por un subgrafo tipo σ ; con ello, cada peso x_i^{σ} del grafo inicial se sustituye por el peso $x_i^{(\sigma+1) \bmod 2}$ en los vértices correspondientes del grafo obtenido.

Para la propiedad de los grafos estructurales de ser grafos tipo $\pi\sigma$ son prohibidas las figuras H_Γ (fig. 4.31). Por consiguiente, la desintegración mi-

Al sintetizar los esquemas lógicos en una base funcional B , es necesario determinar resultados de los co-operaciones tipo

$$\begin{aligned} & x^{f_b}(b_i \in B), \text{ es decir,} \\ & x^{f_b}(H) = \{H_1, H_2, \dots, H_{k_{ent}}\}, \end{aligned} \quad (4.14)$$

donde k_{ent} es el coeficiente de entrada del elemento b_i .

Se puede obtener el resultado de la co-operación x^{f_b} empleando respectivamente una serie de co-operaciones tipo x^\vee y x^- en el orden inverso del que siguen las operaciones \vee y $-$ en la descomposición de la operación f_b en \vee y $-$. Por ejemplo, determinamos los resultados de la co-operación de Sheffer $x'(H)$ empleando la coálgebra de los grafos del modo siguiente.

Sea necesario hallar el resultado $\{H_1, H_2\}$ de la co-operación $x'(H)$, es decir,

$$x'(H) = \{H_1, H_2\}. \quad (4.15)$$

Los grafos H_1 y H_2 son los resultados de $x'(H)$ si, y sólo, si según la definición de co-operación, sus funciones correspondientes $f_1(H_1)$ y $f_2(H_2)$ están ligadas como

$$f(H) = f_1(H_1) | f_2(H_2).$$

Pero, por otra parte, $f(H) = \overline{f_1(H_1) \vee f_2(H_2)}$. De aquí, según la definición de la co-operación y las igualdades

$$x^\vee(H) = \{H_1', H_2'\}, \quad x^-(H_1') = H_1, \quad x^-(H_2') = H_2,$$

es decir, para la base de Sheffer la igualdad (4.15) puede reducirse al sistema

$$\begin{cases} H_1 = x^-(H_1' \in x^\vee(H)), \\ H_2 = x^-(H_2' \in x^\vee(H)). \end{cases}$$

A continuación el j -ésimo componente del resultado de la co-operación $x^{f_b}(H) = \{H_1, H_2, \dots, H_j, \dots, H_{k_{ent}}\}$ se designará por $x^{f_b}(H) | j$, es decir, $H_j = x^{f_b}(H) | j$.

En el caso que se examina la ecuación (4.15) está reducida al sistema de la forma

$$\begin{cases} H_1 = x^-(x^\vee(H) | 1), \\ H_2 = x^-(x^\vee(H) | 2). \end{cases}$$

Reduciendo el problema de determinar el resultado $\{H_1, H_2, \dots, H_{k_{ent}}\}$ de la co-operación $x^{f_b}(H)$ al problema de determinar los resultados de las co-operaciones x^\vee y x^- se reduce, en esencia, la ecuación estructural (4.14) resuelta respecto a $\{H_1, H_2, \dots, H_{k_{ent}}\}$ al sistema compuesto de k_{ent} ecuaciones estructurales con la particularidad de que cada ($j = 1, \dots$

grafo $H_b (i \in \{1, 2, \dots, n\})$ correspondiente a la función $f_b (i \in \{1, 2, \dots, n\})$, realizada por uno de los elementos de la base, contiene un ciclo.

Por ejemplo, la solución de la ecuación $x^\Delta(H) = \{H_a, H_b, H_c, H_d\}$ correspondiente a un elemento principal de SUANI (fig. 4.34) se reduce a la solución del sistema de la forma:

$$\begin{cases} H_a = x^-(x^\vee(x^-(x^\vee(H) | 1)) | 1), \\ H_b = x^\vee(x^-(x^\vee(x^-(x^\vee(H) | 1)) | 2)) | 1), \\ H_c = x^-(x^\vee(x^-(x^\vee(x^-(x^\vee(H) | 1)) | 2)) | 1), \\ H_d = x^-(x^\vee(x^-(x^\vee(H) | 2)) | 2), \\ x^-(x^\vee(x^-(x^\vee(H) | 1)) | 2) = x^\vee(x^-(x^\vee(H) | 2)) | 1. \end{cases}$$

Aquí Δ es el signo de la operación que se realiza por un elemento principal de SUANI.

En este caso, cuando se realiza la síntesis en las bases conexas, la determinación de los resultados de las co-operaciones se reduce también a la solución de un sistema de ecuaciones estructurales, pero ya dependientes una de otra.

Para sintetizar los esquemas lógicos en las bases funcionales construimos, por la función f realizada, un grafo estructural que luego transformamos en el funcional empleando la coálgebra de los grafos. El grafo funcional obtenido es el esquema lógico buscado. Proponemos el siguiente procedimiento de transformar el grafo estructural en el funcional. Examinemos el algoritmo dado a continuación en el ejemplo de la síntesis de un grafo funcional que realiza una función booleana de tres variables del contador de paridad en la base $B = \{\rightarrow, 0\}$.

El algoritmo comprende los siguientes pasos:

1. El grafo estructural H_f que realiza la función booleana f se pone en correspondencia a un elemento maximal del grafo H_b correspondiente al elemento básico $b \in B$.

2. Conforme al grafo H_b que determina las propiedades funcionales y estructurales del elemento $b \in B$ se cumplen las operaciones correspondientes de descomposición y de inversión sobre el grafo H_f .

3. Como resultado del cumplimiento de las operaciones del punto 2 se determinan los grafos estructurales H_f correspondientes a los elementos minimales del grafo H_b .

Señalemos que el elemento maximal del grafo H_b corresponde a la salida del elemento básico b , y los elementos minimales, a las entradas de este elemento.

En la fig. 4.35 se ilustra el cumplimiento de los primeros tres pasos para el ejemplo considerado.

Para cada uno de los grafos hallados H_b , los puntos 1, 2 y 3, se cumplen hasta obtener grafos estructurales que realizan funciones booleanas admi-

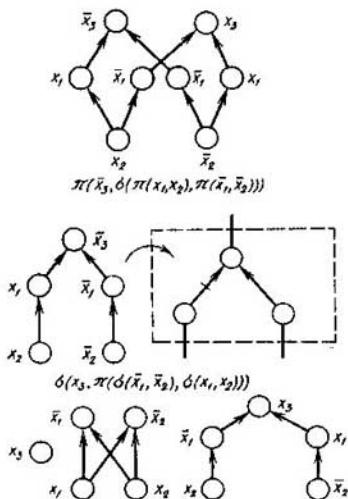


Fig. 4.35

sibles en las entradas del esquema lógico. Por lo general, tales funciones son las variables x_i o las variables y sus negaciones.

Si en i -ésimo paso de la transformación de un grafo estructural en un funcional fueron obtenidos N grafos que realizan una misma función con exactitud de hasta conjunciones idénticamente iguales a cero, estos grafos se juntan en $\lceil N/k_{ent} \rceil$ grupos ($\lceil \cdot \rceil$ es el signo del número entero más próximo; k_{ent} es el coeficiente de salida (coeficiente de ramificación) del elemento básico). De tal modo, construyendo el grafo funcional se tiene en cuenta el coeficiente de ramificación de k_{ent} elementos básicos.

En la fig. 4.36 se muestra el resultado definitivo de la transformación del grafo estructural en el funcional en la base $\{\rightarrow, 0\}$.

En las bases de la primera y la tercera clase topológica la conversión $H \rightarrow S$ de un grafo estructural H a un esquema de conmutación S se realiza de una manera trivial, sustituyendo los vértices por los elementos conmutadores y los arcos, por los canales de conexión de conductividad unilateral.

En las bases de la segunda clase topológica durante la transformación $H \rightarrow S$ pueden aparecer los caminos excedentes debidos a la conductividad bilateral de los canales de conexión, si el grafo estructural H contiene el subgrafo Q_B representado en la fig. 4.37, *a*. Al quitar la orientación de los arcos (fig. 4.37, *b*) los elementos incomparables se hacen comparables y aparece un camino excedente, lo que hace el grafo salir de la clase de los grafos equivalentes en el sentido de la realización de la función prefija-

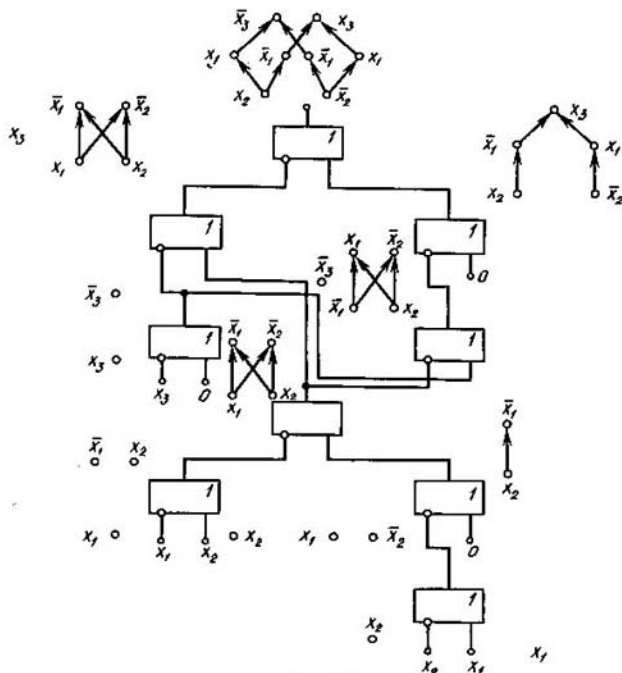


Fig. 4.36

da. Para eliminar los caminos excedentes es necesario orientar la arista diagonal colocando un diodo de desacoplo en ésta (fig. 4.37, c).

El grafo Q_B (fig. 4.37, a) es una figura prohibida de la transformación $H \rightarrow S$. Por consiguiente, la minimización de los diodos de desacoplo se reduce al cubrimiento de la tabla semántica, en la cual las figuras prohibidas son los subgrafos Q_B y sus componentes son las aristas diagonales, donde se colocan los diodos de desacoplo.

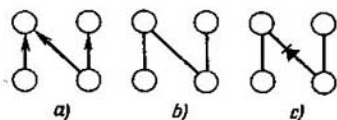


Fig. 4.37

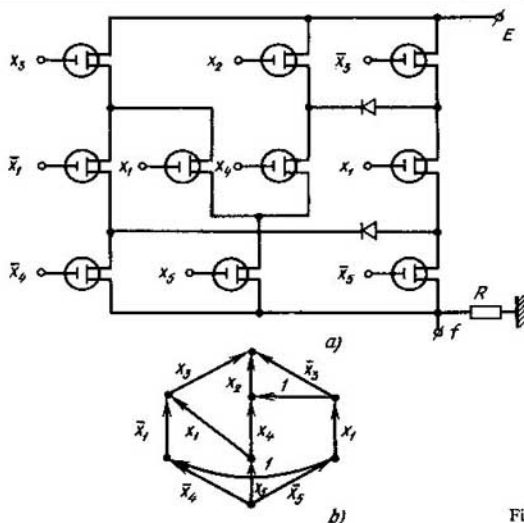


Fig. 4.38

En las bases de la segunda clase topológica, la conversión $H \rightarrow S$ se realiza (igual que en las bases de la primera clase), orientando las aristas diagonales en las figuras prohibidas Q_B . En las bases de esta clase, en la transformación $H \rightarrow S$, la abstracción del esquema de conmutación (fig. 4.38, a) es un grafo lineal (fig. 4.38, b), cuyos arcos están ponderados por las termas primarias o unidades correspondientes a los diodos de desacoplo. El grafo lineal se obtiene al sustituir los vértices del grafo estructural H por los arcos con los mismos pesos conservando los caminos de partida.

En las bases de la cuarta clase topológica, durante la transformación $H \rightarrow S$ pueden aparecer los caminos excedentes no solamente en virtud de la conductividad bilateral de los canales de conexión, sino como resultado de la conductividad bilateral de los elementos. En este caso, los caminos excedentes surgen, si el grafo estructural contiene el subgrafo Q_H , homeomorfo al subgrafo Q_B . Por consiguiente, en las bases de esta clase, la figura prohibida complementaria de la conversión $H \rightarrow S$ es el subgrafo Q_H . Para eliminar los caminos excedentes provocados por la conductividad bilateral de los propios elementos, uno de éstos que está en la diagonal de la figura prohibida, se orienta conectándole en serie el diodo de desacoplo.

En las bases de la clase considerada, la conversión $H \rightarrow S$ se realiza (igual que en las bases de la primera clase) orientando las aristas diagonales en las figuras prohibidas Q_B , Q_H .

En las bases de la cuarta clase topológica, en la transformación $H \rightarrow S$ la abstracción de esquemas de conmutación es un grafo lineal, cuyas aristas están ponderadas por las termas primarias o unidades correspondientes a los diodos de desacoplo que se obtienen al sustituir los vértices del grafo estructural H por las aristas con los mismos pesos conservando caminos de partida.

En las bases de la primera y la tercera clase topológica, la complejidad de los esquemas de conmutación es igual a la complejidad del grafo estructural. En la segunda clase topológica se agrega a esta complejidad el número de diodos de desacoplo determinado por la distribución de los subgrafos Q_B . En la cuarta clase, la complejidad de los esquemas es igual a la complejidad del grafo estructural correspondiente más el número de diodos de desacoplo determinado por la distribución de las figuras Q_B y Q_H menos el número de circuitos de longitud 2, con la particularidad de que las aristas de cada uno están ponderadas por la misma terma primaria. Cada uno de estos circuitos se sustituye por un elemento de conmutación de conductividad bilateral.

De este modo, contando solamente los elementos de conmutación, en las bases topológicas la complejidad de los esquemas lógicos es igual a la complejidad del correspondiente grafo estructural H . En las bases funcionales, el grafo estructural $H(f)$ que determina la función booleana en realización se transforma en un esquema lógico por medio de la coálgebra de los grafos.

Para describir el funcionamiento del dispositivo automático en tiempo utilizando funciones booleanas, pongamos que todas las señales que llegan puedan cambiarse en tiempo solamente de modo discreto. Al elegir intervalos suficientemente cortos de tiempo consideremos que la señal cambia solamente en la frontera de los intervalos de tiempo y no cambia dentro del intervalo.

Escogemos la duración del intervalo, partiendo de los siguientes razonamientos. Desarrollemos una *serie booleana en tiempo* $x(t)$ en la suma de *patrones de función* de la forma

$$x(t) = \bigvee_i \alpha_i a_i(t).$$

Los patrones de función más simples a_0, a_1, a_2, \dots se dan en la fig. 4.39 donde T_0 es el tiempo del análisis de funcionamiento del dispositivo automático; con ello,

$$T_i = T_0 / 2^i, \quad (4.17)$$

donde T_i es el periodo del patrón de función a_i . Según (4.7) la duración del intervalo minimal

$$T_k = T_0 / 2^{\gamma+1},$$

donde γ es la *profundidad de cuantificación en tiempo*.

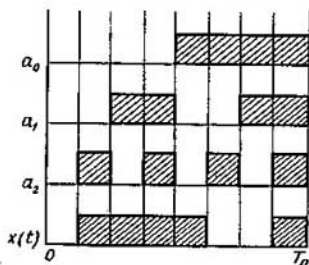


Fig. 4.39

Para cualquier dispositivo automático existe el tiempo mínimo T_{\min} entre dos transiciones adyacentes. Para describir el funcionamiento del dispositivo automático en tiempo con el grado suficiente de exactitud, es necesario que

$$T_{\min} \geq T_k \text{ o } \gamma \geq \log_2 \frac{T_0}{T_{\min}} \quad (4.18)$$

Así, pues, para la representación analítica de las series booleanas en tiempo con el grado dado de exactitud, es necesario tener la profundidad de cuantificación en tiempo $\gamma \geq \log_2 (T_0/T_{\min})$.

Un intervalo temporal mínimo T_k se denominará *cuanto en tiempo*. Entonces, a cada cuanto le corresponde una constituyente de la unidad de una función booleana temporal $\varphi(a_0, a_1, a_2, \dots, a_\gamma)$ cuyas variables son patrones de función $a_0, a_1, a_2, \dots, a_\gamma$. Por consiguiente, cualquier serie booleana en tiempo con el grado dado de exactitud puede representarse como una función booleana de $a_0, a_1, \dots, a_\gamma$ y utilizarse como una función booleana habitual. Por ejemplo, la serie booleana en tiempo $x(t)$ (fig. 4.39) puede representarse en la forma

$$x(t) = \bar{a}_0(a_1 \vee a_2) \vee a_0(\bar{a}_1 \bar{a}_2 \vee a_1 a_2).$$

Cada elemento real que integra el esquema lógico que se sintetiza posee una constante de tiempo. Por lo tanto, en los esquemas lógicos prácticos tienen lugar *procesos transitorios* que debemos tener en cuenta. Examinemos el empleo del concepto de la derivada para investigar los procesos transitorios en esquemas.

$$\frac{\partial x}{\partial t} = x(t) \oplus x(t \neq 1), \quad (4.19)$$

se llama *derivada en tiempo* $\frac{\partial x}{\partial t}$ de una variable booleana $x(t)$; $x(t)$ es una sucesión booleana en tiempo que toma los valores 0, 1 en los momentos de tiempo 0, 1, 2, ... Para más precisión, en (4.19) tomemos el signo «-».

La derivada en tiempo muestra la variación de la señal en tiempo. Examinaremos las series booleanas periódicas.

Ilustremos el concepto de la derivada temporal en el siguiente ejemplo. Si la variable $x(t)$ en tiempo $t = 0, 1, 2, 3, 4, 5$ cambia respectivamente así: $x(t) = 0, 1, 0, 1, 1, 1$, la derivada temporal cambia como $\frac{\partial x}{\partial t} = 1, 1, 1, 1, 0, 0$.

Consideremos un esquema lógico de dos entradas que se conmutan simultáneamente. Para más precisión examinaremos la transición $(0, 0) \rightarrow (1, 1)$. La condición de conmutación simultánea de los canales de entrada x_1 y x_2 es ideal. En realidad, con la probabilidad casi igual a la unidad, cambia, primero, una entrada y, dentro de un rato otra. En dependencia del orden de su conmutación, el camino de $(0, 0)$ a $(1, 1)$ puede pasar a través de $(1, 0)$ ó $(0, 1)$. Analicemos dos casos:

1. Las funciones $f(0, 0)$ y $f(1, 1)$ no son iguales. La señal de salida antes y después de la conversión es diferente. Entonces, en los estados intermedios puede tener lugar ora el valor «viejo» de la señal, ora el «nuevo». Si en el esquema faltan picos, en principio no puede haber ninguna señal falsa.

2. Las funciones $f(0, 0)$ y $f(1, 1)$ son iguales. La señal de salida antes y después de la conversión es igual, es decir, en la salida la conversión «no se siente». Pero, si en los estados intermedios la señal de salida se difiere de $f(0, 0)$ ($f(1, 1)$), durante la conmutación puede aparecer una señal falsa, la conversión es crítica. Este fenómeno se denominará *riesgo* en el esquema lógico. El camino, por el cual la señal pasará del estado $(0, 0)$ al $(1, 1)$, depende de los parámetros físicos de las señales del esquema y de la suerte, ya que los parámetros tienen carácter estadístico.

Siguiendo a Bochmann, pongamos las condiciones necesarias y suficientes de la conversión crítica. Para que la conversión del estado (σ_1, σ_2) en el $(\bar{\sigma}_1, \bar{\sigma}_2)$ ($\sigma_1, \sigma_2 = 0, 1$) sea crítica es necesario y suficiente que

$$a) f(\sigma_1, \sigma_2) = f(\bar{\sigma}_1, \bar{\sigma}_2);$$

$$b) f(\bar{\sigma}_1, \sigma_2) \neq f(\sigma_1, \sigma_2) \text{ o } f(\sigma_1, \bar{\sigma}_2) \neq f(\sigma_1, \sigma_2).$$

Es fácil enunciar estas condiciones empleando las derivadas. En caso de la conversión crítica tenemos:

$$a) \text{ la conmutación simultánea de ambas entradas: } \frac{\partial x_1}{\partial t} \frac{\partial x_2}{\partial t} = 1;$$

$$b) \text{ el valor de la señal de salida antes y después de la conversión es igual a: } \frac{\partial^2 f}{\partial(x_1 x_2)} = 0;$$

$$c) \text{ la conmutación solamente de una entrada lleva a la conmutación de la salida: } \frac{\partial f}{\partial x_1} \vee \frac{\partial f}{\partial x_2} = 1.$$

Por lo tanto, el error se expresa del modo siguiente:

$$\Delta f = \frac{\partial x_1}{\partial t} \frac{\partial x_2}{\partial t} \left(\frac{\partial f}{\partial x_1} \vee \frac{\partial f}{\partial x_2} \right) \frac{\partial^2 f}{\partial(x_2 x_1)} \quad (4.20)$$

donde Δf es la función del error.

En la expresión (4.20), la conjunción $\frac{\partial x_1}{\partial t} \frac{\partial x_2}{\partial t}$ determina las propiedades de la señal, a continuación la denominaremos *miembro de señal*. La conjunción $\left(\frac{\partial f}{\partial x_1} \vee \frac{\partial f}{\partial x_2} \right) \frac{\partial^2 f}{\partial(x_1 x_2)}$ determina las propiedades de la función que se realiza; en adelante la denominaremos *miembro funcional* de la fórmula que define el error.

Uno de los procedimientos para aumentar el rendimiento del ordenador es el incremento de la frecuencia de trabajo de los elementos utilizados. Los elementos electrónicos moleculares modernos tienen frecuencia de trabajo de miles megahertzios. Durante el funcionamiento de los esquemas lógicos, el retardo en el circuito de los elementos se hace comparable con el periodo de trabajo del esquema, lo que lleva a la necesidad de tomar en consideración las señales falsas determinadas por la expresión (4.20).

El error no surge, si $\Delta f = 0$, para lo cual es necesario hacer restricciones en el miembro de señal o en el funcional de la fórmula.

Examinemos el caso de los esquemas lógicos, en los cuales tres canales de entrada se conmutan simultáneamente. Según (4.20) el error tiene lugar cuando se conmutan dos entradas arbitrarias. Por consiguiente, en la conmutación de tres entradas x_a, x_b, x_c la función de error contiene los siguientes miembros:

$$\begin{aligned} & \frac{\partial x_a}{\partial t} \frac{\partial x_b}{\partial t} \left(\frac{\partial f}{\partial x_a} \vee \frac{\partial f}{\partial x_b} \right) \frac{\partial^2 f}{\partial(x_a x_b)} ; \\ & \frac{\partial x_a}{\partial t} \frac{\partial x_c}{\partial t} \left(\frac{\partial f}{\partial x_a} \vee \frac{\partial f}{\partial x_c} \right) \frac{\partial^2 f}{\partial(x_a x_c)} ; \\ & \frac{\partial x_b}{\partial t} \frac{\partial x_c}{\partial t} \left(\frac{\partial f}{\partial x_b} \vee \frac{\partial f}{\partial x_c} \right) \frac{\partial^2 f}{\partial(x_b x_c)} . \end{aligned}$$

La fórmula que determina las condiciones en que surgen los errores al conmutar tres entradas, contiene otro miembro que tiene en cuenta la conmutación simultánea de tres entradas:

$$\frac{\partial x_a}{\partial t} \frac{\partial x_b}{\partial t} \frac{\partial x_c}{\partial t} \left(\frac{\partial^2 f}{\partial(x_a x_b)} \vee \frac{\partial^2 f}{\partial(x_a x_c)} \vee \frac{\partial^2 f}{\partial(x_b x_c)} \right) \frac{\partial^3 f}{\partial(x_a x_b x_c)} .$$

Por lo tanto, para la conmutación de tres entradas la función de error tiene forma:

$$\Delta f = \frac{\partial x_a}{\partial t} \frac{\partial x_b}{\partial t} \left(\frac{\partial f}{\partial x_a} \vee \frac{\partial f}{\partial x_b} \right) \frac{\partial^2 f}{\partial(x_a x_b)} \vee$$

$$\begin{aligned} & \vee \frac{\partial x_a}{\partial t} \frac{\partial x_c}{\partial t} \left(\frac{\partial f}{\partial x_a} \vee \frac{\partial f}{\partial x_c} \right) \frac{\overline{\partial^2 f}}{\partial(x_a x_c)} \vee \\ & \vee \frac{\partial x_b}{\partial t} \frac{\partial x_c}{\partial t} \left(\frac{\partial f}{\partial x_b} \vee \frac{\partial f}{\partial x_c} \right) \frac{\overline{\partial^2 f}}{\partial(x_b x_c)} \vee \\ & \vee \frac{\partial x_a}{\partial t} \frac{\partial x_b}{\partial t} \frac{\partial x_c}{\partial t} \left(\frac{\partial f}{\partial(x_a x_b)} \vee \frac{\partial^2 f}{\partial(x_a x_c)} \vee \frac{\partial^2 f}{\partial(x_b x_c)} \right) \frac{\overline{\partial^3 f}}{\partial(x_a x_b x_c)}. \end{aligned} \quad (4.21)$$

Designemos el miembro funcional de la función de error que integra una misma conjunción con $\frac{\partial x_{a_1}}{\partial t} \frac{\partial x_{a_2}}{\partial t} \dots \frac{\partial x_{a_n}}{\partial t}$, como $F(x_{a_1}, x_{a_2}, \dots, x_{a_n})$. Entonces (4.21) puede escribirse en la forma

$$\begin{aligned} \Delta f = & \frac{\partial x_a}{\partial t} \frac{\partial x_b}{\partial t} F(x_a, x_b) \vee \frac{\partial x_a}{\partial t} \frac{\partial x_c}{\partial t} F(x_a, x_c) \vee \\ & \vee \frac{\partial x_b}{\partial t} \frac{\partial x_c}{\partial t} F(x_b, x_c) \vee \frac{\partial x_a}{\partial t} \frac{\partial x_b}{\partial t} \frac{\partial x_c}{\partial t} F(x_a, x_b, x_c). \end{aligned}$$

Generalizando la función de error para el caso de conmutación de n entradas en el esquema lógico, obtenemos

$$\begin{aligned} F(x_{a_1}, x_{a_2}, \dots, x_{a_n}) = & \left(\bigvee_{\substack{i_1, i_2, \dots, i_{n-1} = x_{a_1}, x_{a_2}, \dots, x_{a_n} \\ i_j \neq i_{j+1}, \dots, i_{j-2} \neq i_{j-1}}} \frac{\partial^{n-1} f}{\partial(i_1; i_2, \dots, i_{n-1})} \right) \times \\ & \times \frac{\overline{\partial^n f}}{\partial(x_{a_1}, x_{a_2}, \dots, x_{a_n})} \end{aligned} \quad (4.22)$$

y la función de error tiene forma

$$\begin{aligned} \Delta f = & \bigvee \frac{\partial i_1}{\partial t} \frac{\partial i_2}{\partial t} F(i_1, i_2) \vee \bigvee \frac{\partial i_1}{\partial t} \frac{\partial i_2}{\partial t} \frac{\partial i_3}{\partial t} F(i_1, i_2, i_3) \vee \dots \vee \\ & \vee \frac{\partial x_{a_1}}{\partial t} \frac{\partial x_{a_2}}{\partial t} \dots \frac{\partial x_{a_n}}{\partial t} F(x_{a_1}, x_{a_2}, \dots, x_{a_n}), \quad i_1, i_2, \dots = \\ & = a_1, a_2, \dots, a_n. \end{aligned} \quad (4.23)$$

De la fórmula (4.23) se desprende que con el aumento del número de canales de entrada que se conmutan aumenta la probabilidad de la conversión crítica. A la conmutación de tres canales en la entrada del esquema, la probabilidad de riesgo no es menor del 75%. La función de error Δf es de carácter estadístico.

§ 4.8. Simulación de los sistemas de autómatas mediante las redes de Petri

Debido al empleo cada vez más amplio de los sistemas de cálculo paralelos y distribuidos adquieren especial actualidad las estructuras discretas que representan los procesos en paralelo. Como aparato para describir los siste-

mas complejos de los procesos en interacción se presentan los sistemas formales tipo *redes de Petri* que simulan las propiedades dinámicas de los sistemas.

El carácter formal de las redes de Petri tipo común se basa en el concepto del acopio que es cierta generalización del concepto del conjunto. Al igual que el conjunto, el *acopio* es un juego de los elementos, pero cualquier elemento puede incluirse en este acopio más de una vez. En otras palabras, la relación de inclusión que une los elementos y conjuntos se sustituye por una *función del número de ejemplares* del elemento en el acopio que se denota por medio de $\#(x, B)$ (se lee: «el número x en el acopio B »). El conjunto es un caso particular del acopio.

Muchos conceptos de la teoría de los conjuntos se distribuyen también en los acopios. Así, el *acopio vacío* es análogo al conjunto vacío. La *potencia de un acopio* es el número total de ejemplares de los elementos en el acopio. El acopio A se incluye en el acopio B (es un subacopio), si para cualquier x $\#(x, A) \leq \#(x, B)$. Empleando la función $\#$ es fácil determinar las *operaciones sobre los acopios*: para la *unión de los acopios* A y B $\#(x, A \cup B) = \max(\#(x, A), \#(x, B))$; para la *intersección de los acopios* A y B $\#(x, A \cap B) = \min(\#(x, A), \#(x, B))$; para la *suma de los acopios* A y B $\#(x, A + B) = \#(x, A) + \#(x, B)$; para la *diferencia de los acopios* A y B $\#(x, A - B) = \#(x, A) - \#(x, A \cap B)$. Si M es un conjunto, M^n es el conjunto de todos los acopios, contruidos de los elementos M , tales que $\#(x, B) \leq n, B \in M^n$; M^∞ es el conjunto de todos los acopios contruidos de los elementos M sin limitar el número de ejemplares del elemento en el acopio.

La *red de Petri* es una cuaterna $C = (P, T, I, O)$ donde P es un conjunto finito de *posiciones*, T es un conjunto finito de *transiciones*, $I: T \rightarrow P^\infty$ es la función de entrada que aplica las transiciones a los acopios de posiciones; $O: T \rightarrow P^\infty$ es la función de salida que aplica las transiciones a los acopios de posiciones. La red de Petri se representa gráficamente en forma de un grafo múltiple con los vértices de dos tipos: los círculos corresponden a las posiciones, las rayas, a las transiciones. Las funciones I y O se representan por arcos (fig. 4.40).

Las posiciones desde las cuales los arcos llevan a la transición t_j se denominan *de entrada* para t_j ; de manera análoga, las posiciones a las cuales llevan los arcos desde la transición t_j se llaman *de salida* para t_j . El conjunto de las posiciones de entrada se denota mediante $I(t_j)$, de salida, por medio de $O(t_j)$. En la red de Petri representada en la fig. 4.40 se tiene $I(t_1) = \{p_1, p_1, p_1\}$, $O(t_1) = \{p_3, p_4, p_4\}$. Es cómodo generalizar las funciones I y O sobre la aplicación de las posiciones en los acopios de transiciones ($P \rightarrow T^\infty$), lo que permite designar los conjuntos de las transiciones de entrada y de salida de la posición p_i (que se determinan de modo análogo a los conjuntos de las posiciones de entrada y de salida de la transición) por medio de

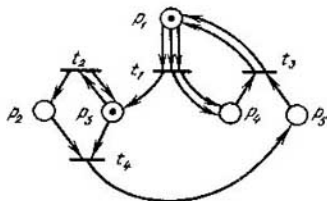


Fig. 4.40

$I(p_i)$ y $O(p_i)$, respectivamente. En la red de Petri representada en la fig. 4.40 se tiene $I(p_3) = \{t_1, t_2\}$, $O(p_3) = \{t_2, t_4\}$.

Los conceptos introducidos se refieren a la estructura estática de la red de Petri. Las propiedades dinámicas de esta red se determinan utilizando el concepto de la marcación. La *marcación* μ de la red de Petri $C = (P, T, I, O)$ es una función que aplica el conjunto de posiciones P en el conjunto de los números enteros no negativos N . La marcación se representa empleando *fichas* (puntos) que se colocan dentro de las posiciones. Así, la marcación de la red de Petri dada en la fig. 4.40 se determina como $\mu(p_1) = \mu(p_3) = 1$, $\mu(p_2) = \mu(p_4) = \mu(p_5) = 0$.

Es cómodo representar la marcación ora como un n -vector $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ (donde $n = |P|$), cada elemento del cual μ_i es $\mu(p_i)$, ora como un acopio μ que comprenda las posiciones de la red $p_i \in P$ y $\#(p_i, \mu) = \mu(p_i)$. La red de Petri C , con la marcación μ determinada en ella, se denomina *red marcada* de Petri.

El *arranque de transiciones* puede cambiar la marcación de la red. La transición t_j de una red marcada de Petri C con la marcación μ se denomina *permitida*, si $I(t_j) \subseteq \mu$, es decir, en cada posición de entrada p_j hay número de fichas que no es menor que el número de arcos salientes de esta posición en t_j . Se puede hacer arrancar cualquier transición permitida. Como resultado del *arranque* de la transición t_j la marcación μ de la red se sustituye por una nueva: $\mu' = \mu - I(t_j) + O(t_j)$, es decir, de cualesquiera posición de entrada p_i de una transición t_j se quitan tantas fichas, cuantos arcos conducen de p_i a t_j y en cada posición de salida p_k se colocan tantas fichas, cuantos arcos conducen de t_j a p_k . La sucesión de arranques de las transiciones se denomina *cumplimiento de la red de Petri*.

Examinemos el cumplimiento de la red de Petri mostrada en la fig. 4.40. En la *marcación inicial* se permite solamente la transición t_2 . Durante su arranque la ficha se quita de p_3 y luego en cada una de las posiciones p_2 y p_3 se añade una ficha más, es decir, después del arranque en la marcación nueva μ' aparece también una ficha en p_2 . Ahora las transiciones t_2, t_4 se hacen permitidas. Ya que se puede hacer arrancar cualesquiera transición permitida supongamos que tal es la transición t_4 . Después de su arranque las fichas se quitan de las posiciones p_2 y p_3 y en la posición p_5 aparece una ficha. En la marcación obtenida μ'' ninguna transición es permitida. Así se termina el cumplimiento de la red de Petri.

Analicemos la marcación μ de la red de Petri $C = (P, T, I, O)$. La marcación μ' se denomina *directamente alcanzable* desde μ , si existe una transición $t_j \in T$, permitida en μ , tal que después de su arranque se obtiene la marcación μ' ; en este caso el par (μ, μ') pertenece a la *relación de alcance directa* determinada sobre P^∞ . La clausura transitiva de esta relación se denomina *relación de alcance*. Las marcaciones μ' , tales que (μ, μ') pertenecen a la relación de alcance y se llaman *alcanzables* desde μ . El conjunto de marcaciones alcanzables desde μ de la red de Petri C se denomina *conjunto de alcance* y se denota mediante $R(C, \mu)$.

La interpretación de las redes de Petri se basa en los conceptos de *condición* y *suceso*. El estado del sistema se describe por una colección de condiciones. El funcionamiento del sistema consiste en realizar la sucesión de ciertas actuaciones, es decir, sucesos. Para que surja un suceso, es necesario cumplir unas condiciones llamadas *precondiciones*. El surgimiento del suceso puede infringir las precondiciones y cumplir otras llamadas *postcondiciones*. En la red de Petri las posiciones simulan las condiciones, las transiciones simulan los sucesos. Las precondiciones del suceso se representan por las posiciones de entrada de la correspondiente transición, las poscondiciones, por las posiciones de salida. El surgimiento del suceso se simula por el arranque de la transición. En las correspondientes posiciones, la presencia de fichas representa el cumplimiento de condiciones, su ausencia, el no cumplimiento.

Por ejemplo, analicemos un sistema simple de cálculo que procesa en serie los trabajos llegados a la cola de entrada. Si el procesor está libre y hay trabajo en la cola de entrada, éste se trata por el procesor y después sale. La red de Petri dada en la fig. 4.41 puede simular este sistema.

Establezcamos que particularidades de los sistemas tienen en cuenta las redes de Petri. En primer lugar, es la *asincronidad*. En la red de Petri no existe el concepto de tiempo. El tiempo de surgimiento de los sucesos no se indica de ningún modo. No obstante, la estructura de la red de Petri

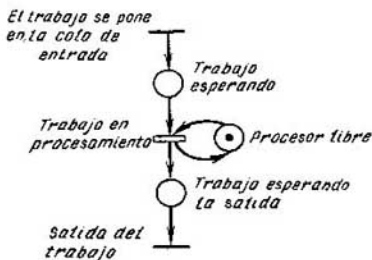


Fig. 4.41

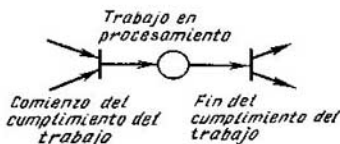


Fig. 4.42

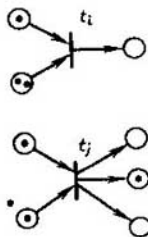


Fig. 4.43

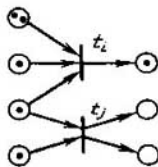


Fig. 4.44

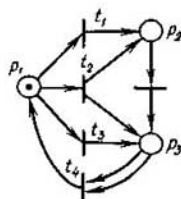


Fig. 4.45

establece el orden parcial de surgimiento de los sucesos. Es más, puesto que el surgimiento de los sucesos se representa por el arranque de transiciones, se supone que los sucesos transcurren *en un instante*. Si el suceso que se simula tiene una duración distinta de cero, por ejemplo, el suceso «trabajo en procesamiento» (fig. 4.41), y esto es sustancial, lo representamos en forma de dos sucesos instantáneos de tipo «comienzo del suceso», «fin del suceso» y la condición «suceso transcurriendo» (fig. 4.42). Además, se considera que los sucesos transcurren *no simultáneamente* (los sucesos instantáneos no pueden acontecer al mismo tiempo). En efecto, si admitimos el surgimiento simultáneo de algunos sucesos i y j , a los cuales en la red de Petri les corresponden las transiciones t_i y t_j , se puede introducir una transición complementaria t_{ij} con $I(t_{ij}) = I(t_i) + I(t_j)$, $O(t_{ij}) = O(t_i) + O(t_j)$ que se interpreta como el surgimiento simultáneo de los sucesos i y j . En este caso se puede hacer que las transiciones arranquen en serie.

Otra propiedad importante de las redes de Petri en calidad de instrumento de simulación es su capacidad de representar el *paralelismo* y *situaciones conflictivas*. El paralelismo de dos sucesos se representa por dos transiciones permitidas, cuyos conjuntos de posiciones de entrada no se intersecan (fig. 4.43), el conflicto se representa por las transiciones con una posición común de entrada (fig. 4.44).

En lo principal, las redes de Petri se utilizan como un aparato formal en la simulación de los sistemas de paralelismo inherente. Al examinar el proceso de proyección en total son posibles dos enfoques, distintos en principio, del empleo de las redes de Petri. En el primer caso, el sistema se simula por una red de Petri que se transforma, según las reglas determinadas, hasta llegar a un aspecto «óptimo». La red obtenida de Petri se transforma en un proyecto del sistema. Se supone que es también «óptimo». Aquí las redes de Petri se aplican directamente para proyectar. Empero, este enfoque tiene las dificultades vinculadas con la multiformidad de la transformación inversa, las redes de Petri en el proyecto del sistema, lo que pone en duda la optimicidad del proyecto que se obtiene. En el segundo

enfoque más conocido por todos primeramente se crea, empleando los medios habituales, el proyecto del sistema y, según éste, se construye un modelo en forma de la red de Petri. Después se investigan las propiedades de la red obtenida y se hacen deducciones sobre las propiedades y características del proyecto. Si no son satisfactorios, los datos obtenidos al investigar las redes de Petri se utilizan para modificar el proyecto. El proyecto modificado vuelve a transformarse en la red de Petri y el ciclo se repite. Este proceso se termina cuando la red de Petri posea las propiedades necesarias.

Examinemos que propiedades de las redes de Petri en calidad de modelo del sistema pueden interesar al diseñador. Una de las propiedades más importantes es la seguridad. La posición de la red de Petri se denomina *segura*, si en ésta el número de fichas nunca supera 1. La red marcada de Petri es *segura*, si todas sus posiciones son seguras. Esta propiedad es muy importante para interpretar las posiciones como condiciones simples: si en la posición hay una ficha, la condición se cumple, si no hay, no se cumple. Si la interpretación de fichas es más complicada (por ejemplo, el número de fichas indica el número de unidades informativas), puede tener interés la pregunta, si es limitado el número de fichas en la posición dada y, si lo es, cuáles son sus límites. De este modo, llegamos a la propiedad de limitación. La posición se denomina *k-limitada*, si en ésta el número de fichas no supera un k entero para cualquier marcación alcanzable. Una red marcada de Petri se denomina *k-limitada*, si sus posiciones son *k-limitadas*. En la red de Petri dada en la fig. 4.45 las posiciones p_1 y p_2 son seguras, la posición p_3 es 2-limitada y toda la red es 2-limitada.

En el caso de interpretar las fichas como unos recursos, éstas no deben crearse ni eliminarse. En otras palabras, en la red debe actuar la ley de conservación. Una red marcada de Petri se denomina *estrictamente conservante*, si la potencia de marcación (como de un acopio de posiciones) es constante. En caso general la ficha puede interpretarse como un número de recursos elementales con la particularidad de que este número se varía de una posición a otra. Introduzcamos el concepto de *ponderación de posiciones*: un vector $\mathbf{W} = (w_1, w_2, \dots, w_n)$, donde w_n es el peso de la posición p_i . Una red de Petri se denomina *conservante respecto al vector de ponderación \mathbf{W}* , si el producto escalar del vector \mathbf{W} y la marcación (considerada como un vector) es constante; una red de Petri es *conservante*, si es conservante respecto al vector de ponderación \mathbf{W} , todos los elementos del cual son positivos.

Las propiedades consideradas hasta ahora se refieren tanto a los sistemas sucesivos como a los paralelos. Pero pasando de los sistemas en serie a los paralelos surgen nuevas dificultades en principio: la posibilidad de *situaciones tope*. Llámase *tope* en una red de Petri un conjunto de transiciones que no son permitidas en una marcación alcanzable μ' y las marca-

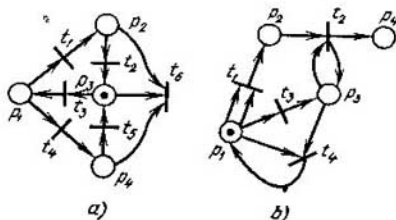


Fig. 4.46

ciones posteriores alcanzables de μ' . La posibilidad de que surjan topes en el sistema se simula por la propiedad de actividad en las redes de Petri. Una transición t_j se denomina *activa*, si no intergira ningún tope. La transición se llama *pasiva*, si no es permitida en ninguna marcación alcanzable. En la investigación detallada de la actividad de la red de Petri se usa también el concepto de niveles de actividad. La transición t_j posee la *actividad* del *nivel 0*, si no puede ser arrancada (es pasiva); del *nivel 1*, si potencialmente puede ser arrancada, o sea, si existe una marcación alcanzable, en la cual la transición es permitida; del *nivel 2*, si para cualquier k entero existe una sucesión de arranques de transiciones, en la cual esta transición está presente no menor que k veces; del *nivel 3*, si existe la sucesión infinita de arranques, en la cual esta transición está presente con la frecuencia infinita; del *nivel 4*, si potencialmente puede arrancar de cualesquiera marcación alcanzable (o sea, es activa). En la red de Petri dada en la fig. 4.46, *a* la transición t_3 es activa, las t_1, t_2, t_4, t_5 tienen nivel de actividad 3, la t_6 es pasiva. En la red de Petri representada en la fig. 4.46, *b* la transición t_4 es pasiva, la t_3 posee la actividad del nivel 1, la t_2 , la del nivel 2, la t_1 , la del nivel 3.

Uno de los problemas más importantes en el análisis de las redes de Petri es el *problema de alcance*: ¿es alcanzable la marcación μ' de la marcación inicial μ para la red de Petri dada? La importancia de este problema se desprende de que la marcación sirve de interpretación del estado del sistema. La solución del problema de alcance permite determinar, si es alcanzable un estado determinado sea «bueno» o «malo» para el sistema.

Las propiedades descritas y los problemas correspondientes del análisis de las redes de Petri son los más comunes aunque no abarcan todo el conjunto de cuestiones que pueden surgir al analizar las redes de Petri. Hay dos procedimientos principales para resolver los problemas del análisis. El primero se basa en la construcción del árbol de alcance. Llámase *árbol de alcance* un árbol orientado con raíz, a los vértices del cual les corresponden las marcaciones posibles y a los arcos, las transiciones. Al vértice radical le corresponde la marcación inicial. De cada vértice salen los arcos correspondientes a las transiciones permitidas. La construcción del árbol

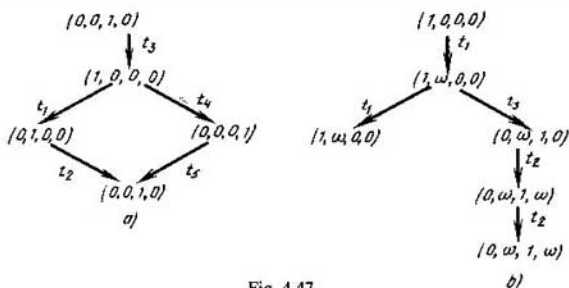


Fig. 4.47

se realiza sucesivamente, partiendo del vértice radical; a cada paso se forma un nivel siguiente del árbol. Por ejemplo, después de tres pasos el árbol de alcance para la red de Petri representada en la fig. 4.46, *a* tiene forma dada en la fig. 4.47, *a* (los vectores representan las marcaciones). Es obvio, que si construyendo un árbol no utilizamos acuerdos determinados, las redes activas (hasta limitadas) de Petri tendrán un árbol infinito de alcance.

Llamaremos *de frontera* los vértices (y las marcaciones respectivas) construidos a un paso siguiente del algoritmo. Si en una marcación de frontera no hay transiciones permitidas, la denominaremos *terminal*. Si un vértice de frontera tiene marcación ya existente en el árbol, la llamaremos *duplicadora*. Para los vértices terminales y duplicadores no construiremos los arcos salientes de ellos. Esto asegura el árbol finito de alcance para la red limitada de Petri (por ejemplo, fig. 4.46, *a* y 4.47, *a*). Para las redes ilimitadas hace falta designar de cualquier modo el número infinito de fichas en la posición. Sea que ω designa este número con tal que $\omega + a = \omega$, $\omega - a = \omega$, $a < \omega$, $\omega \leq \omega$, donde a es un número positivo entero arbitrario. En la construcción del árbol de alcance usaremos la siguiente regla. Sea que un vértice de frontera μ no es terminal, ni duplicador. En la marcación μ para cada transición permitida t_j construyamos un arco saliente de μ y lo denotemos por la transición t_j . La marcación μ' del vértice nuevo se determina del siguiente modo. Si $\mu(p_i) = \omega$, $\mu'(p_i) = \omega$. Si, en el camino del vértice radical a μ , existe un vértice μ'' , tal que después del arranque de la transición t_j en μ , el número de fichas en toda posición no es menor que en μ'' y en la posición p_i es estrictamente mayor, entonces $\mu'(p_i) = \omega$. En caso contrario $\mu'(p_i)$ es el número de fichas en la posición p_i que se obtiene después de que t_j arranque de μ (fig. 4.47, *b*).

Teorema 4.4. *El árbol de alcance de cualesquiera red de Petri es finita.*

La demostración de esta afirmación se basa en las propiedades ω y en las reglas de introducir este símbolo en la marcación de los vértices de frontera.

El método de análisis basado en el árbol de alcance permite determinar las propiedades de seguridad, limitación, conservación e investigar las propiedades de actividad y de alcance.

La red de Petri es limitada si, y sólo si, no hay símbolo ω en el árbol de alcance. Además, la posición del símbolo ω indica que posiciones son ilimitadas. Si el símbolo ω está ausente en el árbol, el número de marcas alcanzables es finito y todas las cuestiones del análisis pueden resolverse mediante el sondeo simple. En particular, para hallar la frontera de marcación de la posición dada p_i , hay que encontrar el valor máximo del i -ésimo componente entre todos los vértices del árbol. Si esta frontera no supera 1, la posición es segura.

Para establecer si la red de Petri es conservante respecto a un vector de ponderación $\mathbf{W} = (w_1, w_2, \dots, w_n)$, es necesario resolver el siguiente sistema de ecuaciones lineales con restricciones:

$$\begin{cases} \sum_{i=1}^n w_i \mu_j(p_i) = s, & \text{cuando } j = (1, \dots, k), \\ w_i \geq 0, & \text{cuando } i = (1, \dots, n), \end{cases}$$

donde k es el número de vértices del árbol de alcance, a los cuales corresponden distintas marcaciones. (Es evidente que si tenemos $\mu_j(p_i) = \omega$, $w_i = 0$.)

La posibilidad de resolver los problemas de actividad y de alcance está limitada por la existencia del símbolo ω que esconde la información concreta sobre el número de fichas. Por ejemplo, una vez introducidos dos arcos (t_1, p_2) , (p_2, t_2) en la red de Petri representada en la fig. 4.46, *b*, la red obtenida de Petri tendrá el mismo árbol de alcance que la inicial. Al mismo tiempo, en la nueva red de Petri, en la posición p_2 , puede encontrarse solamente un número par de fichas, mientras que en la red inicial puede haber cualquier número de fichas, es decir, los conjuntos de alcance para estas redes de Petri no coinciden. Se puede aducir distintas redes con diferentes propiedades de actividad, pero deben tener un mismo árbol de alcance.

No obstante, aunque el árbol de alcance no presta la información completa sobre las propiedades de alcance y de actividad, en algunos casos permite responder a las preguntas sobre el alcance y la actividad. Por ejemplo, si tiene un vértice terminal, la red de Petri no es activa. Resolviendo el problema de alcance puede ocurrir que la marcación μ' está presente en el árbol de alcance (la respuesta positiva) o que la marcación μ' no se cubre por ningún vértice del árbol de alcance, es decir, $\mu' \not\geq \mu$ para todos los vértices μ (la respuesta negativa).

Otro enfoque del análisis de las redes de Petri se denomina *matricial* y se basa en su representación matricial. Introduzcamos las matrices D^- y D^+ , a cuyas columnas les corresponden las posiciones, a las filas, las transiciones y $D^-(j, i) = \#(p_i, I(t_j))$, $D^+(j, i) = \#(p_i, O(t_j))$. Sea $e(j)$ un

vector-fila, cuyos componentes corresponden a las transiciones y todos son iguales a cero, excepto el j -ésimo que es igual a 1. Entonces la transición t_j está permitida, si $\mu \geq e(j)$. D^- (μ se examina como un vector) y el resultado del arranque de t_j desde μ es $\mu' = \mu - e(j)D^- + e(j)D^+ = \mu + e(j) \times (D^+ - D^-) = \mu + e(j) \cdot D$, donde $D = D^+ - D^-$ es una *mátriz compuesta de cambios*.

La marcación μ' obtenida de μ como resultado del arranque de la sucesión $\sigma = t_{j_1} t_{j_2} \dots t_{j_k}$ se determina como

$$\begin{aligned}\mu' &= \mu + e(j_1)D + e(j_2)D + \dots + e(j_k)D = \\ &= \mu + (e(j_1) + \dots + e(j_k))D = \mu + f(\sigma)D,\end{aligned}$$

donde $f(\sigma) = e(j_1) + \dots + e(j_k)$ es el *vector de arranque*, cuyo j -ésimo componente es igual al número de arranques de t_{j_i} en σ .

Si una red marcada de Petri es conservante respecto a un vector de ponderación W (W es vector-columna), se tiene $\mu W = \mu' W$ para cualquier $\mu' = R(C, \mu)$. Ya que $\mu' = \mu + f(\sigma)D$, entonces $f(\sigma)DW = 0$. Puesto que esto es válido para todos los $f(\sigma)$, tenemos $DW = 0$. Por lo tanto, la red de Petri es conservante respecto a un vector de ponderación si, y sólo si, existe un vector W tal que $DW = 0$. Esta ecuación permite hallar el vector de ponderación W .

Si la marcación μ' es alcanzable de la marcación inicial μ de la red de Petri debe existir una solución entera no negativa de la ecuación $\mu' = \mu + xD$, cuya solución será $x = f(\sigma)$.

Investiguemos el problema de alcance para la red de Petri dada en la fig. 4.46, *b* con la marcación inicial $(1, 0, 0, 0)$ para la marcación $\mu' = (0, 2, 1, 2)$. La ecuación $\mu' = \mu + xD$ toma la forma

$$(0, 2, 1, 2) = (1, 0, 0, 0) + x \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

y tiene solución $x = (4, 2, 1, 0)$ que corresponde a la sucesión de arranques de las transiciones $t_1 t_1 t_1 t_1 t_2 t_2$.

El enfoque matricial del análisis de las redes de Petri, igual que el enfoque basado en el árbol de alcance, en el caso general no permite resolver el problema de alcance y de actividad. Los problemas del análisis matricial consisten en que el vector de arranque obtenido durante la solución de la ecuación 1) no proporciona información sobre el orden de arranque de transiciones y 2) puede corresponder a la sucesión no permitida de arranques.

En el presente está demostrado que los problemas de alcance y de actividad son *equivalentes* pero no se conoce, si son resolubles en general, o sea, no existe el algoritmo que permite resolver estos problemas ni la demostración de su ausencia.

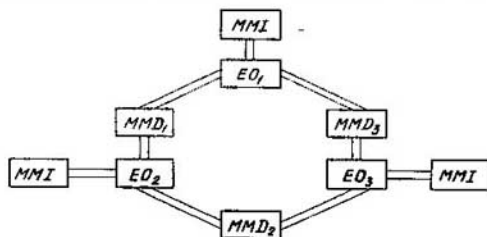


Fig. 4.48

Examinemos el uso de los métodos del análisis de las redes de Petri que simulan los sistemas prácticos.

Un sistema paralelo especializado para realizar los procesos iterativos de cálculo comprende una colección de elementos de procesador (EO) y de módulos de memoria (de memoria de datos (MMD) y de instrucciones (MMI) unidos en el anillo (fig. 4.48). El EO funciona en dos regímenes. En el primer caso ocupa los dos MMD adyacentes usando el izquierdo para sacar los datos de partida para nueva iteración y el derecho para elegir los resultados de la iteración anterior. Terminada la iteración, coloca el resultado en el MMD derecho y libera los dos MMD. En el otro régimen el EO funciona con los datos interiores. Una instrucción leída del MMI indica el régimen. Consideremos dos variantes para realizar unidades de mando del EO. En la primera variante, al realizar la iteración, el MMD se ocupa sucesivamente. El EO puede estar en los siguientes estados: «elaboración de los datos interiores» (S_1), «ocupado el MMD izquierdo» (S_2), «ocupado el MMD derecho» (S_3), «ocupados los dos MMD, elaboración de los datos de la iteración siguiente» (S_4). El MMD puede estar ora libre, ora ocupado. Consideremos la ocupación y la liberación del MMD como sucesos. Esta variante del funcionamiento se representa mediante la red de Petri, en la cual a cada EO le corresponden cuatro posiciones que realizan las condiciones descritas, y a cada MMD le corresponde una posición (en la cual la ficha significa que el MMD está libre) (fig. 4.49). Se puede cerciorarse

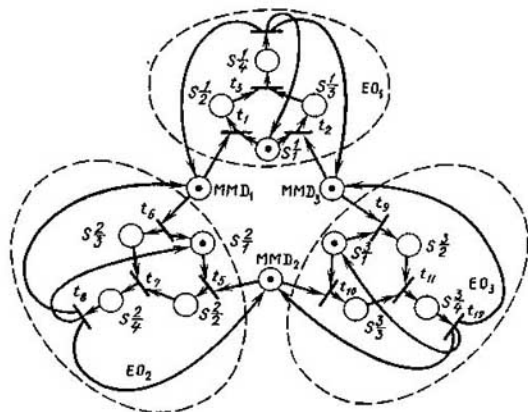


Fig. 4.49

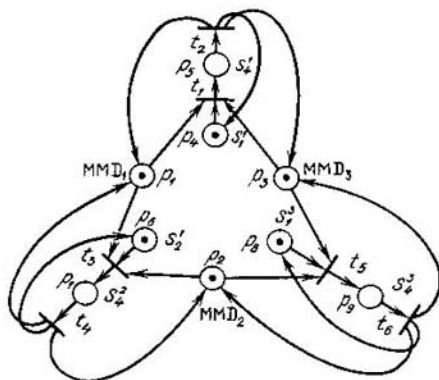


Fig. 4.50

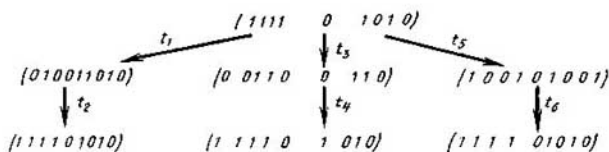


Fig. 4.51

que el árbol de alcance de esta red de Petri contiene dos marcaciones terminales: $\mu_1 = \{S_1^1, S_2^1, S_3^1\}$ y $\mu_2 = \{S_1^2, S_2^2, S_3^2\}$ que representan las situaciones de ocupar los MMD izquierdo y derecho por todo EO, respectivamente. Esto significa que la red de Petri no es activa, o sea, en el sistema, las situaciones tope son posibles.

En otra variante del funcionamiento del sistema del EO se realiza solamente la ocupación simultánea de los MMD adyacentes (si es posible). En este caso a todo EO le corresponden solamente las condiciones S_1 y S_4 (fig. 4.50). El árbol de alcance (fig. 4.51) no contiene marcaciones terminales, la red de Petri es activa. Además, de la consideración del árbol de alcance se desprende obviamente que la red de Petri es segura (las posiciones se interpretan como las condiciones simples). En el sistema se distribuyen los recursos que no aparecen ni desaparecen, es decir, se cumple la ley de conservación. Determinemos si la red de Petri es conservante. Para esto resolvamos la ecuación $DW = 0$ que toma la forma

$$\begin{bmatrix} -1 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \\ W_7 \\ W_8 \\ W_9 \end{bmatrix} = 0$$

Su resolución es $W = (1, 1, 1, 1, 3, 1, 3, 1, 3)$. En efecto, las posiciones p_5, p_7, p_9 son las condiciones vinculadas con tres dispositivos, las demás posiciones son condiciones vinculadas con un dispositivo. De este modo, la red de Petri posee las necesarias propiedades principales, lo que asegura la capacidad de trabajo de la segunda variante.

Las pruebas de simular sistemas reales condujeron a distintas definiciones más completas y modificaciones de las redes de Petri. En lo principal, estas modificaciones están ligadas con el cambio de la regla de arranque de transiciones.

La *potencia de simulación* de las redes habituales de Petri está limitada por la imposibilidad de probar las posiciones para el cero (o sea, de si la marcación de la posición es nula). Uno de los procedimientos para superar esta deficiencia es introducir *arcos retenedores*. Según las nuevas reglas de arranque, la transición está permitida, si las fichas están presentes en sus posiciones habituales de entrada (de las cuales salen los arcos habituales) y ausentes en las posiciones retenedoras de entrada (de las cuales salen los arcos retenedores). El arco retenedor se refleja como el habitual pero en su extremo tiene un círculo pequeño en vez de la flecha (esta designación ha sido adoptada de la teoría de los esquemas conmutadores donde el círculo significa «no») (fig. 4.52). En las redes habituales de Petri la transición arranca según la *lógica Y*, en las *redes de Petri con arcos retenedores* la lógica se extiende incluyendo las negaciones. Ya que se puede representar el suceso mediante unas transiciones se puede simular un suceso, cuya precondición se escribe como la unión de unas conjunciones de condiciones y las negaciones de condiciones que corresponden a las posiciones de la red de Petri con arcos retenedores. Por lo tanto, las redes de Petri permiten simular las precondiciones en forma de la FND, es decir, las condiciones de la forma más general.

El problema examinado de organizar el funcionamiento del sistema especializado de cálculo puede ser resuelto también por medio de la primera variante, en la cual se permite la ocupación sucesiva del MMD (véase la fig. 4.49) pero previniendo situaciones tope. Es obvio, que son posibles dos situaciones tope descritas mediante las marcaciones con fichas en las posiciones S_2^1, S_2^2, S_2^3 y S_2^4, S_2^5, S_2^6 (véase la fig. 4.49). Para evitar la primera situación tope, en las marcaciones $\{S_2^1, S_2^2\}, \{S_2^1, S_2^3\}, \{S_2^2, S_2^3\}$ es necesario evitar que aparezca la ficha

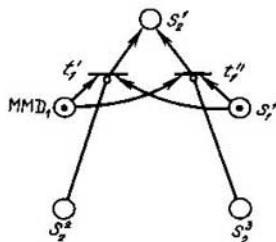


Fig. 4.52

en las posiciones S_2^1, S_2^2, S_2^3 , respectivamente. Por lo tanto, la transición t_1 debe tener en calidad de precondition la conjunción $MMD_1 \& S_1^1 \& (S_1^2 \& S_1^3)$, es decir, hace falta sustituirla por las transiciones t_1' y t_1'' con preconditiones $MMD_1 \& S_1^1 \& S_1^2$ y $MMD_1 \& S_1^1 \& S_1^3$ (fig. 4.52). Hay que tratar de modo análogo las transiciones t_5, t_9 (véase la fig. 4.49). Semejantes pasos se emprenden para evitar la segunda situación tope.

Otras proposiciones para cambiar las reglas de arranque o bien son equivalentes a la introducción de arcos retenedores o bien tienen carácter más particular. Por ejemplo, en las redes de Petri con campos de restricción se tienen conjuntos de posiciones (llamados campos de restricción), en las cuales las fichas no pueden encontrarse simultáneamente. Las reglas de arranque están modificadas de tal modo que no interrumpan esta condición. Si en la red de Petri dada en la fig. 4.49 incluimos dos campos de restricción $\{S_2^1, S_2^2, S_2^3\}$ y $\{S_3^1, S_3^2, S_3^3\}$, se puede evitar el surgimiento de las situaciones tope.

§ 4.9. Problemas y ejercicios

4.1. Formar una tabla funcional de la máquina de Turing cuando se suman 1 y el número escrito en el sistema ternario.

4.2. En la cinta está escrito un número en el sistema de numeración de la base Q . Formar las tablas funcionales, empleando las cuales se puede escribir el número; a) que sigue inmediatamente al dado; b) que precede inmediatamente al dado.

4.3. En la cinta está escrito un número x en el sistema ternario. Formar la tabla funcional, empleando la cual en la cinta se escribe $2x$ si x se divide por 3 sin resto y $x - 1$ en el caso contrario.

4.4. Escribir los números 704, 21, -77 en los sistemas de numeración con las bases $s = 3, 7, 11$. El conjunto de cifras es simétrico.

4.5. Escribir los números 0, 6; -56, 1 en los sistemas de numeración con las bases $s = 5, 7$. El conjunto de cifras es simétrico, el número de órdenes que se toma después de la coma es igual a tres para los dos sistemas.

4.6. Escribir los números 39, 88 y -101 en los sistemas de numeración con las bases $s = 4, 8, 12$. El conjunto de cifras es asimétrico en la dirección positiva.

4.7. Escribir los números 88, 41 en el sistema de numeración con la base 6 y con las cifras 1, 0, 1, 2, 3, 4.

4.8. Escribir el número -0,77 en el sistema de numeración con la base 10 y las cifras -3, -2, -1, 0, 1, 2, 3, 4, 5, 6. Tomar el número de órdenes después de la coma igual a cuatro.

4.9. En el sistema de numeración con la base 5 está escrito el número 22001. Hallar su equivalente decimal. ¿A qué número corresponderá este mismo código, si el sistema de numeración es nonario?

4.10. Establecer en qué sistema de numeración se realizó la siguiente actuación $(23 - 5) + (1 - 642) = 42423$. Se desconocen las cifras sustituidas por guiones.

4.11. Escribir los números 71, -16 y 203 en el sistema de numeración con la base -2. ¿A qué es igual el equivalente decimal del código obtenido, si lo consideramos como la notación en el sistema binario? El conjunto de cifras es natural.

4.12. Demostrar el siguiente teorema: en el sistema de numeración con la base natural, el conjunto natural de cifras permite codificar unívocamente cualquier equivalente cuantitativo.

4.13. Argumentar las reglas de traslación de números enteros y fraccionarios del sistema con la base natural R al sistema con la base natural Q .

4.14. Establecer las reglas de trasladar los números del sistema decimal de numeración al sistema de numeración con la base 2 y el conjunto natural de cifras.

4.15. Establecer las reglas de trasladar las denotaciones del sistema de numeración con la base -3 y el conjunto natural de cifras al sistema de numeración con la base 3 y el mismo conjunto de cifras.

4.16. Determinar cómo cambia la denotación del número al pasar del sistema con $S = 3$ y el conjunto simétrico de cifras al sistema con $S = -3$ y el mismo conjunto de cifras.

4.17. Establecer las reglas de trasladar del sistema de numeración con la base 2 y las cifras 0, 1 al sistema de numeración con la base 4 y el conjunto asimétrico de cifras desplazado a la dirección negativa.

4.18. Escribir el número 61 en el sistema de numeración con la base 0,25. Establecer la relación entre la denotación de los números en este sistema y en el sistema cuaternario con las cifras 0, 1, 2, 3.

4.19. Para el sistema de numeración con la base 0,125, hallar el conjunto de cifras que asegura la representación unívoca de cualquier número.

4.20. Escribir los números 60, -15 , -607 en el código complementario en el sistema de numeración con la base 5 y el conjunto natural de cifras.

4.21. Escribir el cero en el código complementario en el sistema de numeración con la base 6 y el conjunto natural de cifras.

4.22. Escribir los números 55, -70 , 118 en el código inverso en el sistema de numeración con la base 7 y el conjunto natural de cifras.

4.23. Escribir el cero en el código inverso en el sistema de numeración con $s = 11$ y el conjunto natural de cifras.

4.24. Escribir el número -65 en el código complementario en el sistema de numeración con $s = 4$ y las cifras $+3$, -2 , -1 , 0.

4.25. Establecer la relación entre los códigos complementario e inverso de un número x en el sistema con la base natural S y el conjunto natural de cifras.

4.26. Escribir los siguientes números: 75,5, $-0,25$; 0,125; -1000 en la forma semilogarítmica en el sistema binario de numeración con las cifras 0, 1. Según la condición la mantisa debe ser normalizada.

4.27. Establecer para que números x (no se supone que x es menor que el uno) el código complementario de estos números coincide con la denotación del propio número. Resolver el problema análogo para el código inverso (por supuesto, $x < 0$).

4.28. Trasladar el código complementario 6.1124 en la denotación habitual. Resolver el problema análogo para el código inverso 7.770045.

4.29. Hallar la suma de dos números 0,1101 y 0,0010 en el sistema binario con las cifras 0, 1 construyendo de antemano las reglas de adición en un orden. Los órdenes de los números son iguales.

4.30. Hallar el producto de los números 0,3302 y 0,1102 en el sistema cuaternario con el conjunto natural de cifras. Construir con anticipación la tabla de multiplicación para este sistema.

4.31. Hallar la suma y el producto de los números 0,210DD0 y 0,000009 en el sistema de numeración con $S = 11$ y el conjunto natural de cifras: $\{0, 1, 2, \dots, 9, D\}$, donde el equivalente cuantitativo de la cifra D es igual a 10. La denotación convencional de la cifra con el equivalente cuantitativo 10 es D . Construir con anticipación las tablas de adición y multiplicación en el sistema undecimal de numeración.

4.32. Empleando el código complementario, hallar la suma de los números 0,1101 y $-0,1000$ en el sistema binario con las cifras 0, 1. Trasladar el resultado de sumación a la denotación habitual.

4.33. Empleando el código inverso, hallar la suma de los números 0,2210 y $+0,1122$ en el sistema ternario con las cifras 0, 1, 2. Trasladar el resultado a la denotación habitual.

4.34. Hallar la suma de los números $-0,0011$ y 0,1001 en el código inverso en el sistema binario de numeración con las cifras 0, 1.

- 4.35. Hallar la suma de los números $0,00065$ y $-0,01125$ en el código complementario en el sistema septenario de numeración.
- 4.36. Hallar el producto de los números $3^1 \cdot 0,201$ y $-3^5 \cdot 0,102$ en el sistema ternario de numeración con las cifras $0, 1, 2$. Trasladar el resultado a la denotación habitual.
- 4.37. Hallar la suma de los números $-7^{-1} + 0,11066$ y $-7^0 + 0,11055$ en el código complementario en el sistema de numeración con $S = 7$ y el conjunto natural de cifras. Normalizar el resultado.
- 4.38. Hallar la suma de los números binarios $-2^1 \cdot 0,1001$ y $2^0 \cdot 0,1102$ en el código inverso en el sistema con las cifras $0, 1$. Normalizar el resultado.
- 4.39. Hallar el cociente para los números $0,0111$ y $0,1100$ limitándose con cuatro cifras del cociente. Trasladar el resultado a la denotación decimal.
- 4.40. Hallar cuatro cifras del cociente para los números $0,0442$ y $-0,4343$ en el sistema de numeración con la base 5 y el conjunto natural de cifras. Trasladar el resultado a la denotación decimal.
- 4.41. Hallar el cociente de la división de $4 \cdot 0,1101$ por $4^2 \cdot 0,3301$. Buscando la mantisa del cociente, calcular cinco cifras. Trasladar el resultado a la denotación binaria con las cifras $0, 1$.
- 4.42. Hallar el cociente de la división de $2^1 \cdot 0,1101$ por $-2^5 \cdot 0,1001$. El resultado de la división de la mantisa debe contener cinco órdenes y ser normalizado.
- 4.43. Determinar las reglas de adición y multiplicación para un orden en el sistema ternario con el conjunto simétrico de cifras. Empleando las reglas obtenidas, hallar la suma y el producto de los números $0,110\bar{1}$ y $0,110\bar{1}$. En esta denotación $\bar{1}$ corresponde a la cifra -1 .
- 4.44. Establecer las reglas de adición y multiplicación en el sistema de numeración con la base 6 y el conjunto asimétrico de cifras desplazado a la dirección negativa. Hallar la suma y el producto de los números $-0,3012$ y $0,1102$. La raya sobre la cifra significa que la última es negativa.
- 4.45. Determinar las reglas de adición en el sistema binario de numeración con las cifras $-1, 0, 1$. Hallar la suma de los números $0,110\bar{1}$ y $0,110\bar{1}$.
- 4.46. ¿Con cuántos procedimientos se puede codificar diez cifras decimales mediante las tétradas binarias?
- 4.47. ¿Con cuántos procedimientos se puede realizar la codificación del problema anterior, si, además, exigimos la correspondencia biunívoca entre las cifras y las tétradas?
- 4.48. Hallar la suma de los números 5764 y 2433 en el código del desplazamiento directo y en el código por exceso de 3 .
- 4.49. Hallar la suma de los números -79 y -981 en el código por exceso de 3 .
- 4.50. Establecer las reglas de adición en el código por exceso de 6 . Empleándolas adionar los números 203 y 479 .
- 4.51. Demostrar la afirmación: si en la codificación bidecimal todos los pesos p_i son no negativos, entonces son estrictamente positivos.
- 4.52. Demostrar que el único código perfecto es el código con los pesos $2^4 2^2 1$.
- 4.53. Demostrar la afirmación: si en el código de valor ponderado existen dos pesos iguales, su suma no supera 9 (se supone que los pesos son positivos).
- 4.54. Demostrar la afirmación: el código de pesos positivos no puede tener peso mayor que 8 .
- 4.55. Establecer las reglas de adición y multiplicación para el código por exceso de 5 . Determinar la suma y el producto de los números 87 y 56 .
- 4.56. Escribir las tablas de códigos por exceso de 1 a 15 . Cerciorarse que solamente los códigos por exceso de menor que 7 satisfacen las exigencias de unicidad, paridad y ordenación y, además, sólo el código por exceso de 3 satisface la complementariedad.
- 4.57. Establecer las reglas de adición y multiplicación para el código de Aiken Emeriax (el código con los pesos $2^4 2^2 1$). Empleando estas reglas, hallar la suma y el producto de los números -401 y 587 .

4.58. Establecer las reglas de adición para el código 2521 (es decir, para el código con los pesos 2 5 2 1). Hallar la suma de los números 90 y 73.

4.59. Determinar las reglas de adición para el código 3321 y hallar la suma de los números 601 y -670.

4.60. En el sistema de los módulos $q_1 = 2$, $q_2 = 3$, $q_3 = 5$, cierto número tiene código en restos tipo 010. Determinar este número.

4.61. En el sistema de los módulos $q_1 = 7$, $q_2 = 8$, se dan los números x e y por medio de sus códigos en restos 31 y 03. Establecer qué número es mayor, x o y .

4.62. Escribir el algoritmo de traslación de los códigos en restos a la denotación decimal del número.

4.63. Hallar un procedimiento para determinar el signo del número en el código en restos.

4.64. Hallar un procedimiento para comparar los números por el valor en el código en restos.

4.65. Establecer las reglas de división para el código en restos.

4.66. Sintetizar un esquema criotrónico que realiza la función $f(x_1, x_2, x_3, x_4) |_1 = \vee(1, 3, 7, 8, 9, 10, 12, 15)$.

4.67. ¿Cómo se tiene en cuenta el coeficiente de ramificación que se determina por la capacidad de carga del elemento básico dado cuando se utiliza la coálgebra de grafos K ?

4.68. Comparar las complejidades de los contadores de paridad de tres variables en las bases de Webb y de Sheffer.

4.69. Comparar la complejidad del contador de paridad de tres variables en la base de Sheffer construido según el método de simulación de cópulas del álgebra de Boole y según el método que se basa en la aplicación de la coálgebra de los grafos K .

4.70. Determinar la derivada temporal de la función booleana $x(t)$ dada en la tabla 4.9.

Tabla 4.9

t	0	1	2	3	4	5	6
$x(t)$	1	1	0	1	0	1	1

4.71. Hallar la función de error Δf en la salida de un semisumador completo, si el primer sumando $x_1(t)$, el segundo $x_2(t)$ y el traslado $p(t)$ del orden anterior se determinan según la tabla 4.10.

4.72. Hallar la función de error Δf a las salidas del descifrador de cuatro salidas, si a su primera entrada llega la sucesión booleana temporal $x(t)$ prefijada según la tabla 4.11, a y a su segunda entrada, la función $y(t)$ prefijada según la tabla 4.11, b .

Tabla 4.10

t	0	1	2	3	4	5	6	7
$x_1(t)$	0	1	1	1	0	0	1	0
$x_2(t)$	1	1	0	1	1	0	0	1
$p(t)$	0	0	1	0	1	1	1	0

Tabla 4.11, a

t	0	1	2	3	4	5	6	7
$y(t)$	1	1	1	0	0	1	0	1

Tabla 4.11, b

t	0	1	2	3	4	5	6	7
$y(t)$	1	0	0	1	1	0	0	0

4.73. Determinar la función de error a la salida del contador de imparidad que tiene tres entradas x_1 , x_2 , x_3 , a las cuales llegan las sucesiones booleanas temporales periódicas prefijadas en la tabla 4.12.

Tabla 4.12

t	0	1	2	3	4	5	6	7
$x_1(t)$	1	1	0	0	1	0	1	0
$x_2(t)$	0	0	0	0	1	1	0	1
$x_3(t)$	1	0	1	0	1	1	0	1

Comentarios

Crear los sistemas de automatización de diseño, de producción automatizada flexible, las redes locales de cálculo, los sistemas intelectuales de conocimiento y resolver otros problemas es imposible sin la formalización, cuya base constituye la teoría de las gramáticas formales y de los dispositivos automáticos. En el desarrollo de esta teoría un gran aporte pertenece a los científicos soviéticos V.M. Glushkov, M.A. Gavrilov, V.A. Gorbátov, A.V. Kaliáev, V.G. Lázarev, P.P. Parjómenko, D.A. Pospélov, V.P. Chistov, E.A. Yakubaitis y otros.

Para los conocimientos más detallados de las gramáticas formales y los dispositivos automáticos se recomienda la literatura adicional indicada en el apartado de Bibliografía.

De la percepción viva al pensamiento abstracto, *de éste a la práctica*: tal es el camino dialéctico del conocimiento de la *verdad*, del conocimiento de la realidad objetiva.

V. I. Lenin

CAPÍTULO 5

Teoría aplicada de los algoritmos Análisis de caracterización

§ 5.1. Principios del análisis de caracterización.

Construcción de los algoritmos combinatorios

El rasgo característico de la revolución científico-técnica contemporáneo es el creciente papel de los cálculos de carácter combinatorio (de sondeo) en los problemas aplicados. El problema actual de la matemática discreta es la construcción de los algoritmos combinatorios, eficaces tanto por la capacidad de la memoria necesaria como por la acción rápida.

Se puede partir los problemas aplicados en los del análisis y los de la síntesis de los sistemas discretos. Por la resolución del problema del análisis se entiende la determinación del hecho que el modelo Ψ_a que representa el sistema discreto posee las propiedades demandadas. Resolviendo el problema de la síntesis, el modelo Ψ_a se transforma en el modelo Ψ_b para alcanzar el extremo de la funcional prefijada de calidad $\varphi(\Psi_b)$. En ambos casos se puede hablar sobre la equivalentización. En los problemas del análisis, según el modelo Ψ_a se construye su equivalente que revela las propiedades del modelo. En los problemas de la síntesis el modelo Ψ_a se hace equivalente al modelo Ψ_b que se sintetiza. Tanto el análisis como la síntesis se realizan empleando algoritmos combinatorios.

La clase primitiva de los algoritmos combinatorios la representan los FBI-algoritmos (FBI significa Fuerza Brutal e Ignorancia). Estos algoritmos no tienen ninguna «destreza», resuelven los problemas «a ciegas» realizando el sondeo completo de las transformaciones posibles. En este caso faltan las premisas teóricas, basándose en las cuales se podría proponer un algoritmo «refinado» de solución. En los algoritmos de esta clase se realiza la equivalentización sintáctica.

La equivalentización sintáctica llamada *transformación equivalente* corresponde al nivel de conocimientos para el cual se conoce el sistema completo de axiomas y que consiste en la construcción de la siguiente variante para los problemas del análisis y en la sustitución del modelo Ψ_a por Ψ_b para el problema de la síntesis basándose en uno u otro axioma o ley obtenida del sistema de axiomas. El árbol de soluciones para la equivalentización sintáctica está representado en la fig. 5.1, *a*.

Todo vértice pendiente del árbol corresponde a la resolución tope. Las propiedades típicas de este árbol son:

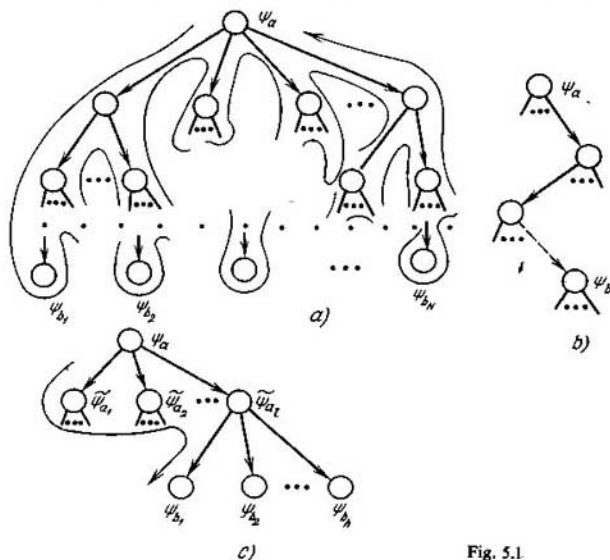


Fig. 5.1

1) el aumento combinatorio del número de vértices pendientes con el aumento lineal de la dimensión del problema;

2) la necesidad de «regresar» al $(i - 1)$ -ésimo nivel anterior cuando se calcula la información que corresponde al vértice «vecino» en el i -ésimo nivel. El corolario de esta propiedad es la necesidad principal de recorrer todo el árbol buscando la resolución mínima.

Estas propiedades determinan el fenómeno denominado la «maldición de la dimensión».

Durante la elaboración del apoyo matemático y de programas para los ordenadores, especialmente trabajando en escala concreta del tiempo, creando sistemas de automatización de proyección (el problema de SAPR), etc., lo actual es la proyección de paquetes de acción rápida de los programas aplicados. Para aumentar la rapidez de acción, se usan heurísticas en forma de correspondientes funcionales hallados en la base de experiencia, análogas y consideraciones sensatas. Como resultado se obtiene la clase de algoritmos heurísticos. Los algoritmos de esta clase realizan la equivalencia heurística, cuando la rapidez de acción del algoritmo incrementa, pero es imposible estimar la calidad de la solución obtenida; incluso no se puede decir si es de tipo tope, o sea, imposible de ser

simplificada más. En la fig. 5.1, b está representado el árbol de soluciones cada vértice del cual corresponde al trabajo dado, cada arco, a una variante del sondeo o a una transformación.

Los algoritmos de la tercera clase, que realizan la equivalentización semántica, permiten lograr la acción más rápida de los algoritmos combinatorios sin engendrar todas las soluciones equivalentes, o sea, simplificada más. En la fig. 5.1, b está representado el árbol de soluciones,

Durante la equivalentización semántica es conocido no sólo el sistema completo de axiomas, sino también el criterio constructivo que permite: para el problema del análisis comprobar la veracidad del predicado $P_0(\Psi_a)$:

$$P_0(\Psi_a) = \begin{cases} 1, & \text{si el modelo } \Psi_a \text{ posee la propiedad prefijada,} \\ 0 & \text{en caso contrario;} \end{cases}$$

para el problema de la síntesis vincular dos abstracciones distintas, los modelos Ψ_a y Ψ_b , en un sistema único empleando el predicado de la integridad funcional $P_0(\Psi_a, \Psi_b)$:

$$P_0(\Psi_a, \Psi_b) = \begin{cases} 1, & \text{si la transformación } \Psi_a \rightarrow \Psi_b \text{ existe junto} \\ & \text{a la correspondencia biunívoca entre los} \\ & \text{elementos de los modelos } \Psi_a \text{ y } \Psi_b, \\ 0 & \text{en caso contrario.} \end{cases}$$

Lo común que caracteriza los modelos Ψ_a y Ψ_b y los diferencia de los demás es el sentido de la transformación $\Psi_a \rightarrow \Psi_b$. En otras palabras el sentido de la transformación $\Psi_a \rightarrow \Psi_b$ es la propiedad de las expresiones lingüísticas, invariante respecto a sus representaciones modelos. La lógica semántica, parte de la metalógica, investiga este sentido. Se comprende como semántica, por regla general, la semántica descriptiva que examina la conexión entre las combinaciones de signos del lenguaje formalizado y sus interpretaciones en los términos del sistema de conceptos, cuya formalización es el lenguaje dado. Aquí se comprende como semántica el estudio de la interpretación de un lenguaje formalizado en las categorías de otro con tal que los dos lenguajes son formalización de un sistema de conceptos. Llamaremos *proyectivo* a este tipo de semántica. En caso particular, cuando dos lenguajes coinciden y se investiga si se puede cumplir la propiedad determinada del modelo de estudio, la semántica proyectiva se denominará *reflexiva*.

La semántica reflexiva permite resolver los problemas del análisis de los modelos. La semántica proyectiva de la transformación $\Psi_a \rightarrow \Psi_b$ permite calcular el valor extremal de la funcional de calidad $\varphi(\Psi_b)$ de la resolución y construir el correspondiente modelo óptimo Ψ_b sin formar todos los modelos equivalentes $\{\Psi_b\}$ que disminuye considerablemente la densidad de trabajo de los algoritmos.

Para determinar la semántica proyectiva de la transformación $\Psi_a \rightarrow \Psi_b$ es necesario:

1. Hallar las características numéricas $\{\nu_i\}$ del modelo Ψ_b que determinan unívocamente el valor $\varphi(\Psi_b)$.
2. Determinar las propiedades S_b del modelo Ψ_b , en cuya presencia se puede calcular $\{\nu_i\}$.
3. Revelar las propiedades S_a del modelo Ψ_a que determinan unívocamente las propiedades S_b del modelo Ψ_b .
4. Hallar las características numéricas del modelo Ψ_a que posee las propiedades S_a que determinan unívocamente $\varphi(\Psi_b)$.

De este modo, la presencia de las propiedades S_a permite calcular unívocamente $\varphi(\Psi_b)$ sin construir de hecho Ψ_b .

Para hallar la semántica reflexiva del análisis del modelo Ψ_a es necesario:

1. Revelar las propiedades S_a del modelo Ψ_a que determinan unívocamente el predicado $P_0(\Psi_a)$.
2. Hallar las características numéricas del modelo Ψ_a que determinan la presencia de las propiedades S_a .

En ambos casos, lo principal es establecer las propiedades S_a del modelo Ψ_a que determinan la veracidad del predicado $P_0(\Psi_a)$ o $P_0(\Psi_a, \Psi_b)$. Estas propiedades del modelo Ψ_a las buscaremos en forma de propiedades de ausencia de las figuras prohibidas que forman la base del criterio de cumplimiento de las propiedades S_a . Si conocemos las figuras prohibidas, podemos resolver eficazmente los problemas del análisis y calcular de manera constructiva la funcional $\varphi(\Psi_b)$ sin engendrar todos los modelos equivalentes $\{\Psi_b\}$ (véase la fig. 5.1, c) durante la solución de los problemas de la síntesis. La búsqueda y el estudio de las semánticas reflexivas y proyectivas basadas en figuras prohibidas los referiremos a la metalógica y denominaremos *semántica constructiva*.

La figura 5.2 ilustra la relación entre las tres semánticas de transformación.

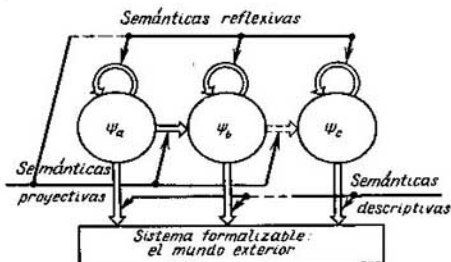


Fig. 5.2

El problema de buscar las figuras prohibidas se denomina *problema de caracterización*. Este problema se define por la clase de modelos $K_a = \{\Psi_a\}$ que se examinan y por la propiedad S_a que se caracteriza y que determina el predicado $P_0(\Psi_a)$ o $P_0(\Psi_a, \Psi_b)$. Para resolver el problema de caracterización hay que determinar el conjunto de figuras prohibidas $K_p = \{\Psi_i\}$, es decir, de tales modelos Ψ_i , cuya ausencia en el modelo dado $\Psi_a \in K_a$ es la condición necesaria y suficiente de que Ψ_a posee la propiedad S_a con ello ninguno de los modelos $\Psi_i \in K_p$ está presente en otra figura prohibida: un modelo $\Psi_j \in K_p$.

Formalicemos los conceptos de *ausencia* y de *presencia* de un modelo en otro. Sobre el conjunto de modelos K_a se puede fijar una relación de ordenación P_s , tal que $(\Psi_i, \Psi_j) \in P_s$, si Ψ_i está presente en Ψ_j . La relación P_s se denomina *relación de subordinación*, el modelo Ψ_i se llama *subordinado* al modelo Ψ_j . La relación de subordinación de modelos es la generalización de la conocida relación de *ser submodelo*. El modelo Ψ_i es el submodelo Ψ_j obtenido después de eliminar unos elementos del portador y de la signatura del modelo Ψ_j . El problema de caracterización con la relación dada de subordinación P_s se hace concreta sobre una clase de modelos, transformándose en la tarea de caracterización. Puesto que sobre la clase de modelos K_a pueden darse muchas relaciones de ordenación, el problema de caracterización puede considerarse como todo un conjunto de tareas de caracterización, cada una de las cuales tiene su propia solución en forma de un conjunto de figuras prohibidas.

Resolviendo la tarea de caracterización muchas cosas dependen de la opción de la relación de subordinación: la compacidad del conjunto de figuras prohibidas y la resolubilidad de esta tarea en general. El hecho de que el problema de caracterización es siempre resoluble tiene carácter de principio. Hace falta sólo escoger correctamente la relación de subordinación y obtener, conforme a esto, el planteamiento del problema resoluble de caracterización. Mostremos cuál debe ser la relación de subordinación.

Teorema 5.1 (principio de localidad). *Para una clase de modelos $K_a = \{\Psi_a\}$ con una relación de subordinación P_s y para una propiedad S_a , el conjunto de figuras prohibidas $K_p = \{\Psi_p\}$ existe (la tarea de caracterización es resoluble) si, y sólo si, es válido que todo modelo Ψ_i subordinado a un modelo Ψ_j con la propiedad S_a posee también esta propiedad.*

□ Supongamos que el conjunto de figuras prohibidas existe. Entonces, según la definición de figura prohibida ningún modelo Ψ_a que posee la propiedad S_a tiene figuras prohibidas como las subordinadas.

Por otra parte, sea que la relación de subordinación es tal que a los modelos Ψ_a con la propiedad S_a inherente les están subordinadas sólo los modelos que también la tienen (fig. 5.3). En este caso, según la definición de figura prohibida el conjunto de figuras prohibidas K_p se forma de los

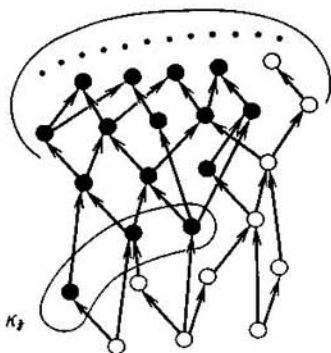


Fig. 5.3

elementos minimales de la relación de ordenación P_s sobre el conjunto $K_a \setminus \tilde{K}_a$, donde $\tilde{K}_a \subset K_a$ es la subclase de modelos Ψ_a que poseen la propiedad S_a . ■

En calidad del ejemplo examinemos el problema de caracterización de los grafos de dos partes. Las tareas de determinar dos partes del grafo y de transformar un grafo en el de dos partes tienen muchas aplicaciones: partiendo de la proyección de autómatas seguros hasta la construcción de sistemas informativos eficaces con el cese del procesamiento en paralelo. Por ejemplo, sea que un sistema informativo de biblioteca se realiza en el ordenador con dos dispositivos de discos (fig. 5.4, a). Para lograr la eficacia máxima de trabajo del sistema informativo de búsqueda es necesario descomponer la base de datos en dos partes de tal modo que se obtenga el tiempo mínimo de procesamiento de la demanda debido a la colocación simultánea de las cabezas de lectura e inscripción en dos dispositivos de disco. A los ficheros de la base de datos les ponemos en correspondencia los vértices del grafo. Dos vértices son adyacentes si los ficheros correspondientes son necesarios para responder a la demanda de cierto tipo (fig. 5.4, b). La distribución óptima de ficheros por los discos se determina por la partición del conjunto de los vértices del grafo en dos conjuntos, dentro de los cuales se encuentra el número mínimo posible de aristas (que corresponde al paralelismo máximo). Esta tarea se reduce a desentrañar la semántica proyectiva de transformación de los grafos en los de dos partes eliminando el número mínimo de aristas. La semántica se determina resolviendo el problema de caracterización de los grafos de dos partes (fig. 5.4, c).

La clase K_a se compone de los modelos, cuya signatura se forma por una relación binaria simétrica antirreflexiva. La propiedad S_a : ser de dos partes. Examinemos dos relaciones de subordinación: P_s^1 , ser grafo parcial y P_s^2 , ser grafo reducible. Llámase reducible un grafo G_a que se obtiene

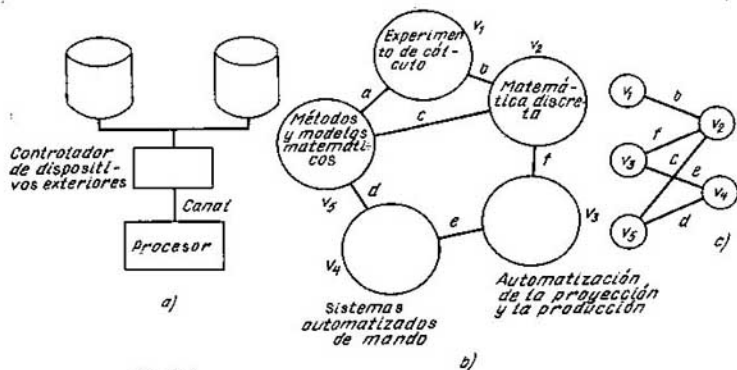


Fig. 5.4

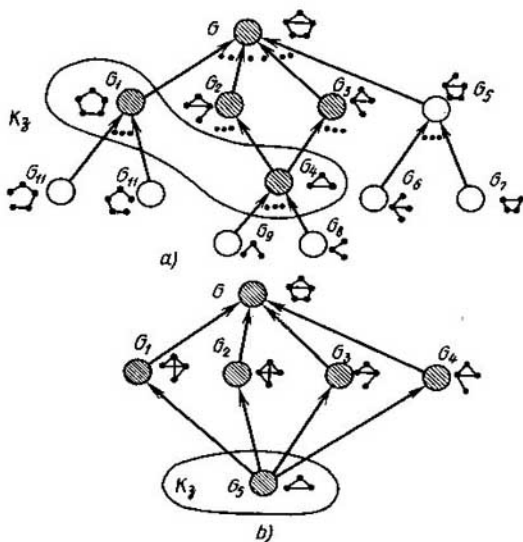


Fig. 5.5

de un grafo G_b después de encolar sucesivamente los vértices no adyacentes (uniéndolos en un vértice) junto a la unión correspondiente de sus entornos. Las dos relaciones satisfacen el principio de localidad. Por lo tanto, en ambos casos el problema de caracterización es resoluble. En la fig. 5.5, a y b se ofrecen los diagramas de las relaciones de subordinación. En el caso

de la relación de subordinación P_1^1 el conjunto de figuras prohibidas es el conjunto de los ciclos impares, en el caso de la relación de subordinación P_2^2 , el conjunto de grafos completos del orden más que dos.

Así, pues, siempre se puede resolver el problema de caracterización y, por consiguiente, obtener el conjunto de figuras prohibidas que determina el criterio semántico constructivo para solucionar los problemas del análisis. Es más, el conocimiento de los conjuntos de figuras prohibidas se utiliza en el método de la equivalentización semántica en los problemas de síntesis. El principio fundamental de la equivalentización semántica consiste en conocer los modelos prohibidos concretos que presencian en el modelo Ψ_a y le molestan poseer la propiedad S_a , lo que permite determinar las estructuras locales, cuya transformación es necesaria para obtener el modelo Ψ_b con la propiedad S_a inherente. Al mismo tiempo, se realiza solamente el sondeo mínimo (inevitable) de variantes de transformaciones, es decir, no se puede mejorar el proceso de cálculo en el sentido de la densidad de trabajo. Durante la equivalentización semántica el árbol de soluciones (véase la fig. 5.1, c) comprende dos estrellas. La primera corresponde a la transformación $\Psi_a \rightarrow \tilde{\Psi}_a$, $P_0(\tilde{\Psi}_a, \Psi_a) = 1$, la segunda, a la $\tilde{\Psi}_a \rightarrow \Psi_b$. Buscando la solución minimal es necesario recorrer todas las ramas de la primera estrella y, en la segunda, es suficiente tomar cualquier rama, ya que todas las ramas de ésta última son equivalentes desde el punto de vista del carácter minimal de la solución que se determina por el valor $\varphi(\Psi_b)$.

Analicemos detalladamente el proceso de la equivalentización semántica. En la base de éste están los procedimientos de transformar las figuras prohibidas en las equivalentes permitidas. El sentido de la transformación $\Psi_a \rightarrow \Psi_b$ determina la equivalencia. Como regla, el procedimiento de transformar una figura prohibida en la permitida es la eliminación, la introducción o la desintegración del elemento del portador o de la signatura o bien el paso a un modelo subordinado.

Durante la equivalentización semántica para transformar un grafo en el de dos partes son posibles varios procedimientos de transformación de las figuras prohibidas, o sea, ciclos de longitud impar. Empleando la transformación para encajar el grafo en el hipercubo durante la proyección de autómatas seguros, la figura prohibida se transforma en la permitida por medio de introducir un número impar de vértices para una arista (estrictamente dicho, aquí tenemos el conjunto de transformaciones y no una sola). Si la transformación se emplea para la descomposición óptima de la base de datos, el procedimiento de transformación consiste en eliminar una arista. Es de principio que el procedimiento de transformación de la figura prohibida en la permitida existe siempre cuando la transformación $\Psi_a \rightarrow \Psi_b$ tiene sentido en general. En efecto, para todo modelo Ψ_a existe el modelo equivalente $\tilde{\Psi}_a$ que posee la propiedad S_a . Por tanto, existe la transformación de Ψ_a en $\tilde{\Psi}_a$ y cualquier transformación del modelo Ψ_a en el modelo

que posee la propiedad S_a transforma obligatoriamente las figuras prohibidas en las permitidas. La localización de la transformación global de Ψ_a en $\bar{\Psi}_a$ sobre la figura prohibida $\Psi_p \subset \Psi_a$ incluye su transformación en la permitida. De este modo, el procedimiento de la transformación de la figura prohibida en la permitida existe siempre.

En el caso general es posible el conjunto de transformaciones de la figura prohibida en la permitida. Sea $R_i = \{r_i\}$ un conjunto de procedimientos de transformar las figuras prohibidas $\Psi_i \in \mathbf{K}_p$ (es decir, para cualquier j $r_{ij}(\Psi_i)$ es figura permitida). Construyamos el conjunto de procedimientos básicos de transformaciones $R_i^0 \subset R_i$, es decir, un conjunto minimal por la inclusión $\{r_{ij}^0\}$ tal que para cualquier r_i , existe una sucesión de transformaciones de R_i^0 que traspasa Ψ_i a $r_i(\Psi_i)$.

Examinemos los procedimientos de transformar las figuras prohibidas para transformar el mografo en el lineal lo que es importante en los problemas de organización de los datos en los sistemas de búsqueda informativa y en las bases de datos. Las características principales del emplazamiento de los datos en la memoria son la capacidad de memoria que se ocupa y el tiempo de acceso. Cuando el sistema de búsqueda informativa se prefija como un mografo la capacidad de memoria se determina por la potencia del portador, y el tiempo de acceso se determina por el tiempo de lectura de las palabras del modelo. Uno de los procedimientos de la organización óptima de los datos en la memoria es el emplazamiento lineal que supone una ordenación lineal (completa) de los objetos de los datos tal que la respuesta a cada demanda es una cadena de datos así ordenados. Pongamos los elementos del portador en correspondencia a los objetos de datos, las palabras, en correspondencia a las respuestas a las demandas. Un mografo se denomina *lineal* si permite el emplazamiento lineal de los elementos del portador, en el cual todas las palabras son cadenas. De este modo, para obtener la organización óptima de los datos, es necesario transformar el mografo en el lineal y, por lo tanto, resolver el problema de caracterización de la linealidad del mografo.

Sin concretizar la forma de figuras prohibidas (lo que examinemos más detalladamente a continuación) señalemos que son posibles dos procedimientos de transformar las figuras prohibidas en las permitidas:

- 1) la desintegración del elemento x del portador en x y x' con la correspondiente sustitución de x por x' en unas palabras que comprendían x ;
- 2) la desintegración de la palabra M , es decir, su sustitución por dos palabras.

Se puede mostrar que estos procedimientos forman un conjunto básico de procedimientos de transformar las figuras prohibidas en las permitidas.

Ejemplifiquemos los procedimientos de transformar las figuras prohibidas en las permitidas empleando el mografo (fig. 5.6, a) que es un sistema informativo hipotético de biblioteca que incluye la información sobre los siguientes libros:

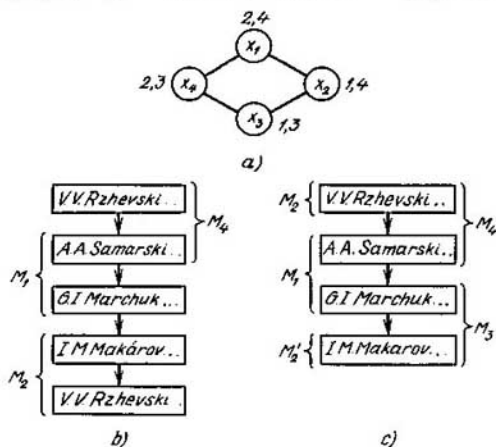


Fig. 5.6

x_1 : V.V.Rzhevski, Procesos de trabajo de la minería a cielo, M., Nedra, 1978;

x_2 : A.A.Samarski, Teoría de los esquemas en diferencias, M., Naúka, 1977;

x_3 : G.I.Marchuk, Métodos de la matemática de cálculos, M., Naúka, 1977;

x_4 : Fundamentos de la automatización del control de la producción (Dirigido por I.M.Makárov, M., Vísshaya shkola, 1983);

el sistema se destina para las siguientes demandas:

M_1 : Libros de los métodos de cálculo. Respuesta $\{x_2, x_3\}$;

M_2 : Libros de automatización de los procesos. Respuesta $\{x_1, x_4\}$;

M_3 : Libros de autores (redactores), cuyos apellidos empiezan con A a M. Respuesta $\{x_3, x_4\}$;

M_4 : Libros de autores (redactores), cuyos apellidos empiezan con N a Z. Respuesta $\{x_1, x_2\}$.

El mografo dado no es lineal. Se puede mostrar que todo elemento del portador y toda palabra integran la figura prohibida. La desintegración del elemento del portador o de la palabra debe transformar el mografo en el lineal. Por ejemplo, consideremos la desintegración del elemento x_1 . El mografo transformado es lineal y se representa por la ordenación lineal dada en la fig. 5.6, b. Desintegrando cualquier palabra, por ejemplo M_2 , el mografo se hace también lineal y se representa por la ordenación lineal dada en la fig. 5.6, c. En el primer caso, aumenta la capacidad de memoria ocupada por los objetos de datos, en el segundo, el promedio del tiempo de respuesta a las demandas.

Cada procedimiento de la transformación r_i se caracteriza por el valor c_i . La funcional de la calidad de transformación $\varphi(\Psi_b)$ se determina por el valor de las transformaciones de las figuras prohibidas en las permitidas.

Para cumplir la transformación $\Psi_a \rightarrow \tilde{\Psi}_a$ es inevitable transformar cada figura prohibida en la permitida. Por eso, para toda figura prohibida $\Psi_p \subset \Psi_a$, escojamos uno de los procedimientos de su transformación; su colección determinará la transformación global $\Psi_a \rightarrow \tilde{\Psi}_a$. En el caso general,

esta transformación posiblemente no logra el extremo de la funcional $\varphi(\Psi_b)$, incluso si para cualquier Ψ_p , está elegido el procedimiento de transformación con el valor mínimo, lo que está condicionado por el posible procedimiento de transformación Ψ_p , que, pese a no tener el valor mínimo, transforma al mismo tiempo otra figura prohibida Ψ_p . Además, la relación entre los procedimientos de transformación de las figuras prohibidas puede resultar muy complicada. Por eso la simple elección de procedimientos de la transformación, uno por cada $\Psi_p \subset \Psi_a$, no asegura el logro del extremo de la funcional de la calidad $\varphi(\Psi_b)$.

Está propuesto el procedimiento de la equivalentización semántica basada en la construcción de la tabla semántica. A las columnas de esta tabla les corresponden figuras prohibidas presentes en el modelo; a las filas, procedimientos de la transformación. El elemento (i, j) de la tabla es igual a 1, si la i -ésima transformación convierte la j -ésima figura prohibida en la permitida y a 0 en caso contrario. El cubrimiento de las columnas por las filas de la tabla semántica determina un conjunto de transformaciones, minimal por la inclusión. Es necesario cumplirlo para obtener del modelo Ψ_a el $\tilde{\Psi}_a$ con la propiedad S_a . Tomando en consideración que todo procedimiento de transformación puede tener valor propio, deducimos que para llegar al extremo $\varphi(\Psi_b)$ es necesario hallar cubrimiento de la tabla semántica, minimal por su valor. A veces, en esta tabla, a las filas les corresponden figuras prohibidas, a las columnas, sus transformaciones en las permitidas. Ya que esto no es de principio, utilizaremos las dos variantes de la construcción de la tabla semántica.

Consideremos el problema de la equivalentización semántica del grafo, representado en la fig. 5.4, *b*, en un grafo de dos partes. La funcional de la calidad es el mínimo de aristas eliminadas. Las figuras prohibidas (ciclos de longitud impar) están formadas por los siguientes conjuntos de aristas: $\Psi_{p_1} = \{a, b, c\}$, $\Psi_{p_2} = \{a, b, d, e, f\}$. El procedimiento de la transformación -eliminación de la arista- tiene el valor 1; el procedimiento concreto de la transformación se designa indicando la arista para eliminar. La tabla semántica tiene siguiente forma:

Tabla 5.1

Ψ_{p_1}	Ψ_{p_2}	
1	1	<i>a</i>
1	1	<i>b</i>
1		<i>c</i>
	1	<i>d</i>
	1	<i>e</i>
	1	<i>f</i>

El cubrimiento minimal es, por ejemplo, $\pi = \{a\}$. Por consiguiente, eliminando la arista a obtenemos un grafo de dos partes (fig. 5.4, c); además, alcanzamos el mínimo $\varphi(\Psi_b) = 1$.

En el caso general, el procedimiento de la transformación puede vincularse no con los elementos del portador o de la signatura del modelo, sino con algunos de sus componentes. Por eso, en el caso general, la equivalentización semántica supone la construcción del sistema jerárquico de las tablas de una profundidad k , determinada por el número de niveles en los procedimientos de transformaciones. El cubrimiento de las columnas por las filas de la primera tabla indica qué componentes de las figuras prohibidas deben cambiarse cuando el modelo Ψ_a se reduce a la forma que se interpreta en los términos del modelo Ψ_b . La determinación de los componentes que deben ser cambiados para cambiar los componentes hallados al paso anterior se reduce al cubrimiento de la segunda tabla construida análogamente, etc., hasta construir la k -ésima tabla, cuyas filas o columnas corresponden a los elementos del portador o de la signatura que deben ser cambiados sin falta cuando el modelo Ψ_a se reduce a la forma interpretada en los términos del modelo Ψ_b (fig. 5.7).

Para determinar el número mínimo de los elementos del portador o de la signatura que corresponden a los elementos de la k -ésima tabla es necesario generar todos los conjuntos, cada uno de los cuales comprende las filas (columnas) que se cubren en la última, k -ésima tabla. Esto corresponde al sondeo de todos los cubrimientos de la $(k - 1)$ -ésima tabla. Para obtener todos los cubrimientos de la $(k - 1)$ -ésima tabla es necesario generar todos los cubrimientos de la $(k - 2)$ -ésima tabla, etc. De este modo, para hallar la solución minimal, es necesario el sondeo de todas las combinaciones de los cubrimientos de las primeras $(k - 1)$ tablas. Este procedimiento contiene, en principio, el sondeo.

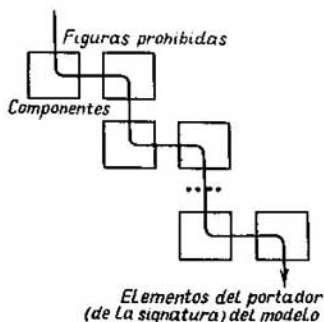


Fig. 5.7

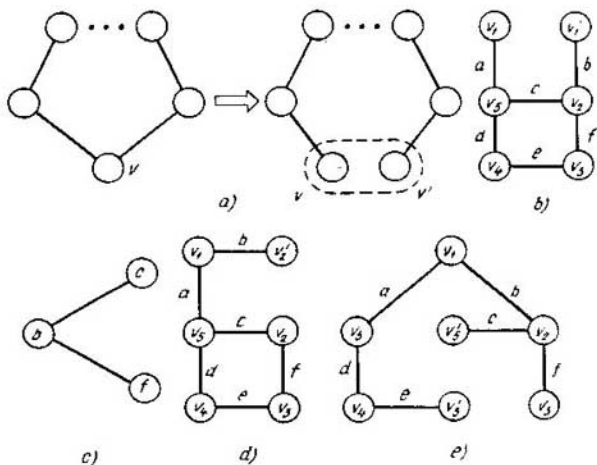


Fig. 5.8

De tal manera, la determinación de la complejidad de solución es un proceso estándar, minimal por su densidad de trabajo. Este proceso requiere el trabajo en varios órdenes menos que el de la generación efectiva de todas las estructuras equivalentes durante la búsqueda de la solución minimal por medio de la equivalentización sintáctica.

Los procedimientos de transformar las figuras prohibidas, siendo unívocos respecto a ellas, pueden ser en total tanto unívocos como no respecto al modelo. Por ejemplo, para el carácter de dos partes del grafo, el procedimiento de transformar figuras prohibidas, ciclos impares, basado en la eliminación de la arista es también unívoco para el modelo en total. Consideremos el procedimiento de transformar ciclos impares que consiste en la desintegración de uno de los vértices v y v' , cada uno de los cuales es incidente a una de dos aristas x e y del vértice que se desintegra (fig. 5.8, a) (designemos el procedimiento de transformación por $v(x, y)$). Este procedimiento de transformación se usa en la equivalentización semántica cuando se descompone el sistema informativo exigiendo la acción más rápida, es decir, el paralelismo máximo del procesamiento de la información. Esta exigencia condiciona la colocación de todas las partes de los datos correspondientes a los vértices adyacentes en los discos diferentes, es decir, lleva a duplicar algunas partes. Este procedimiento de transformar figuras prohibidas no es unívoco para el modelo en total (para el grafo). Es que durante la desintegración del vértice v , incidente a las aristas x e y , la transforma-

ción de la figura prohibida fija solamente, lo que el nuevo vértice v es incidente a la arista x y el nuevo vértice v' , a la arista y . Para el grafo en total, eso significa la desintegración del vértice v en v y v' con la correspondiente partición del entorno Γv cuando x e y se encuentran en distintos entornos nuevos. Este procedimiento de transformación no indica la distribución de todos los vértices por nuevos entornos. En este procedimiento de transformación, la tabla semántica para el grafo (véase la fig. 5.4, b) tiene siguiente forma:

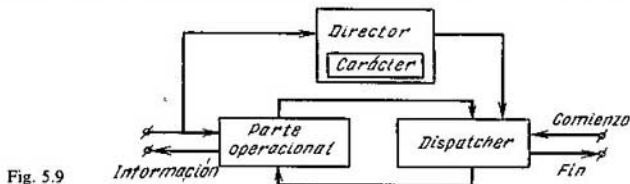
Tabla 5.2

Ψ_{p_1}	Ψ_{p_1}	
1	1	$v_1(a, b)$
1		$v_2(b, c)$
1		$v_3(a, c)$
	1	$v_2(b, f)$
	1	$v_3(e, f)$
	1	$v_4(d, e)$
	1	$v_5(a, d)$

En caso de procedimientos no unívocos de transformaciones es necesario comprobar todos los cubrimientos de la tabla semántica. Consideremos tres cubrimientos: $\pi = \{v_1(a, b)\}$, $\pi_2 = \{v_2(b, c), v_2(b, f)\}$, $\pi_3 = \{v_3(a, c), v_3(e, f)\}$.

El primer cubrimiento con la potencia minimal da la solución minimal (fig. 5.8, b). Empero el segundo cubrimiento con la potencia no minimal también determina la solución minimal. Ambas transformaciones que componen π_2 desintegran v_2 . ¿Cuántos vértices nuevos dan estas desintegraciones juntas? Responden a esta pregunta la construcción y la coloración de un grafo especial construido sobre un conjunto de aristas, incidentes a v_2 (fig. 5.8, c). La coloración del grafo $\{b\}, \{c, f\}$ determina el número de vértices después de la desintegración y la partición del entorno Γv_2 . La solución obtenida es minimal (fig. 5.8, d). La transformación correspondiente a π_3 (fig. 5.8, e) no es minimal.

De este modo, si aplicamos los procedimientos de transformaciones no unívocas para el modelo en total, para todo cubrimiento es necesario cons-



truir grafos especiales y determinar su coloración minimal. Por eso son muy actuales estimaciones del número cromático del grafo (véase el capítulo 3) que permiten separar rápidamente la mayor parte de cubrimientos que corresponden a los grafos con gran número cromático. Son también importantes los métodos «rápidos» de la coloración de los grafos. En total, la equivalentización semántica permite obtener solución absolutamente óptima junto al sondeo minimal (inevitable).

Basándose en la equivalentización semántica obtenemos la estructura de dos circuitos del paquete de acción rápida de programas aplicados (fig. 5.9), en el cual, empleando el módulo «carácter», se realiza la sintonización automática de la estrategia óptima durante la transformación $\Psi_a \rightarrow \Psi_b$.

Esto atribuye «carácter intelectual» al paquete de programas aplicados y permite catalogar estos paquetes en la clase de sistemas del intelecto artificial.

En los capítulos anteriores fueron considerados los problemas de caracterización del encaje del grafo en el plano, en el espacio booleano, de caracterización de la estructura en serie-paralela de los diagramas de Hasse y de la coloración de los grafos. Examinemos ahora los problemas de caracterización de la ordenación parcial del mografo, de proyección de esquemas lógicos en las bases funcionales inconexas, de proyección de esquemas lógicos de salidas múltiples, de descomposición de los grafos de transiciones en el producto cartesiano parcial y, además, problemas de caracterización que surgen durante el diseño de emplazamientos óptimos de los datos en los sistemas informativos. La caracterización de los modelos permite revelar causas objetivas que determinan la complejidad de solución y la densidad de trabajo de su búsqueda.

§ 5.2. Caracterización de la ordenación parcial del mografo

Antes de establecer las causas que llevan a la correspondencia entre una terma primaria y dos (o más) vértices en un grafo estructural (la desintegración de las termas primarias) examinemos la transformación del mografo en el grafo estructural prefijando la FND tope de la función booleana

$$f(x_1, x_2, \dots, x_5) = \underbrace{x_1 \bar{x}_3 \bar{x}_5}_1 \vee \underbrace{x_1 x_4}_2 \vee \underbrace{x_2 x_3}_3 \vee \\ \vee \underbrace{\bar{x}_1 x_3 \bar{x}_5}_4 \vee \underbrace{x_3 \bar{x}_4 x_5}_5 \vee \underbrace{x_2 x_4 x_5}_6,$$

determinada por el modelo $\Psi_a = \langle M, S_2, S_3 \rangle$;

$$M = \{x_1, \bar{x}_1, x_2, x_3, \bar{x}_3, x_4, \bar{x}_4, x_5, \bar{x}_5\},$$

$$S_2 = \{\{x_1, x_4\}, \{x_2, x_3\}\},$$

$$S_3 = \{\{x_1, \bar{x}_3, \bar{x}_5\}, \{\bar{x}_1, x_3, \bar{x}_5\}, \{x_3, \bar{x}_4, x_5\}, \{x_2, x_4, x_5\}\}.$$

El mografo $G^M(\Psi_a)$ se representa en la fig. 5.10, a.

Examinemos el submografo $(G^M)'$ (fig. 5.10, b) que da la tercera, la quinta y la sexta implicante simples. A la tercera implicante $x_2 x_3$ le ponemos en correspondencia la cadena $v(x_2) \leq v(x_3)$. A la quinta implicante simple $x_3 \bar{x}_4 x_5$ le ponemos en correspondencia la cadena $v(x_3) \leq v(\bar{x}_4) \leq v(x_5)$, a la sexta implicante $x_2 x_4 x_5$, la cadena $v(x_2) \leq v(x_4) \leq v(x_5)$. Prefijando así las relaciones de ordenación, obtenemos que $v(x_2) \leq v(x_3) \leq v(\bar{x}_4)$, o sea, $v(x_2)$ es comparable con $v(\bar{x}_4)$, $v(x_2) \not\geq v(\bar{x}_4)$ que contradice a la prefijación.

Para responder si tal prefijación de la relación \leq es simplemente infortunada, sin saber la semántica de transformación, es imprescindible construir completamente el árbol sintáctico de esta transformación, a cuyos vértices pendientes les corresponden diagramas H ; además se consideran no sólo estas tres implicantes sencillas, sino todo el mografo prefijado $G^M(\Psi_a)$. El número de vértices pendientes de este árbol es igual a $2! \cdot 2! \cdot 3! \cdot 3! \cdot 3! \cdot 3! = 5184$. Construyendo tal número de diagramas se puede cerciorarse que no existe modo de prefijación de la relación \leq en el mografo considerado $G^M(\Psi_a)$, cuando tiene lugar una correspondencia biunívoca entre los términos primarios x_i^q y los vértices de diagramas H tal que toda cadena $v(x_a) \leq v(x_b) \leq \dots \leq v(x_c)$ corresponde biunívocamente a la implicante sencilla $x_a \cdot x_b \cdot \dots \cdot x_c$.

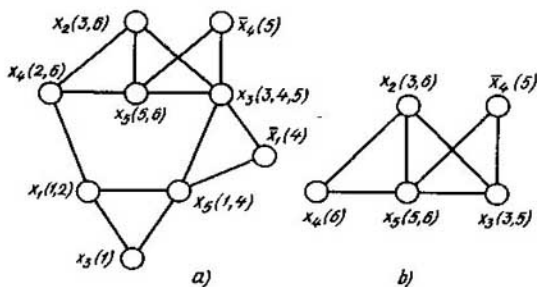


Fig. 5.10

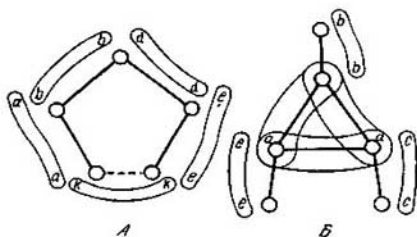


Fig. 5.11

La densidad de trabajo en la revelación de tales contradicciones y, por lo tanto, en la construcción del esquema absolutamente mínimo es considerablemente menor, si se conoce la semántica de transformación determinada por la distribución de las figuras prohibidas.

Teorema 5.2. *Figuras prohibidas Q_A , Q_B de la transformación de un mografo G^M en el diagrama H son los submografos tipos A y B (fig. 5.11).*

El submografo tipo A es un ciclo de longitud impar con la conmutación cíclica de los pesos. El submografo tipo B es un triángulo con vértices pendientes, cada uno de los cuales tiene peso común con el vértice pendiente adyacente y todos los vértices del triángulo tienen peso común con tal que los vértices pendientes pueden coincidir tanto dos a dos como todos juntos uniendo de modo correspondiente sus identificadores. La presencia de una de estas figuras en el mografo hace imposible en principio la prefijación de la relación \leq realizando la transformación $G^M \rightarrow H$ sin desintegrar termas primarias en el mografo G^M , lo que se realiza «a ciegas» sin saber la semántica de la transformación $G^M \rightarrow H$. Además, excepto la eliminación de figuras prohibidas se desintegran los termas primarias en exceso, lo que disminuye la optimicidad de la solución obtenida. Como se muestra a continuación, en el grafo estructural H construido según el mografo G^M sin escoger las figuras prohibidas, se realizan las desintegraciones en exceso.

La selección de las figuras prohibidas de los tipos A y B se reduce a la tarea de hallar los ciclos impar en el mografo junto a la verificación sucesiva de la distribución de pesos sobre ellos. Escogiendo ciclos de longitud impar en el mografo, los vértices de peso igual no se consideran como los vértices que pueden corresponder sólo a los vértices pendientes de las figuras tipo B.

En el ejemplo dado anteriormente, para escoger ciclos de longitud impar, es suficiente considerar el subgrafo $(G^M)''$ representado en la fig. 5.12, a. Examinando este mografo, establecemos que las figuras prohibidas de tipo A son los siguientes submografos:

$$Q_1 = \{x_2(3, 6), x_5(6, 5), x_3(5, 3)\},$$

$$Q_2 = \{x_2(3, 6), x_4(6, 2), x_1(2, 1), \bar{x}_3(1, 4), x_3(4, 3)\},$$

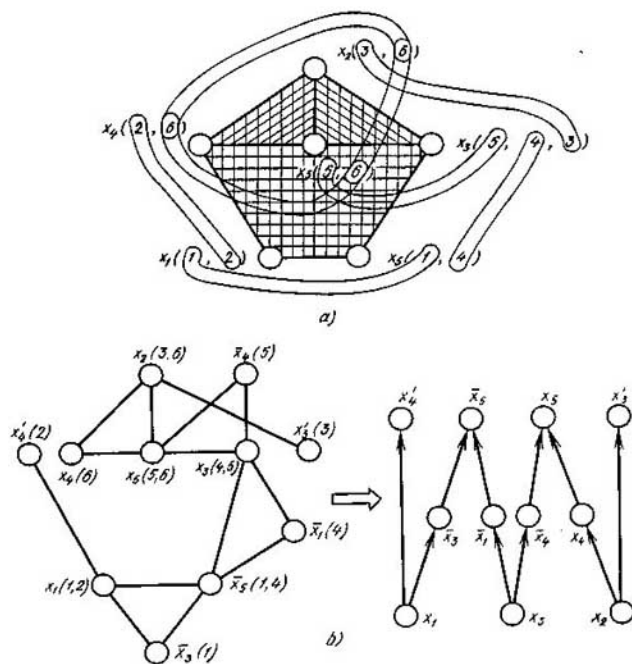


Fig. 5.12

$$Q_3 = \{x_5(6, 5), x_3(5, 4), x_5(4, 1), x_1(1, 2), x_4(2, 6)\};$$

la figura prohibida tipo B es el subgrafo $\{[x_2, x_4, x_5], [x_1, x_4], [x_2, x_3], [x_5, x_4]\}$.

En adelante, la figura de tipo B se prefijará como un triángulo correspondiente. En este caso, la cuarta figura prohibida tiene la siguiente forma: $Q_4 \supset \{x_4(2, 6), x_5(5, 6), x_2(3, 6)\}$.

La propiedad principal de las figuras prohibidas tipos A y B consiste en que, al desintegrar cualquier vértice de una figura tipo A y cualquier vértice del triángulo de una figura tipo B, estos subgrafos dejan de ser figuras prohibidas. De este modo, las desintegraciones semejantes son procedimientos para transformar las figuras prohibidas tipos A y B en las permitidas. El procedimiento de la equivalentización semántica del mgrafo en el parcialmente ordenado es estándar. En la tabla semántica, a las filas

les corresponden las figuras prohibidas tipo A o B, a las columnas, los vértices en desintegración del mografo. La tabla semántica para el caso a examinar se ofrece en la tabla 5.3.

Tabla 5.3

Q_i	$x_1(1, 2)$	$x_2(3, 6)$	$x_3(4, 5)$	$x_3(3, 4)$	$x_3(3, 5)$	$x_4(2, 6)$	$x_5(5, 6)$	$\bar{x}_5(1, 4)$
Q_1	0	1	0	0	1	0	1	0
Q_2	1	1	0	1	0	1	0	1
Q_3	1	0	1	0	0	1	1	1
Q_4	0	1	0	0	0	1	1	0

Para disminuir la densidad de trabajo de la determinación del cubrimiento de la tabla semántica, eliminaremos filas y columnas que se absorben. En este caso, las reglas de absorción son las siguientes.

La columna α se absorbe por la columna β , si no existe una tercera columna ponderada por la misma letra que la columna α y, también, el producto vectorial de las columnas α y β es igual a α .

La fila α se absorbe por la fila β , si el producto vectorial de estas filas es igual a la fila β .

En el caso considerado, la primera y la octava columna se absorben por la sexta. Borrando las columnas que se absorben, tenemos seis cubrimientos de la tabla semántica: $\{x_2(3, 6), x_4(2, 6)\}$, $\{x_2(3, 6), x_5(5, 6)\}$, $\{x_2(3, 6), x_3(4, 5)\}$, $\{x_4(2, 6), x_5(5, 6)\}$, $\{x_3(3, 4), x_5(5, 6)\}$, $\{x_3(3, 5), x_4(2, 6)\}$. Cada uno de estos cubrimientos engendra dos desintegraciones. Por consiguiente, la potencia de la extensión del portador del modelo Ψ_a es igual a 2 y la realización absolutamente minimal de la FND tope que consideramos contiene 11 llaves $L = |M_a| + |\Delta M_a| = 11$.

Para más precisión, consideremos el último cubrimiento y distingamos la letra x_3 en la tercera palabra de la x_3 en la quinta añadiendo la raya en el índice superior: x_3' . A continuación, esta redenominación se llamará *rayado* de la letra en la palabra correspondiente. Análogamente, rayamos la letra x_4 en la segunda palabra. Como resultado obtenemos el modelo $\tilde{\Psi}_a$:

$$\tilde{\Psi}_a = \langle M_a, S_2, S_3 \rangle,$$

$$M_a = \{x_1, \bar{x}_1, x_2, x_3, x_3', \bar{x}_3, x_4, x_4', \bar{x}_4, x_5, \bar{x}_5\},$$

$$S_2 = \{\{x_1, x_4'\}, \{x_2, x_3'\}\},$$

$$S_3 = \{\{x_1, \bar{x}_3, \bar{x}_5\}, \{\bar{x}_1, x_3, \bar{x}_5\}, \{x_3, \bar{x}_4, x_5\}, \{x_2, x_4, x_5\}\};$$

que es equivalente al inicial y se interpreta en los términos de conjunto parcialmente ordenado (fig. 5.12, b):

$$x_1 \leftrightarrow v(x_1), x_2 \leftrightarrow v(x_2), x_3 \leftrightarrow v(x_3),$$

$$\bar{x}_1 \leftrightarrow v(\bar{x}_1), x_3' \leftrightarrow v(x_3'), \bar{x}_3 \leftrightarrow v(\bar{x}_3),$$

$$x_4 \leftrightarrow v(x_4), \quad x_4' \leftrightarrow v(x_4'), \quad \bar{x}_4 \leftrightarrow v(\bar{x}_4), \\ x_5 \leftrightarrow v(x_5), \quad \bar{x}_5 \leftrightarrow v(\bar{x}_5),$$

además,

$$x_1 \bar{x}_3 \bar{x}_5 \leftrightarrow v(x_1) \leq v(\bar{x}_3) \leq v(\bar{x}_5), \\ x_1 x_4' \leftrightarrow v(x_1) \leq v(x_4'), \quad x_2 x_3' \leftrightarrow v(x_2) \leq v(x_3'), \\ \bar{x}_1 x_3 \bar{x}_5 \leftrightarrow v(x_3) \leq v(\bar{x}_1) \leq v(\bar{x}_5), \\ x_3 \bar{x}_4 x_5 \leftrightarrow v(x_3) \leq v(\bar{x}_4) \leq v(x_5), \\ x_2 x_4 x_5 \leftrightarrow v(x_2) \leq v(x_4) \leq v(x_5).$$

Si el cubrimiento de la primera tabla tiene al menos dos letras iguales en el sentido lexicográfico, las palabras donde se realiza su rayado se determinan por la coloración del grafo G . El grafo G está construido sobre el conjunto de palabras que contienen esas letras. Dos vértices suyos son adyacentes, si la letra debe desintegrarse por estas palabras.

De este modo, el conocimiento de la semántica de la transformación $G^M \rightarrow H$ permitió sustituir el sondeo de 5184 diagramas H_i prácticamente construidos por el sondeo de seis cubrimientos de la tabla semántica. En el caso general, si se conoce la semántica, la densidad de trabajo disminuye por el número combinatorio de veces en comparación con el número de todas las soluciones equivalentes.

Examinemos los ejemplos que ilustran el teorema 5.2.

Ejemplo 5.1. El mografo $G^M = \langle V, S_2, S_3 \rangle$, $V = \{a, b, c, d, e\}$, $S_2 = \{ \underbrace{\{a, e\}}_1, \underbrace{\{b, d\}}_2, \underbrace{\{d, c\}}_3 \}$, $S_3 = \{ \underbrace{\{a, b, c\}}_4 \}$ contiene las figuras prohibidas $Q_1 = \{b(2, 4), c(4, 3), d(3, 2)\}$; $Q_2 = \{a(4, 1), b(4, 2), c(4, 3)\}$ (fig. 5.13, a) que engendran la tabla semántica (tabla 5.4).

Tabla 5.4

Figuras prohibidas	$a(1, 4)$	$b(2, 4)$	$c(3, 4)$	$d(2, 3)$
Q_1	0	1	1	1
Q_2	1	1	1	0

La tabla 5.4 tiene tres cubrimientos: $\pi_1 = \{b(2, 4)\}$, $\pi_2 = \{c(3, 4)\}$, $\pi_3 = \{a(1, 4), d(2, 3)\}$; con su ayuda el mografo de partida puede reducirse a la forma que se interpreta mediante tres procedimientos (fig. 5.13, b).

Ejemplo 5.2. El mografo $G^M = \langle V, S_2, S_3 \rangle$,

$$V = \{a, b, c, d\}, \quad S_2 = \{ \underbrace{\{a, d\}}_1, \underbrace{\{b, d\}}_2, \underbrace{\{c, d\}}_3 \}, \quad S_3 = \{ \underbrace{\{a, b, c\}}_4 \}.$$

contiene tres figuras prohibidas de tipo A y una tipo B: $Q_1 = \{a(1, 4), b(2, 4), d(1, 2)\}$, $Q_2 = \{b(2, 4), c(3, 4), d(2, 3)\}$, $Q_3 = \{a(1, 4), c(3, 4), d(1, 3)\}$, $Q_4 = \{a(1, 4), b(2, 4), c(3, 4)\}$ que engendran la tabla semántica (tabla 5.5).

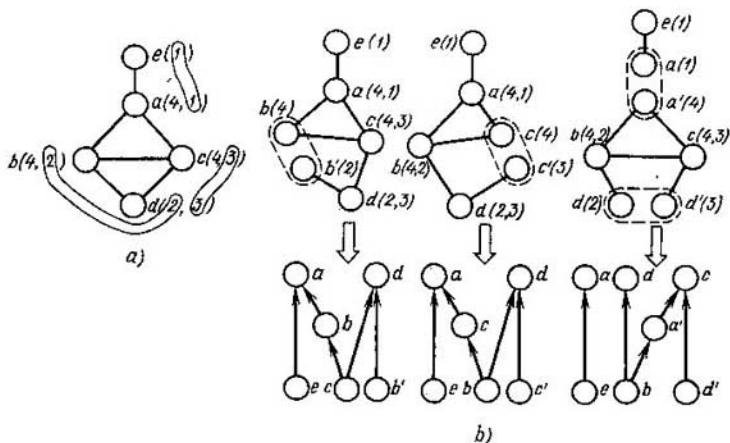


Fig. 5.13

Tabla 5.5

Figuras prohibidas	$a(1,4)$	$b(2,4)$	$c(3,4)$	$d(1,3)$	$d(1,2)$	$d(2,3)$
Q_1	1	1	0	0	1	0
Q_2	0	1	1	0	0	1
Q_3	1	0	1	1	0	0
Q_4	1	1	1	0	0	0

La tabla 5.5 tiene seis cubrimientos: $\pi_1 = \{a(1, 4), b(2, 4)\}$, $\pi_2 = \{a(1, 4), c(3, 4)\}$, $\pi_3 = \{b(2, 4), c(3, 4)\}$, $\pi_4 = \{a(1, 4), d(2, 3)\}$, $\pi_5 = \{b(2, 4), d(1, 3)\}$, $\pi_6 = \{c(3, 4), d(1, 2)\}$; cada uno de ellos engendra el diagrama de Hasse de complejidad 6. Para ser precisos, tomemos el primer cubrimiento, rayemos la letra a en la primera palabra y la letra b , en la segunda. Como resultado obtenemos el mografo.

$$\tilde{G}^M = \langle \tilde{V}, S_2, S_3 \rangle,$$

$$\tilde{V} = \{a, a', b, b', c, d\}, S_2 = \{ \underbrace{\{a', d\}}_1, \underbrace{\{b', d\}}_2, \underbrace{\{c, d\}}_3 \}, S_3 = \{ \underbrace{\{a, b, c\}}_4 \}$$

que equivalentiza al inicial y se interpreta en las categorías del conjunto parcialmente ordenado $\langle V, \leq \rangle$:

$$\{a', d\} \leftrightarrow v(a') \leq v(d), \{b', d\} \leftrightarrow v(b') \leq v(d),$$

$$\{c, d\} \leftrightarrow v(c) \leq v(d), \{a, b, c\} \leftrightarrow v(c) \leq v(a) \leq v(b).$$

Analicemos la estabilidad de figuras prohibidas en dependencia de las condiciones de frontera, es decir, de las condiciones de intersección de la figura prohibida con otra parte del mografo.

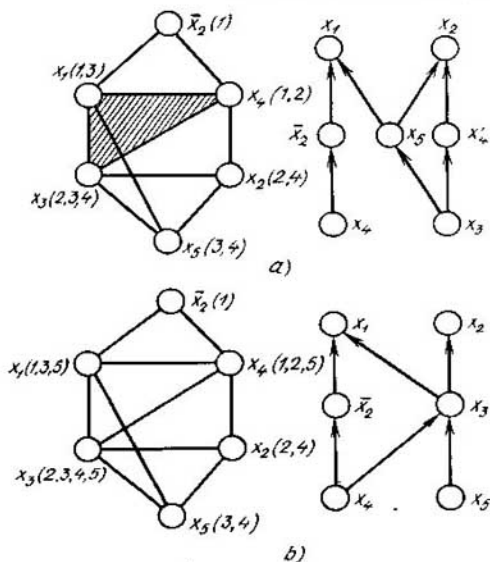


Fig. 5.14

Condiciones de inestabilidad de una figura prohibida tipo A. La composición de una figura de tipo A y una palabra, cuyos portadores coinciden, no debe infringir condiciones de la interpretabilidad del mografo en categorías del conjunto parcialmente ordenado.

Examinemos el mografo

$$G^M = \langle V, S_3 \rangle, \quad V = \{x_1, \bar{x}_2, x_2, x_3, x_4, x_5\},$$

$$S_3 = \left\{ \underbrace{\{x_1, \bar{x}_2, x_4\}}_1, \underbrace{\{x_2, x_3, x_4\}}_2, \underbrace{\{x_1, x_3, x_5\}}_3, \underbrace{\{x_2, x_3, x_5\}}_5 \right\},$$

que contiene la figura prohibida tipo A:

$$Q_A = \{x_1(1, 3), x_3(3, 2), x_4(2, 1)\}.$$

Transformando este mografo en el grafo estructural, es necesario desintegrar una de las letras x_1, x_3, x_4 . Para ser precisos, desintegremos x_4 en la segunda palabra. Como resultado obtenemos el mografo

$$\tilde{G}^M = \langle \tilde{V}, \tilde{S}_3 \rangle, \quad \tilde{V} = \{x_1, \bar{x}_2, x_2, x_3, x_4, x_4', x_5\},$$

$$\tilde{S}_3 = \left\{ \underbrace{\{x_1, \bar{x}_2, x_4\}}_1, \underbrace{\{x_2, x_3, x_4'\}}_2, \underbrace{\{x_1, x_3, x_5\}}_3, \underbrace{\{x_2, x_3, x_5\}}_4 \right\},$$

que se interpreta en categorías del conjunto parcialmente ordenado (fig. 5.14, a).

Introduzcamos el concepto del par de vértices exteriormente inestables. Llámase *par de vértices exteriormente inestables respecto a un submografo* $(G^M)'$ de un mografo G^M a un par de vértices v_k, v_l ponderados por terms primarias $\bar{x}_i, x_i, v_k(x_i), v_l(x_i)$ tales que la unión de vértices adyacentes a $v_k(\bar{x}_i)$ conforme a un identificador α y de vértices adyacentes a $v_l(x_i)$ conforme a un identificador β incluye el portador del submografo $(G^M)'$.

En el ejemplo a examinar se tiene un par de vértices exteriormente inestables $v(\bar{x}_2), v(x_2)$ respecto al portador de la figura prohibida seleccionada Q . El identificador 1 juega el papel de α , 2 ó 4, el de β .

Según la relación de Poretski

$$A\bar{x} \vee Bx = A\bar{x} \vee Bx \vee AB,$$

al mografo se le puede añadir la palabra AB sin infringir la equivalencia de prefijación por el de la función booleana.

En el caso de presencia de un par de vértices exteriormente inestables en el mografo, la palabra AB tiene la forma

$$(\alpha \setminus \bar{x}_i) \cup (\beta \setminus x_i).$$

En este caso es $x_1x_3x_4$ puesto que el mografo es un retículo distributivo.

La palabra añadida $\{x_1, x_3, x_4\}$ despierta la inestabilidad de la figura prohibida $\{x_1(1, 3), x_3 = (2, 3), x_4(1, 2)\}$. Como resultado el mografo

$$G^M = \langle V, S_3 \rangle, \quad V = \{x_1, \bar{x}_2, x_2, x_3, x_4, x_5\},$$

$$S_3 = \{ \underbrace{\{x_1, \bar{x}_2, x_4\}}_1, \underbrace{\{x_2, x_3, x_4\}}_2, \underbrace{\{x_1, x_3, x_5\}}_3, \underbrace{\{x_2, x_3, x_5\}}_4, \underbrace{\{x_1, x_3, x_4\}}_5 \}$$

es interpretable en categorías del conjunto parcialmente ordenado (fig. 5.14, b).

Añadiendo palabras en el caso de un par de vértices exteriormente inestables respecto a la figura prohibida debe considerarse también la añadida de conexiones que pueden conducir a que aparezcan las figuras prohibidas complementarias. En el mografo, las conexiones (aristas) se añaden, si la palabra que se añade incluye estrictamente al portador de la figura prohibida.

Condiciones de inestabilidad de la figura prohibida tipo B. 1. Una figura prohibida tipo B es inestable si, ponderando el vértice pendiente, el identificador de la palabra, conforme a la cual este vértice es adyacente al vértice del triángulo de la figura, pondera otro más vértice del triángulo.

Examinemos el mografo

$$G^M = \langle V, S_2, S_3 \rangle, \quad V = \{x_1, \bar{x}_6, x_2, x_3, x_4, x_5\},$$

$$S_2 = \{ \underbrace{\{x_1, \bar{x}_6\}}_1, \underbrace{\{x_3, x_4\}}_2 \}, \quad S_3 = \{ \underbrace{\{x_1, x_3, x_5\}}_3, \underbrace{\{x_2, x_3, x_5\}}_4 \},$$

que satisface esta condición. No contiene figuras prohibidas: la correspondencia biunívoca entre los terms primarios (letras) del mografo y

los vértices del grafo estructural, con la cual cada palabra corresponde biunívocamente al camino tiene la forma siguiente:

$$\{x_1, \bar{x}_6\} \leftrightarrow v(x_1) \leq v(\bar{x}_6), \{x_3, x_4\} \leftrightarrow v(x_4) \leq v(x_3),$$

$$\{x_1, x_3, x_5\} \leftrightarrow v(x_1) \leq v(x_5) \leq v(x_3),$$

$$\{x_2, x_3, x_5\} \leftrightarrow v(x_2) \leq v(x_5) \leq v(x_3).$$

2. Una figura prohibida tipo B es inestable, si la condición 1 se cumple para dos vértices pendientes, pero es estable si esta condición se cumple para los tres vértices.

Analicemos el mografo

$$G_\alpha^M = \langle V, S_3 \rangle, V = \{a, b, c, d, e, f\},$$

$$S_3 = \{ \underbrace{\{a, c, d\}}_1, \underbrace{\{a, b, e\}}_2, \underbrace{\{b, c, f\}}_3, \underbrace{\{a, b, c\}}_4 \},$$

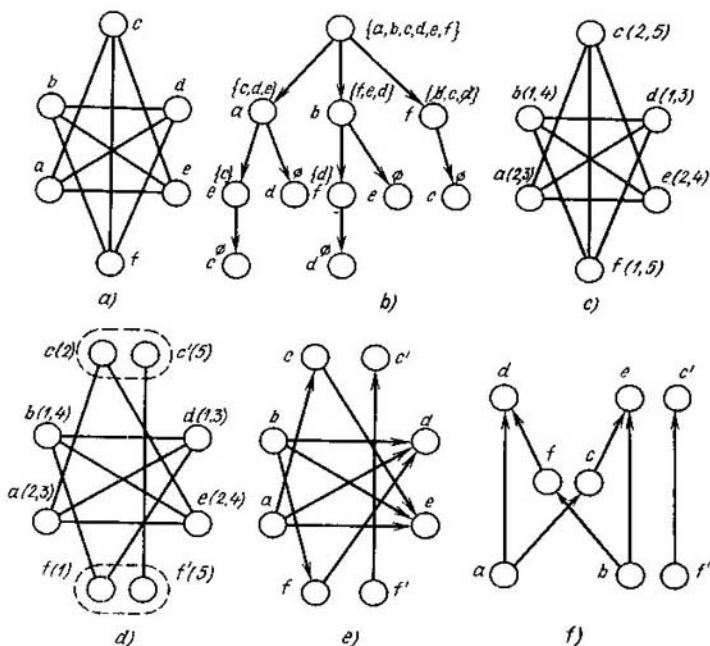


Fig. 5.15

que contiene una figura prohibida inestable de tipo A

$$Q_A = \{a(1, 2), b(2, 3), c(1, 3)\}$$

y una figura prohibida estable tipo B, cuya base es el triángulo con los vértices a, b, c , y los vértices pendientes d, e, f . Esta figura será inestable si se añade una de las palabras $\{b, e, f\}$, $\{c, d, f\}$, $\{a, d, e\}$.

El caso particular de la ordenación parcial del mografo es la orientación transitiva del grafo. Aquí el mografo se considera sin simulación, o sea, las palabras del modelo obtenido son los subgrafos máximamente completos del grafo, sobre el cual se realiza la orientación transitiva.

Para este caso tenemos tres figuras prohibidas. Dos primeras figuras son tipo Q_A y Q_B consideradas sin simulación, la tercera es el grafo G_α^M sin simulación, el que es la composición de una figura prohibida inestable tipo A y una estable tipo B.

Examinemos la aplicación del teorema 5.2 durante la orientación transitiva de grafos en el ejemplo de un representante de una de las familias transitivamente no orientadas de grafos: un grafo que hace completo un ciclo de longitud más que cinco. Consideremos un ciclo de longitud igual a seis. Lo hace completo el grafo G dado en la fig. 5.15, *a*.

Aplicando el algoritmo de formación de subgrafos completos al grafo considerado (fig. 5.15, *b*), obtenemos el conjunto de subgrafos completos $\{\{b, d, f\}, \{a, c, e\}, \{a, d\}, \{b, e\}, \{c, f\}\}$ que forma la signatura del mografo (fig. 5.15, *c*):

$$G^M = \langle V, S_2, S_3 \rangle, V = \{a, b, c, d, e, f\},$$

$$S_3 = \underbrace{\{\{b, d, f\}\}}_1, \underbrace{\{\{a, c, e\}\}}_2, \underbrace{\{\{a, d\}\}}_3, \underbrace{\{\{b, e\}\}}_4, \underbrace{\{\{c, f\}\}}_5$$

El mografo $G^M = \langle V, S_2, S_3 \rangle$ contiene dos figuras prohibidas estables de tipo B: $Q_{B1} = \{\{b, d, f\}, \{b, e\}, \{a, d\}, \{c, f\}\}$, $Q_{B2} = \{\{a, c, e\}, \{a, d\}, \{c, f\}, \{b, e\}\}$. Desintegrando uno de los vértices del triángulo de cada figura prohibida, obtenemos el grafo transitivamente orientado. Son posibles nueve procedimientos de la desintegración. En cada uno de ellos hay que desintegrar dos vértices ya que estos triángulos no se intersecan. Para ser precisos, escogemos los vértices c y f (fig. 5.15, *d*). Como resultado obtenemos el grafo transitivamente orientado (fig. 5.15, *e*) que se transforma en el grafo estructural (fig. 5.15, *f*) después de eliminar arcos de cierre transitivo.

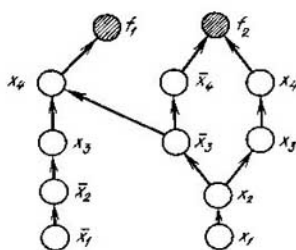
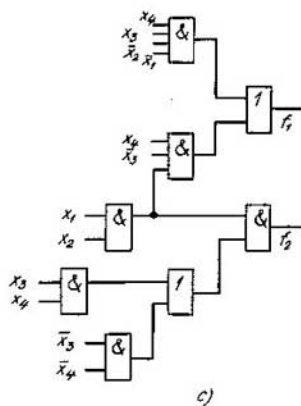
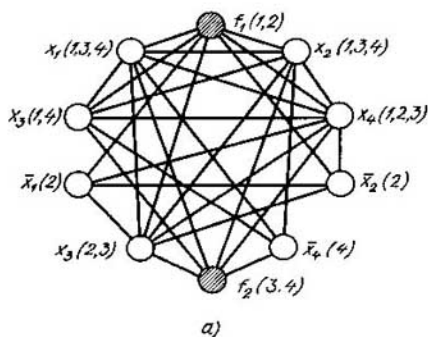
§ 5.3. Caracterización de la conexión de salida de los circuitos lógicos. Minimización estructural

El diseño de los circuitos lógicos de salidas múltiples en las bases topológicas no se distingue del diseño de los circuitos de una salida, si se puede leer la información de elementos que no son minimales ni maximales y, además, se pone una sola restricción: no desintegrar los elementos del grafo, ponderados por las letras de salida f_i . Como regla, diseñando los circuitos de salidas múltiples en las bases funcionales el sistema realizado de funciones booleanas se reduce a una función o bien se buscan las intersecciones de dominios unitarios de funciones booleanas y se sintetizan los circuitos

por las funciones booleanas que describen estas intersecciones. El circuito final es una composición de los circuitos que realizan el comportamiento de las funciones booleanas dadas en las intersecciones de las zonas de trabajo y los circuitos-montaje, cuyas salidas coinciden con los canales de salida del circuito buscado. Si los dominios unitarios no se intersectan, en el empleo de los métodos conocidos tiene lugar la realización inconexa del sistema de funciones booleanas. Precisamente este caso se encuentra con frecuencia en la práctica. En el enfoque propuesto para el diseño de los circuitos lógicos tenemos realización conexa del sistema de funciones booleanas. Examinemos el siguiente ejemplo. Sea dado un sistema de funciones booleanas de tipo

$$f_1 = \underbrace{x_1 x_2 x_3 x_4}_1 \vee \underbrace{\bar{x}_1 x_2 x_3 x_4}_2; \quad f_2 = \underbrace{x_1 x_2 x_3 x_4}_3 \vee \underbrace{x_1 x_2 \bar{x}_3 \bar{x}_4}_4.$$

Transformemos el mografo $G^M(\{f_i\})$ que prefija este sistema (fig. 5.16, a) en un grafo estructural $H(\{f_i\})$ de tal modo que los elementos maximales



b)

Fig. 5.16

sean ponderados por las letras que identifican los canales de salida y ninguno de los vértices que no son elementos maximales sea ponderado por la letra de salida f_i . El grafo estructural está representado en la fig. 5.16(b). Con ayuda de la coálgebra de grafos lo transformemos en el circuito lógico (fig. 5.16, c). Coloremos los vértices del mografo y del grafo estructural que determinan el circuito lógico de salidas múltiples con dos colores: los vértices ponderados por las letras de entrada x_i , \bar{x}_i sean blancos, los ponderados por las letras de salida f_i sean negros (en las figuras el rayado corresponde al color negro).

Al transformar el mografo $G^M(\{f_i\})$, que prefija el sistema de las funciones booleanas, en el grafo estructural $H(\{f_i\})$, sobre este último se pone la siguiente restricción: los elementos maximales del grafo estructural y sólo ellos, deben ser ponderados por las letras de salida f_i^- que no se desintegran con eso.

Representemos un sistema de funciones booleanas $f(x) = \{f_i\}$ en forma del modelo

$$\Psi_a = \langle M_a, p, S_1, S_2, \dots, S_n \rangle,$$

donde $M_a = \{m_1, m_2, \dots, m_n; m_{n+1}, \dots, m_{n+k}\}$, $S_i \subset M_a^i$, $i = 1, 2, \dots, n$, p es predicado monádico que parte M_a en dos subconjuntos:

$$p = \begin{cases} 0 & \text{sobre los elementos } m_i, \text{ cuando } i = 1, 2, \dots, n, \\ 1 & \text{sobre los elementos } m_j, \text{ cuando } j = (n+1), \dots, (n+k). \end{cases}$$

El grafo estructural $H(F(x))$ puede representarse en forma del modelo

$$\Psi_b = \langle M_b, \leq, q \rangle$$

donde

$$q = \begin{cases} 1, & \text{si de } p(m_i) \geq p(m_j) \text{ se deduce } m_i \geq m_j, \\ 0 & \text{en caso contrario.} \end{cases}$$

Investigar la transformación del modelo Ψ_a en el Ψ_b y la construcción del grafo estructural de salidas múltiples deben considerarse modelos con las siguientes restricciones

1) si en el modelo Ψ_a existen dos palabras μ_α y μ_β tales que $\mu_\alpha \subset \mu_\beta$, estas palabras se sustituyen por una palabra μ_α ;

2) si la palabra tiene por lo menos dos letras iguales m_a y m_b ($m_a = m_b$), una de ellas sustituye la otra;

3) si en el modelo Ψ_a existen al menos dos palabras μ_α y μ_β tales que

$$\mu_\alpha = \alpha m_r, \quad \mu_\beta = \alpha m_r,$$

donde $p(m_r) = p(m_i) = 1$, α se compone de letras m_i , $p(m_i) = 0$, una de estas palabras sustituye la otra.

Obviamente, para que sea posible ordenar parcialmente las letras del modelo Ψ_a , es imprescindible que el mografo G^M no contenga figuras

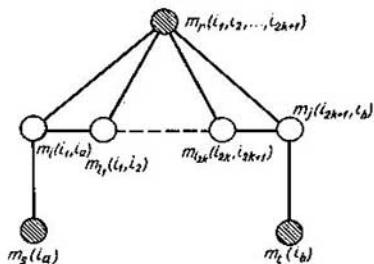


Fig. 5.17

prohibidas Q_A , Q_B (véase la fig. 5.11). En este caso es posible la ordenación parcial de las letras de los modelos Q_A , Q_B sin tener en cuenta el predicado q en el modelo Ψ_b .

Hallemos las figuras prohibidas en el mografo $G^M F(x)$ que caracterizan la fijación de los elementos maximales (minimales) en el grafo estructural correspondiente $H(F(x))$ durante la transformación $G^M F(x) \rightarrow H(F(X))$. Para ser más precisos, fijemos los elementos maximales que corresponden a los canales de salida del circuito lógico en proyección.

El siguiente teorema establece la condición de ordenación parcial de las letras de un modelo Ψ_a , en la cual se toman en consideración los elementos maximales dados.

Teorema 5.3. *Entre las letras de un modelo*

$$\Psi_a = \langle M_a, p, S_1, S_2, \dots, S_n \rangle,$$

cuyo mografo G^M no contiene los mografos Q_A y Q_B existe la relación de ordenación parcial si, y sólo si, el mografo G^M no contiene los subgrafos modelos Q_E (fig. 5.17).

En lo principal, la complejidad de los circuitos lógicos se determina por la complejidad del grafo estructural correspondiente H . Por consiguiente, la minimización estructural de una función booleana f se determina por la distribución de figuras prohibidas Q_A y Q_B en el mografo $G^M(f)$, la minimización estructural de un sistema de funciones booleanas $\{f_i\}$, por la distribución de figuras prohibidas Q_A , Q_B , Q_E en el mografo $G^M(\{f_i\})$. De este modo, al circuito lógico minimal le corresponderán aquellas FND de funciones booleanas, en las cuales es necesario realizar el número minimal de desintegraciones de las termas primarias para que el mografo formado sea interpretable en categorías de grafos estructurales.

Analicemos la minimización estructural exacta de la función booleana f que se reduce al cubrimiento de la tabla semántica de profundidad igual a dos (fig. 5.18) y, si es imprescindible, a la coloración de los grafos que corresponden a los cubrimientos de la segunda tabla.

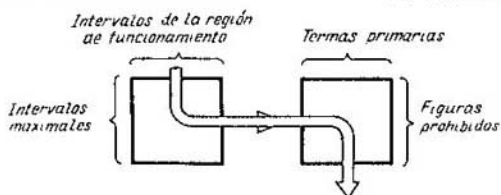


Fig. 5.18

La primera tabla se forma basándose en el cubrimiento de la tabla de distinciones para las funciones booleanas prácticas. El cubrimiento de la primera tabla genera una FND tope, a la cual corresponde el mografo G^M . Su transformación en el parcialmente ordenado, es decir, interpretable por el grafo estructural H , se cumple con ayuda del cubrimiento de la segunda tabla. Consideremos un ejemplo.

Ejemplo 5.3. Determinemos la complejidad del grafo (diagrama) estructural absolutamente minimal que realiza la función booleana $f(x_1, x_2, x_3, x_4)|_1 = \vee(0, 1, 2, 4, 9, 11, 13)$ y es igual a cero sobre los otros juegos.

Formemos intervalos maximales y construyamos la tabla de Quine para función considerada (tabla 5.6).

Tabla 5.6

Números de filas	Intervalos maximales	Puntos unitarios						
		0000	0001	0010	0100	1001	1011	1101
1	000—	✓	✓					
2	00—0	✓		✓				
3	0—00	✓		—	✓			
4	—001		✓		—	✓		
5	10—1					✓	✓	
6	1—01					✓	—	✓

En la tabla de Quine, el signo subrayado corresponde a un intervalo maximal obligatorio. (Un intervalo maximal es obligatorio si existe un punto unitario perteneciente a este, y sólo a este, intervalo). En conjunto de intervalos maximales obligatorios forma el núcleo del cubrimiento.

Hallamos las FND tope de la función dada cubriendo las columnas de la tabla de Quine por las filas de la tabla. Tenemos dos cubrimientos: las primera, segunda, tercera, quinta y sexta filas y las segunda, tercera, cuarta, quinta y sexta filas. A estos dos cubrimientos les corresponden FNDT de la función f de tipo

$$f'(x_1, x_2, x_3, x_4) = \underbrace{x_1 \bar{x}_3 x_4}_1 \vee \underbrace{x_1 \bar{x}_2 x_4}_2 \vee$$

$$\underbrace{\bar{x}_1 \bar{x}_3 \bar{x}_4}_3 \vee \underbrace{\bar{x}_1 \bar{x}_2 x_4}_4 \vee \underbrace{\bar{x}_1 \bar{x}_2 \bar{x}_3}_5;$$

$$f''(x_1, x_2, x_3, x_4) = x_1 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_4 \vee \bar{x}_1 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_4 \vee \bar{x}_2 \bar{x}_3 \bar{x}_4.$$

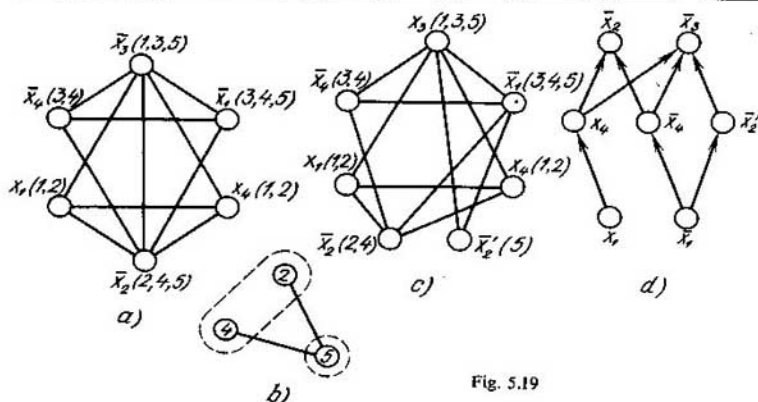


Fig. 5.19

A la FNDT le corresponde el mografo representado en la fig. 5.19, *a*. En este grafo se tienen ciclos de longitud impar con la permutación cíclica de pesos de la forma siguiente:

$$Q_{A1} = \{\bar{x}_2(4, 5), \bar{x}_3(5, 3), \bar{x}_4(3, 4)\};$$

$$Q_{A2} = \{x_1(1, 2), \bar{x}_2(2, 5), \bar{x}_3(5, 1)\};$$

$$Q_{A3} = \{x_4(1, 2), \bar{x}_2(2, 5), \bar{x}_3(5, 1)\};$$

A la primera FNDT le corresponde la tabla semántica (tabla 5.7).

Tabla 5.7.

Q_i	$x_1(1, 2)$	$\bar{x}_2(4, 5)$	$\bar{x}_2(2, 5)$	$\bar{x}_3(3, 5)$	$\bar{x}_3(1, 5)$	$\bar{x}_4(3, 4)$	$x_4(1, 2)$
Q_1	0	1	0	1	0	1	0
Q_2	1	0	1	0	1	0	0
Q_3	0	0	1	0	1	0	1

Uno de los cubrimientos minimales es $\pi = \{\bar{x}_2(4, 5), \bar{x}_2(2, 5)\}$. Puesto que las transformaciones que lo integran desintegran las letras iguales en el sentido lexicográfico, construimos el grafo sobre las palabras $\{2, 4, 5\}$ (fig. 5.19, *b*). Coloreamos este grafo con dos colores: $\{2, 4\}$ y $\{5\}$.

Por lo tanto, está lograda la desintegración minimal de letras. Después de rayar la letra \bar{x}_2 en la quinta palabra obtenemos FND interpretable en categorías del diagrama de Hasse de la complejidad 7 (fig. 5.19, *c*):

$$f'(x_1, x_2, x_3, x_4, \bar{x}_2') = x_1 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_4 \vee \bar{x}_1 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2' \bar{x}_3.$$

Al examinar de manera análoga los demás cubrimientos y la distribución de figuras prohibidas en el mografo de la segunda FND tope obtenemos que la FND a ordenar hallada es absolutamente minimal. Por lo tanto, la complejidad del diagrama que realiza esta función es también igual a 7 (fig. 5.19, *d*).

Citemos la solución exacta del problema de la minimización estructural del sistema de funciones booleanas basada en el uso de figuras prohibidas cuando el mografo se transforma en el diagrama con los elementos maximales fijados. Consiste en el cumplimiento de las siguientes etapas:

1. Empleando uno de los métodos conocidos se forma el conjunto de implicantes simples de salidas múltiples (ISSM) (intervalos maximales de salidas múltiples).

2. Se construye la tabla implicante de Quine, en la cual a toda fila le corresponden la ISSM, a la columna, las constituyentes de la unidad (o de la implicante) de funciones booleanas de partida $f_i(X) \in F(X)$. Además, la constituyente de la unidad (la implicante) entre en la tabla tantas veces, cuantas funciones tomen el valor unidad sobre ella.

3. Se hallan los cubrimientos de las columnas por las filas de la tabla implicante. De este modo se separan las FNDD de sistemas de funciones booleanas.

4. Para toda FNDD del sistema de funciones booleanas, a la cual corresponde el modelo Ψ_a , se construye una FND reticular (FNDR) del sistema de funciones booleanas de la complejidad minimal, es decir, se realiza la transformación $\Psi_a \rightarrow \Psi_b$.

5. De todas las FNDR del sistema de funciones booleanas se escoge la FNDR de complejidad minimal. Luego se construye el grafo estructural H de salidas múltiples.

Para eliminar todas las figuras prohibidas construyamos la tabla semántica R , a cada fila de la cual corresponde biunívocamente una letra (entre paréntesis se indican los identificadores de dos palabras que contienen esta letra durante la transformación de la figura prohibida), a cada columna le corresponde la figura prohibida Q_A, Q_B, Q_E ;

$$r_{ij} = \begin{cases} 1, & \text{si la letra correspondiente a la } i\text{-ésima fila integra} \\ & \text{la } j\text{-ésima figura;} \\ 0 & \text{en caso contrario.} \end{cases}$$

A las filas les corresponden las letras del modelo $m_i \in M_a$, para las cuales $p(m_i) = 0$. Entonces, el cubrimiento de las columnas por las filas en la matriz R corresponde al conjunto de letras que deben desintegrarse, si se realiza la transformación $\Psi_a \rightarrow \Psi_b$.

Ilustremos el método exacto de la minimización de sistemas de funciones booleanas teniendo en cuenta sus propiedades teóricas y estructurales.

Ejemplo 5.4. Sea prefijado el sistema de las funciones booleanas $F(X) = \{f_1(X), f_2(X), f_3(X), f_4(X)\}$ que depende de cinco variables (tabla 5.8). Formemos todas las ISSM, después construyamos la tabla de Quine y como resultado obtenemos dos FNDD del sistema dado:

$$F_1(X) = x_4x_3f_1 \vee x_2x_4f_2 \vee x_3x_5f_3 \vee \bar{x}_1x_2f_4 \vee x_1\bar{x}_2f_4 \vee x_2x_3f_4,$$

Tabla 5.8

x_5	x_4	x_3	x_2	x_1	f_1	f_2	f_3	f_4
0	0	0	0	1	0	0	1	1
0	0	0	1	0	0	0	0	1
0	0	1	0	1	0	0	0	1
0	0	1	1	0	0	0	0	1
0	0	1	1	1	0	0	0	1
0	1	1	0	0	0	1	0	0
0	1	1	0	1	0	1	0	1
0	1	1	1	0	0	1	0	1
0	1	1	1	1	0	1	0	1
0	1	0	0	1	0	0	0	1
0	1	0	1	0	0	0	0	1
1	0	0	0	1	0	0	0	1
1	0	0	1	0	1	0	0	1
1	0	0	1	1	1	0	0	0
1	0	1	0	0	0	0	1	0
1	0	1	0	1	0	0	1	1
1	0	1	1	0	1	0	1	1
1	0	1	1	1	1	0	1	1
1	1	0	0	0	1	0	0	0
1	1	0	0	1	1	0	0	1
1	1	0	1	0	1	0	0	1
1	1	0	1	1	1	0	0	0
1	1	1	0	0	1	1	1	0
1	1	1	0	1	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1

Examinemos la primera FNDT representándola en forma del modelo Ψ_a^1 :

$$M_a = \{x_1, \bar{x}_1, x_2, \bar{x}_2, x_3, x_4, x_5, f_1, f_2, f_3, f_4\};$$

$$p(x_1) = p(\bar{x}_1) = p(x_2) = p(\bar{x}_2) = p(x_3) = p(x_4) = p(x_5) = 0,$$

$$p(f_1) = p(f_2) = p(f_3) = p(f_4) = 1.$$

Las palabras del modelo Ψ_a^1 corresponden a las conjunciones de la función $F_1(X)$. Construyamos el grafo modelo G^M (fig. 5.20, a). Enumeremos las figuras prohibidas que se comprenden en G^M :

$$Q_{A1} = \{x_5(1, 4), x_4(1, 3), x_3(3, 4)\};$$

$$Q_{A2} = \{x_5(2, 4), x_3(4, 7), x_2(2, 7)\};$$

$$Q_{E3} \supset \{x_5(1, 4), x_4(1, 3)\}; Q_{E4} \supset \{x_5(1, 4), x_3(3, 4)\};$$

$$Q_{E5} \supset \{x_5(2, 4), x_3(3, 4)\}; Q_{E6} \supset \{x_5(1, 4), x_3(4, 7)\};$$

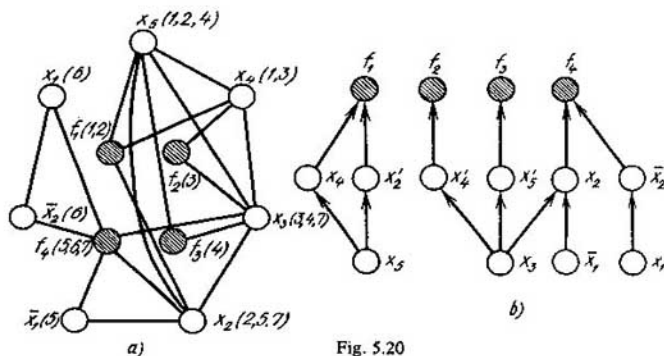
$$Q_{E7} \supset \{x_5(2, 4), x_3(4, 7)\}; Q_{E8} \supset \{x_5(2, 4), x_2(2, 5)\};$$

$$Q_{E9} \supset \{x_5(2, 4), x_2(2, 7)\}; Q_{E10} \supset \{x_4(1, 3), x_3(3, 4)\};$$

$$Q_{E11} \supset \{x_4(1, 3), x_3(3, 7)\}; Q_{E12} \supset \{x_2(2, 7), x_3(3, 7)\};$$

$$Q_{E13} \supset \{x_2(2, 7), x_3(4, 7)\}.$$

Los cubrimientos de la tabla semántica (tabla 5.9) son los conjuntos de palabras: $\{1, 2, 3, 8\}$, $\{1, 2, 3, 5, 6\}$, $\{1, 2, 4, 5, 6\}$, $\{1, 2, 4, 5, 8\}$, $\{3, 4, 6, 7, 8\}$, $\{1, 4, 5, 6, 7, 8\}$.



Para las letras iguales en el sentido lexicográfico construyamos grafos sobre los conjuntos de palabras que contienen esas letras. Su coloración determina la extensión necesaria (ΔM_a) del portador M_a en cada caso concreto.

Tabla 5.9

Letras	Figuras prohibidas												
	1	2	3	4	5	6	7	8	9	10	11	12	13
$x_3(1, 4)$	1	0	1	1	0	1	0	0	0	0	0	0	0
$x_3(2, 4)$	0	1	0	0	1	0	1	1	1	0	0	0	0
$x_4(1, 3)$	1	0	1	0	0	0	0	0	0	1	1	0	0
$x_3(3, 4)$	1	0	0	1	1	0	0	0	0	1	0	0	0
$x_3(3, 7)$	0	0	0	0	0	0	0	0	0	0	1	1	0
$x_3(4, 7)$	0	1	0	0	0	1	1	0	0	0	0	0	1
$x_2(2, 5)$	0	0	0	0	0	0	0	1	0	0	0	0	0
$x_2(2, 7)$	0	1	0	0	0	0	0	0	1	0	0	1	1

La potencia de la extensión del portador $|\Delta M_a|$ para cada uno de los cubrimientos es respectivamente igual a 3, 3, 3, 3, 4. Por consiguiente, la complejidad minimal L del grafo estructural H es igual a 14 (fig. 5.20, b): $L = |M_a| + |\Delta M_a| = 14$. Consideremos el modelo $\Psi_a^{(2)}$ que corresponde a la segunda FNDT

$$M_a = \{x_1, \bar{x}_1, x_2, \bar{x}_2, x_3, x_4, x_5, f_1, f_2, f_3, f_4\};$$

$$p(x_1) = p(\bar{x}_1) = p(x_2) = p(\bar{x}_2) = p(x_3) = p(x_4) = p(x_5) = 0;$$

$$p(f_1) = p(f_2) = p(f_3) = p(f_4) = 1.$$

El mografo G^M (fig. 5.21, a) que define este modelo contiene figuras prohibidas de la siguiente forma:

$$Q_{A1} = \{x_3(1, 4), x_4(1, 3), x_3(3, 4)\};$$

$$Q_{E2} \supset \{x_3(1, 4), x_4(1, 3)\}, \quad Q_{E3} \supset \{x_3(1, 4), x_3(3, 4)\};$$

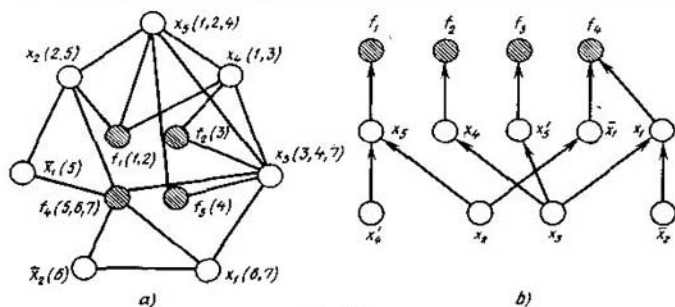


Fig. 5.21

$Q_{E4} \supset \{x_5(2, 4), x_3(3, 4)\}$, $Q_{E5} \supset \{x_3(1, 4), x_3(4, 7, 3)\}$;

$Q_{E8} \supset \{x_4(1, 3), x_3(3, 7)\}$, $Q_{E9} \supset \{x_5(2, 4), x_2(2, 5)\}$;

Cubriendo la tabla semántica (tabla 5.10) obtenemos que la extensión minimal del portador $|\Delta M|$ es igual a dos. La misma engendra el grafo estructural minimal (fig. 5.21, b) que define el sistema dado de las funciones booleanas $F(X)$.

Tabla 5.10

Letras	Figuras prohibidas								
	1	2	3	4	5	6	7	8	9
$x_5(1, 4)$	1	1	1	0	1	0	0	0	0
$x_5(2, 4)$	0	0	0	1	0	1	0	0	1
$x_4(1, 3)$	1	1	0	0	0	0	1	1	0
$x_3(3, 4)$	1	0	1	1	0	0	1	0	0
$x_3(3, 7)$	0	0	0	0	0	0	0	1	0
$x_3(4, 7)$	0	0	0	0	1	1	0	0	0
$x_2(2, 5)$	0	0	0	0	0	0	0	0	1

Examinemos las funcionales de optimización para truncar variantes construidas teniendo en cuenta la estructura de figuras prohibidas. Investiguemos más detalladamente el proceso de reducción del mografo arbitrario a la forma ordenable.

El número de marcas de cada vértice v_i del mografo es igual a la frecuencia propia de la letra correspondiente del modelo y el número de marcas comunes para un par de vértices v_i y v_j (de la arista (i, j)) es igual al valor de la frecuencia mutua de letras correspondientes. Por lo tanto, la arista (i, j) del mografo puede caracterizarse por tres colores: f_i , f_j y f_{ij} , donde f_i es la frecuencia propia de la letra i , f_j es la frecuencia propia de la letra j , f_{ij} es la frecuencia mutua de las letras i y j ($i \neq j$).

Observando el proceso de formación de las figuras prohibidas Q_A , Q_B , Q_E notamos que cuanto mayor sea la suma $f_i + f_j$ de las frecuencias propias de las letras i y j para la frecuencia mutua dada f_{ij} o cuanto menor sea la frecuencia mutua f_{ij} de las letras i y j para la suma dada de sus propias frecuencias, tanto mayor será la probabilidad de que los submodelos tipos Q_A , Q_B , Q_E contengan las letras i y j .

Caractericemos cada arista (i, j) de un mografo $\Psi(f)$ por el valor de la derivada de modelo calculada sobre el par correspondiente de letras:

$$\frac{\partial \Psi}{\partial S}(i, j) = \frac{f_i - 2f_{ij} + f_j}{f_{ij}}$$

Entonces cuanto mayor sea el valor de la derivada, tanto mayor será el grado de participación desigual de las letras en las palabras, cuanto mayor es la heterogeneidad del mografo, tanto mayor será la probabilidad de formación de submodelos tipo Q_A , Q_B , Q_E en este modelo, tanto más complicado (en el sentido del número de elementos) será el grafo sintetizado correspondiente a este modelo. Por eso estimemos el intervalo maximal I de la función f , al cual corresponde el subgrafo completo, por el valor medio $p(I)$ de la derivada calculada para cada arista de este subgrafo, es decir, por la expresión

$$p(I) = \frac{1}{r(r-1)} \sum_{i=1}^{r-1} \sum_{j=i+1}^r \frac{f_i - 2f_{ij} + f_j}{f_{ij}} \quad (5.1)$$

donde r es el rango de la implicante simple I (es igual al número de termas primarias que forman la implicante). Por tanto, la estimación (5.1) permite sintetizar las FND óptimas de la función booleana teniendo en cuenta sus propiedades teóricas y estructurales.

Empleando la funcional de optimización representada en forma de la estimación (5.1), pongamos el algoritmo aproximado de la minimización estructural de la función booleana.

1. Prefijamos la FND dada de la función f en forma de la matriz de incidencia Q .

2. Formamos las implicantes simples de la función f y las inscribimos en la lista I .

3. Construimos el núcleo de la función f . Si es vacío, pasamos al p. 6, en caso contrario, borramos elementos del núcleo de la lista I y pasamos al punto 4.

4. En la matriz Q , los intervalos unitarios de la función f , que se cubren por las implicantes simples borradas de la lista I , los sustituimos por estas mismas implicantes.

5. Si cualquier fila de la matriz Q es implicante simple, pasamos al p. 8, en caso contrario, al punto 6.

6. Según la matriz Q construimos la matriz de frecuencia de relaciones $F = Q^T \times Q$.

7. Conforme a (5.1) estimamos toda implicante simple de la lista I . Escogemos la implicante simple de la estimación minimal, la borramos de la lista I y pasamos al p. 4.

8. La matriz Q obtenida de la función booleana minimizada, teniendo en cuenta sus propiedades teóricas y estructurales.

Ejemplo 5.5. Tomando en consideración las propiedades teóricas y estructurales, minimizamos la función booleana $f(x_1, x_2, x_3, x_4)_1 = \vee (2, 3, 4, 5, 8, 9, 11, 12, 14, 15)$; sobre los otros juegos es igual a cero.

1. La matriz Q tiene forma

$$Q = \begin{array}{c} \begin{array}{cccccccc} x_1 & \bar{x}_1 & x_2 & \bar{x}_2 & x_3 & \bar{x}_3 & x_4 & \bar{x}_4 \end{array} \\ \left(\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 3 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 4 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 5 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 8 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 9 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 11 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 12 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 14 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 15 \end{array} \right) \end{array}$$

2. La lista I de implicantes simples de la función booleana es la siguiente:

$$\begin{array}{l} 001 - 100 - 11 - 0 \\ 010 - -011 1 - 11 \\ -100 10 - 1 111 - \end{array}$$

3. La función f tiene un núcleo compuesto de las implicantes simples obligatorias $010 - , 001 -$ y $100 -$. Después de borrar los elementos del núcleo, la lista I toma la forma

$$\begin{array}{l} -100 11 - 0 \\ -011 1 - 11 \\ 10 - 1 111 - \end{array}$$

4. Como resultado de la sustitución correspondiente de las filas, la matriz Q tiene siguiente forma

$$Q' = \begin{array}{c} \begin{array}{cccccccc} x_1 & \bar{x}_1 & x_2 & \bar{x}_2 & x_3 & \bar{x}_3 & x_4 & \bar{x}_4 \end{array} \\ \left(\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array} \right) \begin{array}{l} 001 - \\ 010 - \\ 100 - \\ 11 \\ 12 \\ 14 \\ 15 \end{array} \end{array}$$

5. Las filas de la matriz Q' , de la cuarta a la séptima, no corresponden a las implicantes simples, por eso pasamos al punto 6.

6. La matriz de frecuencia de la relación F' ($F' = (Q')^T \times Q'$) correspondiente a la matriz Q' es tal:

$$F' = \begin{array}{c} \begin{array}{cccccc} x_1 & \bar{x}_1 & x_2 & \bar{x}_2 & x_3 & \bar{x}_3 & x_4 & \bar{x}_4 \end{array} \\ \left\| \begin{array}{cccccc} 5 & 0 & 3 & 2 & 3 & 2 & 2 & 2 \\ 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 \\ 3 & 1 & 4 & 0 & 2 & 2 & 1 & 2 \\ 2 & 1 & 0 & 3 & 2 & 1 & 1 & 0 \\ 3 & 1 & 2 & 2 & 4 & 0 & 2 & 1 \\ 2 & 1 & 2 & 1 & 0 & 3 & 0 & 1 \\ 2 & 0 & 1 & 1 & 2 & 0 & 2 & 0 \\ 2 & 0 & 2 & 0 & 1 & 1 & 0 & 2 \end{array} \right\| \begin{array}{l} x_1 \\ \bar{x}_1 \\ x_2 \\ \bar{x}_2 \\ x_3 \\ \bar{x}_3 \\ x_4 \\ \bar{x}_4 \end{array} \end{array}$$

7. Estimamos cada implicante simple de la lista obtenida en el punto 3. En virtud de (5.1) tenemos

$$p(-1000) = p(x_2 \bar{x}_3 \bar{x}_4) = \frac{1}{3 \cdot 2} \left(\frac{4 - 2 \cdot 2 + 3}{2} + \frac{4 - 2 \cdot 2 + 2}{2} + \frac{3 - 2 \cdot 1 + 2}{2} \right) \approx 0,91;$$

$$p(-011) \approx 0,91; \quad p(10-1) \approx 1,98; \quad p(1-11) \approx 0,58;$$

$$p(11-0) \approx 0,58; \quad p(111-) \approx 0,67.$$

Escogemos la implicante simple $11-0$ y pasamos al punto 4.

4. En la matriz Q' , las constituyentes de la unidad de la función f , que se cubren por la implicante $11-0$, las sustituimos por esta misma implicante. Como resultado obtenemos la matriz

$$Q'' = \begin{array}{c} \begin{array}{cccccc} x_1 & \bar{x}_1 & x_2 & \bar{x}_2 & x_3 & \bar{x}_3 & x_4 & \bar{x}_4 \end{array} \\ \left\| \begin{array}{cccccc} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array} \right\| \begin{array}{l} 001- \\ 010- \\ 100- \\ 11-0 \\ 11 \\ 15 \end{array} \end{array}$$

5. La matriz Q comprende las constituyentes de la unidad de la función que no son implicantes simples. Pasamos al punto 6.

6. La matriz de frecuencia de la relación F'' correspondiente a Q'' tiene forma

$$F'' = \begin{array}{c} \begin{array}{cccccc} x_1 & \bar{x}_1 & x_2 & \bar{x}_2 & x_3 & \bar{x}_3 & x_4 & \bar{x}_4 \end{array} \\ \left\| \begin{array}{cccccc} 4 & 0 & 2 & 2 & 2 & 1 & 2 & 1 \\ 0 & 2 & 1 & 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 3 & 0 & 1 & 1 & 1 & 1 \\ 2 & 1 & 0 & 3 & 2 & 1 & 1 & 0 \\ 2 & 1 & 1 & 2 & 3 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 0 & 2 & 0 & 0 \\ 2 & 0 & 1 & 1 & 2 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right\| \begin{array}{l} x_1 \\ \bar{x}_1 \\ x_2 \\ \bar{x}_2 \\ x_3 \\ \bar{x}_3 \\ x_4 \\ \bar{x}_4 \end{array} \end{array}$$

7. Según la fórmula (5.1) estimamos las implicantes simples de la lista I que cubren las demás constituyentes de la unidad de la función f . Tenemos

$$p(-0111) \approx 0,75; \quad p(1-11) \approx 0,5; \quad p(10-1) \approx 0,91;$$

$p(111-) \approx 1,16$. Escogemos la implicante simple $1-11$ y pasamos al punto 4.

4. Después de la sustitución correspondiente de las filas obtenemos la matriz

$$Q^m = \begin{array}{cccc|cccc|ccc} x_1 & \bar{x}_1 & x_2 & \bar{x}_2 & x_3 & \bar{x}_3 & x_4 & \bar{x}_4 & & & & \\ \hline 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & - \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & - \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & - \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & - 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & - 1 \end{array}$$

5. Cualquier fila de la matriz Q^m corresponde a una implicante simple. Pasamos al punto 8. Realizamos el punto 5 utilizando la siguiente posición. Si la constituyente de la unidad de la función f es una implicante simple, se contiene en el núcleo del cubrimiento. De aquí, si en el punto 4 no ha cambiado ninguna fila de la matriz, pasamos al punto 8.

8. La matriz Q^m representa la función booleana f minimizada, teniendo en cuenta sus propiedades teóricas y estructurales.

Empleando el método de la equivalentización semántica establezcamos qué lejos está la FNDD obtenida de la función $f(x_1, x_2, x_3, x_4)$ de la FNDD de la función f que corresponde al grafo estructural minimal. Hallemos todas las FND tope de la función f . Para hacerlo construyamos la tabla implicante y determinemos sus cubrimientos. Representemos toda FNDD de la función f en forma de mografo. Formemos figuras prohibidas de los tipos A y B y construyamos las tablas semánticas. Luego hallemos y estimemos sus cubrimientos. Resultará que la FND tope obtenida con ayuda del algoritmo propuesto de minimización, libre del sondeo de todas las FND tope, corresponde a la solución absolutamente minimal.

Analicemos la minimización estructural del sistema de funciones booleanas $F(X)$. En este caso, a toda implicante simple de salidas múltiples, igual que a la implicante simple en la minimización de una función booleana, le corresponde un subgrafo completo, cuyos vértices están ponderados por el identificador de esta ISSM en el mografo G^M . En el caso considerado, el cálculo de la distribución de figuras prohibidas puede estimarse por la expresión (5.1). Al mismo tiempo, se calculan las derivadas sólo para los arcos que unen los vértices ponderados por las letras m_i y m_j para los cuales $p(m_i) = p(m_j) = 0$. Para simplificar, en la expresión (5.1) omitamos el componente constante de frecuencia y estimemos la ISSM por la siguiente fórmula:

$$c(I) = \frac{1}{r(r-1)} \left(\sum_{i=1}^{r-1} \sum_{j=i+1}^r \frac{f_i + f_j}{f_{ij}} \right). \quad (5.2)$$

Propongamos el siguiente procedimiento de minimización de sistemas de funciones booleanas teniendo en cuenta sus propiedades teóricas y estructurales. Este procedimiento se basa en la aplicación de la funcional de optimización.

1. Prefijamos el sistema de las funciones booleanas $F(X) = \{f_1(X), f_2(X), \dots, f_k(X)\}$ por conjuntos M_i^1, M_i^0 . Con ello,

$$f_i(X) = \begin{cases} 1 & \text{sobre los elementos de } M_i^1, \\ 0 & \text{sobre los elementos de } M_i^0. \end{cases}$$

2. Hallamos todas las ISSM de las funciones booleanas por uno de los procedimientos conocidos del sistema y las inscribimos en la lista I .

3. Construimos la matriz Q , a cada columna de la cual corresponde la terma primaria, a la fila, la constituyente de la unidad (implicante) de la función $f_i(X) \in F(X)$ y

$$q_{kl} = \begin{cases} 1, & \text{si } x_k \text{ integra } l\text{-ésima constituyente (implicante);} \\ 0 & \text{en caso contrario.} \end{cases}$$

4. Determinamos las ISSM obligatorias en la lista I . Si ellas existen, pasamos al punto 5 borrando las implicantes obligatorias de la lista I . En caso contrario, pasamos al punto 7.

5. Hacemos correcciones en la matriz Q sustituyendo las constituyentes de la unidad que se cubren por las ISSM borradas, por esas mismas ISSM.

6. Si cualquier fila de la matriz Q es una ISSM, pasamos al p. 9, en caso contrario al punto 7.

7. Según la matriz Q construimos la matriz de frecuencia de relaciones $F = Q^T \times Q$.

8. Estimamos cada ISSM calculando el valor de $c(I)$. Escogemos la ISSM con el valor minimal $c_{\min}(I)$. Borramos la ISSM escogida de la lista I y pasamos al punto 5.

9. La matriz Q prefija el sistema minimizado de las funciones booleanas $F(X)$ tomando en consideración las propiedades teóricas y estructurales.

Ejemplo 5.6. Prefijado (tabla 5.11) el sistema de las funciones booleanas $F_X = \{f_1(X), f_2(X), f_3(X)\}$.

Tabla 5.11

1	x_4	x_3	x_2	x_1	f_1	f_2	f_3
	0	0	0	0	1	0	1
	0	0	0	1	0	0	1
	0	0	1	0	0	0	1
	0	0	1	1	1	0	1
	0	1	0	0	1	0	1
	0	1	0	1	0	0	1
	0	1	1	0	0	0	0
	0	1	1	1	1	0	0
	1	0	0	0	1	1	0
	1	0	0	1	1	0	0
	1	0	1	0	0	1	1
	1	0	1	1	1	0	1
	1	1	0	0	1	1	1
	1	1	0	1	1	0	1
	1	1	1	0	0	1	1
	1	1	1	1	1	0	1

2. La lista I de las implicantes simples de salidas múltiples del sistema $F(x)$ es la siguiente:

$$I_1 = 00 - - (3), I_2 = 0 - 0 (3),$$

$$I_3 = - - 00 (1), I_4 = - 01 - (3), I_5 = - 10 - (3),$$

$$\begin{aligned}
 I_6 &= 1 - 0 - (1), I_7 = 1 - - 0 (2), I_8 = - - 1 1 (1), \\
 I_9 &= 1 - - 1 (1), I_{10} = 1 - - 1 - (3), I_{11} = 1 1 - - (3), \\
 I_{12} &= 1 1 - 1 (1, 3), I_{13} = 1 - 1 1 (1, 3), I_{14} = 1 1 - 0 (2, 3), \\
 I_{15} &= 1 1 0 - (1, 3), I_{16} = 1 - 1 0 (2, 3), I_{17} = - 0 1 1 (1, 3), \\
 I_{18} &= 1 - 0 0 (1,2), I_{19} = - 1 0 0 (1, 3), I_{20} = 0 - 0 0 (1, 3), \\
 I_{21} &= 1 1 0 0 (1, 2, 3).
 \end{aligned}$$

Entre paréntesis se indican los números de las funciones, cuyos puntos de trabajo se cubren por el intervalo correspondiente.

3. Componemos la matriz Q , en la cual cada constituyente de la unidad se repite tantas veces cuantas integre las funciones booleanas.

4. La ISSM obligatoria es la implicante I_8 .

5. En la matriz Q sustituimos las filas $0 0 1 1$, $0 1 1 1$, $1 0 1 1$, $1 1 1 1$ que corresponden a las constituyentes de la función booleana f_1 por la implicante simple de salidas múltiples I_8 . Como resultado obtenemos la matriz:

$$Q^{(1)} = \begin{array}{c} \begin{array}{cccccccc} \bar{x}_4 & x_4 & \bar{x}_3 & x_3 & \bar{x}_2 & x_2 & \bar{x}_1 & x_1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array} \end{array}$$

6. Las filas de la matriz, de la 1-a a la 22-a, no son ISSM. Pasamos al punto 7.

7. La matriz de frecuencia de relaciones tiene forma

$$F = \begin{array}{c} \begin{array}{cccccccc} x_1 & \bar{x}_1 & x_2 & \bar{x}_2 & x_3 & \bar{x}_3 & x_4 & \bar{x}_4 & \\ \hline 9 & 0 & 4 & 5 & 4 & 4 & 5 & 3 & x_1 \\ 0 & 14 & 5 & 9 & 7 & 7 & 9 & 5 & x_1 \\ 4 & 5 & 9 & 0 & 3 & 5 & 6 & 3 & \bar{x}_2 \\ 5 & 9 & 0 & 14 & 8 & 6 & 8 & 6 & x_2 \\ 4 & 7 & 3 & 8 & 11 & 0 & 8 & 3 & \bar{x}_3 \\ 4 & 7 & 5 & 6 & 0 & 11 & 6 & 5 & x_3 \\ 5 & 9 & 6 & 8 & 8 & 6 & 14 & 0 & \bar{x}_4 \\ 3 & 5 & 3 & 6 & 3 & 5 & 0 & 8 & x_4 \end{array} \end{array}$$

8. Estimamos cada ISSM calculando el valor de $c(I)$. Para la implicante $I_3 = \bar{0} \bar{0} 0$ (1) esta estimación es minimal e igual a 1,55. Escogemos $I_3 = \bar{0} \bar{0} 0$ (1) y pasamos al punto 5.

5. En la matriz $Q^{(1)}$, las constituyentes de la unidad, cubiertas por la implicante I_3 , las sustituimos por esta misma implicante. Como resultado obtenemos la matriz

$$Q^2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

6. La matriz $Q^{(2)}$ contiene filas que no son ISSM. Pasamos al punto 7, etc. Resulta la matriz

$$\begin{pmatrix} \bar{x}_4 & x_4 & \bar{x}_3 & x_3 & \bar{x}_2 & x_2 & \bar{x}_1 & x_1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

que corresponde a la solución

$$F(X) = x_1 x_2 f_1 \vee \bar{x}_1 \bar{x}_2 f_3 \vee \bar{x}_2 x_4 f_1 \vee \bar{x}_1 x_4 f_2 \vee \bar{x}_2 \bar{x}_4 f_3 \vee x_2 x_3 f_3 \vee x_3 x_4 f_3.$$

Determinemos la distancia entre la solución obtenida y la minimal. Para esto prefijemos cada FNDT de este sistema por el mografo y determinemos la distribución de figuras prohibidas. Haciendo los mografos obtenidos equivalentes de modo semántico a los mografos interpretados en categorías de grafos estructurales, obtenemos que la FNDT sintetizada del sistema de funciones booleanas $F(X)$ corresponde a la solución absolutamente minimal sin realizar el sondeo de todas las FNDT equivalentes.

§ 5.4. Caracterización de la descomposición del grafo de transiciones en el producto cartesiano parcial

La complejidad y la seguridad del dispositivo automático se determinan en muchos aspectos por los códigos de estados interiores. Uno de los problemas actuales es el de minimización de conexiones entre los elementos de memoria. Consideremos dos elementos de memoria *inconexos* si la función de excitación de uno de ellos no depende del estado del otro elemento de memoria y viceversa. En caso de la inconexión de un elemento de memoria con los demás, su función de excitación se determina por el estado de este elemento y por el vector de entrada:

$$\varphi_i = \varphi_i(Z_i^+, X).$$

Caracterizaremos la memoria del dispositivo automático por su conexión

$$S = \sum_{i=1}^s \alpha_i$$

donde s es el número total de los elementos de memoria, α_i es el número de elementos de memoria, distintos del i -ésimo elemento, cuyos valores son necesarios para calcular la función de excitación del i -ésimo elemento de memoria.

El valor de la conexión de memoria S , igual a cero, significa que los elementos de memoria son funcionalmente inconexos y la función de excitación de cualquier elemento de memoria se determina por su valor y por el vector de entrada.

Examinemos el problema de descomposición de un grafo arbitrario de transiciones en el producto cartesiano parcial de n factores funcionalmente inconexos uno con otro, cada uno de los cuales corresponde a un dispositivo subautomático. La conexión funcional entre los bloques surge cuando es infringido el carácter determinado por lo menos en un grafo de transiciones G_i . Para describir las situaciones de infracción del carácter determinado introduzcamos el grafo de enganche G_{en} , a cada vértice del cual le corresponde biunívocamente un estado interior del dispositivo automático a la arista, un par de estados enganchados con la particularidad de que cada arista está ponderada mediante los vectores de entrada que enganchan los estados correspondientes del dispositivo automático.

Dos estados S_α, S_β se denominan *enganchados*, si existen: un juego X_i que transforma el estado S_α en S_γ , $S_\alpha \xrightarrow{X_i} S_\gamma$, y un juego X_j , $X_i \subset X_j$, que transforma S_β en S_δ , $S_\beta \xrightarrow{X_j} S_\delta$, tales que $S_\gamma \neq S_\delta$ y (S_α, S_γ) y (S_β, S_δ) no forman lazos simultáneamente.

El estado del grafo de transiciones $G = \langle V, (U, X) \rangle$ después de su descomposición en el producto cartesiano parcial $G = \prod_i G_i$

$$G_i = \langle V_i, (U_i, X) \rangle$$

puede caracterizarse por un vector a i -ésimo orden del cual le corresponde el estado del i -ésimo dispositivo subautomático. En la descomposición cada uno de los dispositivos subautomáticos se caracteriza por el número tolerable de estados $|V_i|$. Es obvio que

$$\prod_i |V_i| \geq |V|.$$

A los estados enganchados les deben corresponder los vectores distintos uno de otro en cada orden. En caso contrario, si en j -ésimo orden los vectores de estados enganchados coinciden al transmitir a la entrada del dispositivo automático un vector X , por el cual ellos se enganchan, en el caso general, será infringido el carácter determinado de transición en este dispositivo subautomático. Por lo tanto, la construcción de la descomposición paralela abstracta del dispositivo automático se reduce a la coloración de componentes múltiples (multicoloración) del grafo de enganche, realizando la cual a los vértices adyacentes se les ponen en correspondencia los espectros de colores, distintos uno de otro, en todo componente.

La descomposición del grafo de transiciones en el producto cartesiano parcial no saca el grafo resultante de la clase de los grafos de transiciones. Las figuras prohibidas de esta semántica son los grafos casi completos.

Teorema 5.4. *Si un grafo de enganche construido para cada uno de los dispositivos subautomáticos no contiene grafo casi completo de la casi densidad $q + 1$, el dispositivo automático correspondiente es descomponible en el producto cartesiano parcial de los factores funcionalmente inconexos entre ellos, o sea, de los dispositivos subautomáticos, el número de estados de cada uno de los cuales no supera q .*

De este modo, los grafos casi completos son las figuras prohibidas que caracterizan la condición suficiente de la inconexión funcional de subautomatas, cuando se busca la descomposición paralela del autómatas de mando. A continuación, esta clase de figuras prohibidas se designará mediante Q_{ch} . Al mismo tiempo el grafo de enganche para el primer dispositivo subautomático a examinar es un grafo de enganche construido por el grafo de transiciones según su definición. El grafo de enganche del i -ésimo dispositivo subautomático es un grafo de enganche del primer dispositivo subautomático, en el cual se han añadido aristas que unen los vértices de espectro igual de colores. Estas aristas se añaden para identificar unívocamente los estados del dispositivo automático.

Analicemos la construcción de la descomposición paralela abstracta del dispositivo automático basada en la semántica hallada, examinando el ejemplo siguiente.

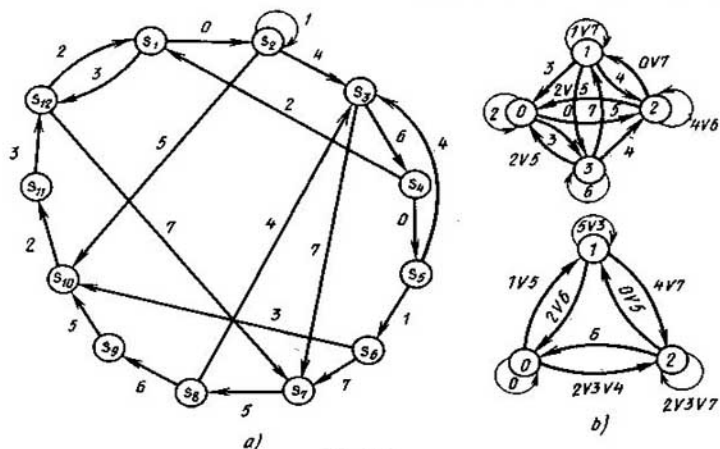


Fig. 5.22

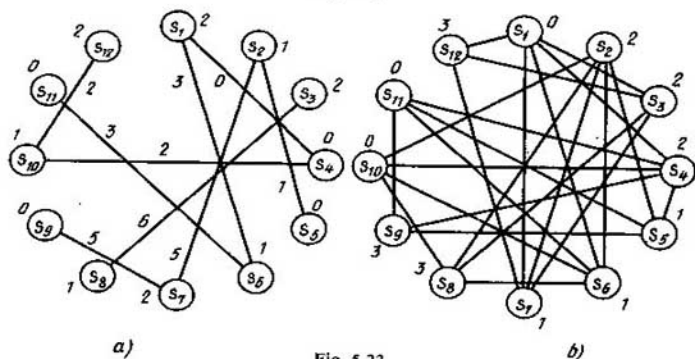


Fig. 5.23

El dispositivo automático tiene tres canales de entrada. El grafo de transiciones está representado en la fig. 5.22, a. Los vectores de entrada se denotan por los equivalentes decimales de los correspondientes juegos binarios. Hallemos la descomposición paralela del dispositivo automático en forma de dos dispositivos con el número de estados interiores igual a 3 y 4, respectivamente.

En la fig. 5.23, a se da el grafo de enganche del primer dispositivo subautomático. No contiene figuras prohibidas; por consiguiente, es posible colorarlo con tres colores $\{0, 1, 2\}$. Cada color corresponde a un estado del primer dispositivo subautomático. Construimos su grafo de transiciones $G_1 = \langle V_1, (U_1, X) \rangle$. Como resultado de la coloración del grafo G_{ent} , el conjunto de estados del dispositivo automático inicial se parte en tres conjuntos concolo-

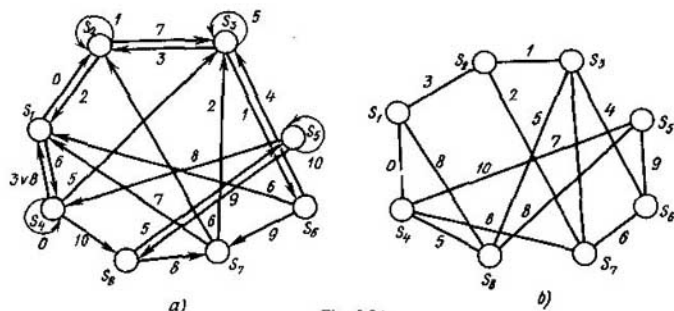


Fig. 5.24

reados, cada uno de los cuales corresponde al estado del primer dispositivo subautomático. Los designaremos por S'_{10}, S'_1, S'_{12} . Tenemos: $\{S_1, S_2, S_3, S_{11}\} = S'_{10}$; $\{S_2, S_6, S_8, S_{10}\} = S'_{11}$; $\{S_1, S_3, S_7, S_{12}\} = S'_{12}$. De aquí se tiene que las condiciones de la transición φ_{i-j} del estado S'_{1i} en el estado S'_{1j} ($i, j = 0, 1, 2$) del primer bloque son las condiciones de la transición del estado $S_{\alpha} \in S'_{1i}$ al estado $S_{\beta} \in S'_{1j}$ que se determinan por el grafo inicial de transiciones: $\varphi_{0-0} = 0$, $\varphi_{0-1} = 1 \vee 5$, $\varphi_{0-2} = 2 \vee 3 \vee 4$; $\varphi_{1-1} = 3 \vee 5$, $\varphi_{1-0} = 2 \vee 5$; $\varphi_{1-2} = 4 \vee 7$; $\varphi_{2-2} = 1 \vee 3 \vee 5$, $\varphi_{2-0} = 6$, $\varphi_{2-1} = 0 \vee 5$.

El grafo de enganche que corresponde al segundo bloque está representado en la fig. 5.23, b. No contiene figuras prohibidas; por consiguiente, el dispositivo automático inicial es realizable en forma de dos dispositivos subautomáticos funcionalmente inconexos que funcionan en paralelo. La coloración del segundo grafo de enganche parte el conjunto de estados del dispositivo automático de partida en cuatro conjuntos siguientes:

$$S_{20} = \{S_1, S_{10}, S_{11}\}, S_{21} = \{S_5, S_6, S_7\}.$$

$$S_{22} = \{S_2, S_3, S_4\}, S_{23} = \{S_8, S_9, S_{12}\}.$$

Para los estados del segundo dispositivo automático las funciones de transición tienen siguiente forma: $\varphi_{0-0} = 2$, $\varphi_{0-2} = 0$, $\varphi_{0-3} = 3$, $\varphi_{1-1} = 1 \vee 7$, $\varphi_{1-0} = 3$, $\varphi_{1-2} = 4$, $\varphi_{1-3} = 5$, $\varphi_{2-2} = 4 \vee 6$, $\varphi_{2-0} = 2 \vee 5$, $\varphi_{2-1} = 0 \vee 7$, $\varphi_{3-3} = 6$, $\varphi_{3-0} = 2 \vee 5$, $\varphi_{3-1} = 7$, $\varphi_{3-2} = 4$.

En la fig. 5.22, b se ofrece la descomposición del grafo dado de transiciones en factores funcionalmente inconexos.

Construyamos la descomposición abstracta del dispositivo automático utilizando la semántica de la descomposición del grafo de transiciones en el producto de factores. Sea prefijado el grafo de transiciones G (fig. 5.24, a) que debe descomponerse en el producto $G = G_1 \times G_2$, donde G_1 y G_2 son los grafos de transiciones, la potencia del portador de cada uno de éstos no supera 3.

Construyamos el grafo de enganche G_{en} y realicemos su coloración por el primer componente (fig. 5.24, b). Para ello formemos todos los subgrafos vacíos $\{S_2, S_4, S_6\}$, $\{S_3, S_4\}$, $\{S_1, S_6\}$, $\{S_1, S_3, S_5\}$, $\{S_1, S_2, S_7\}$, $\{S_2, S_5\}$, $\{S_2, S_6, S_8\}$, $\{S_7, S_8\}$ del grafo de enganche.

Tabla 5.12

Portadores de subgrafos vacíos	Vértices							
	S_1	S_2	S_3	S_6	S_5	S_4	S_8	S_7
$\{S_1, S_5, S_7\}$	1	0	0	0	1	0	0	1
$\{S_1, S_3, S_5\}$	1	0	1	0	1	0	0	0
$\{S_1, S_6\}$	1	0	0	1	0	0	0	0
$\{S_2, S_4, S_6\}$	0	1	0	1	0	1	0	0
$\{S_2, S_6, S_8\}$	0	1	0	1	0	0	1	0
$\{S_2, S_5\}$	0	1	0	0	1	0	0	0
$\{S_3, S_4\}$	0	0	1	0	0	1	0	0
$\{S_7, S_8\}$	0	0	0	0	0	0	1	1

Cubrimos las columnas por las filas de la tabla (tabla 5.12), en la cual a cada fila le corresponde biunívocamente un subgrafo vacío, a la columna, un vértice, y en la intersección de la i -ésima fila y la j -ésima columna se halla 1, si el j -ésimo vértice se contiene en el portador del i -ésimo subgrafo vacío y 0 en caso contrario. El número de estados tolerables del grafo G_1 no supera tres; por lo tanto, la potencia del cubrimiento de esta tabla tampoco debe ser más que tres. El grafo de enganche no contiene figuras prohibidas por el primer componente (grafos casi completos de la casi densidad 4); por consiguiente, tales cubrimientos existen, ya que el número cromático del grafo es igual a su casi densidad. Estos cubrimientos tienen siguiente forma:

$$\pi_1 = \{ \{S_2, S_4, S_6\}, \{S_1, S_3, S_5\}, \{S_7, S_8\} \},$$

$$\pi_2 = \{ \{S_1, S_5, S_7\}, \{S_2, S_6, S_8\}, \{S_3, S_4\} \}.$$

Para ser precisos escojamos el primer cubrimiento. Le corresponde la coloración del grafo de enganche representada en la fig. 5.25, *a*. En virtud de esta coloración, las funciones de transiciones φ_{ijk} , donde i es el número del componente de la descomposición, j es el estado (color), al cual se realiza la transición del primer factor del grafo de transiciones G_1 , tienen siguiente forma: $\varphi_{100} = 0 \vee 1$, $\varphi_{101} = 2 \vee 3 \vee 4 \vee 5 \vee 6$, $\varphi_{102} = 9 \vee 10$, $\varphi_{110} = 1 \vee 3 \vee 7 \vee 8 \vee 0$, $\varphi_{111} = 5 \vee 10$, $\varphi_{112} = 4 \vee 9$, $\varphi_{120} = 6$, $\varphi_{121} = 2 \vee 5 \vee 7$, $\varphi_{122} = 8$.

Para que el producto de grafos satisfaga la condición de autómatas, antes de colorar el grafo de enganche por el segundo componente es necesario unir los vértices, concoloreados en el primer componente, por aristas (fig. 5.25, *b*). Para establecer si el grafo obtenido (fig. 5.25, *c*) contiene figuras prohibidas, formemos subgrafos completos de densidad 4. Volvamos a utilizar el algoritmo dado anteriormente. Lo modificaremos de tal modo que en el nivel se hallen situados los vértices no adyacentes que, al mismo tiem-

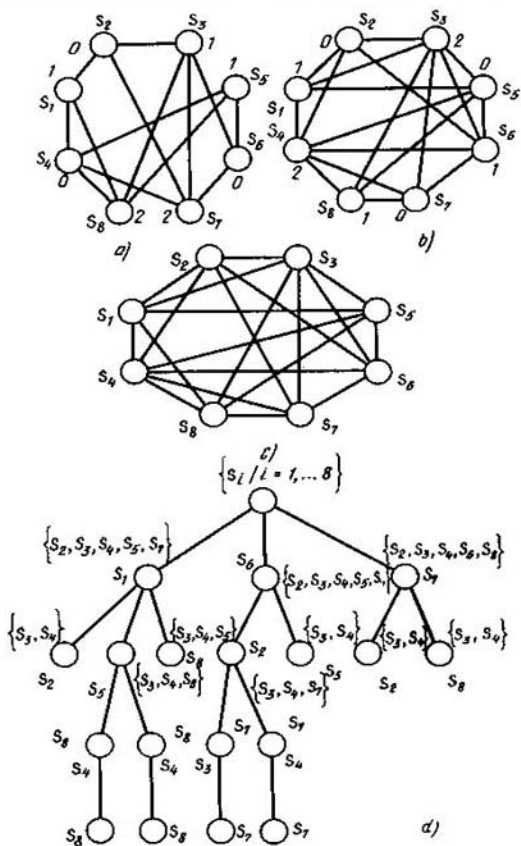


Fig. 5.25

po, se ponderan por los vértices adyacentes. Rompamos los caminos, cuya longitud es menos que cuatro (fig. 5.25, d). Además de los subgrafos completos formados de densidad 4, cuyos portadores tienen respectivamente forma de $\{S_1, S_3, S_5, S_8\}$, $\{S_1, S_4, S_5, S_8\}$, $\{S_2, S_3, S_6, S_7\}$, $\{S_2, S_4, S_6, S_7\}$, el grafo de enganche, al realizar la coloración por el segundo componente, contiene otras seis figuras prohibidas: grafos casi completos de casi densidad 4 (fig. 5.26).

Construyamos la tabla semántica (tabla 5.13). A cada fila de ésta le

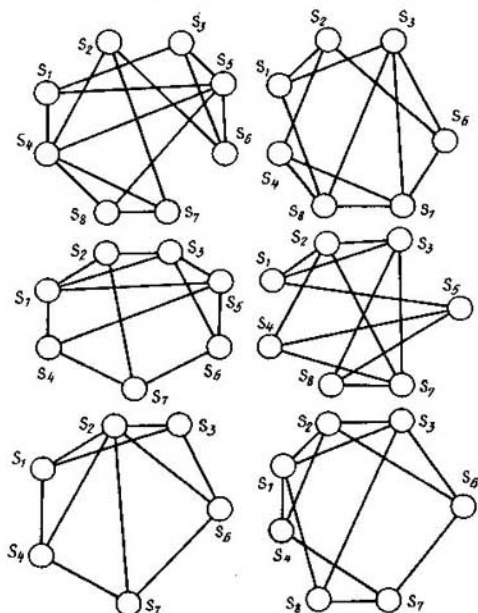


Fig. 5.26

corresponde biunívocamente una arista de figura prohibida, a cada columna, una figura prohibida y

$$(i, j) = \begin{cases} 1, & \text{si la } i\text{-ésima arista se contiene en la } j\text{-ésima figura;} \\ 0 & \text{en caso contrario.} \end{cases}$$

Tabla 5.13

Aristas	Grafos casi completos de la casi densidad 4									
$\{S_1, S_4\}$	1	0	1	0	1	0	1	1	0	0
$\{S_1, S_5\}$	1	0	1	0	1	1	0	0	0	1
$\{S_1, S_6\}$	1	0	0	1	0	0	0	1	0	1
$\{S_4, S_6\}$	1	0	1	1	0	0	0	0	0	0
$\{S_4, S_5\}$	1	0	1	0	1	1	0	0	0	0
$\{S_5, S_6\}$	1	0	1	0	0	1	0	0	0	1
$\{S_2, S_7\}$	0	1	1	0	1	1	1	0	1	0
$\{S_2, S_3\}$	0	1	0	0	1	1	1	1	0	0
$\{S_2, S_6\}$	0	1	1	1	0	0	1	1	1	0
$\{S_3, S_7\}$	0	1	0	1	0	1	0	0	0	0
$\{S_6, S_7\}$	0	1	0	1	1	0	1	1	1	0

Aristas	Grafos casi completos de la casi densidad 4									
$\{S_3, S_6\}$	0	1	1	1	1	0	1	1	0	0
$\{S_2, S_4\}$	0	0	1	1	0	1	1	1	1	0
$\{S_3, S_8\}$	0	0	0	1	0	1	0	1	0	1
$\{S_3, S_5\}$	0	0	1	0	1	0	0	0	0	1
$\{S_5, S_6\}$	0	0	1	0	1	0	0	0	0	0
$\{S_4, S_7\}$	0	0	1	1	1	1	1	1	1	0
$\{S_1, S_3\}$	0	0	1	1	1	1	1	1	0	1
$\{S_7, S_8\}$	0	0	1	1	0	1	0	1	0	0
$\{S_1, S_2\}$	0	0	0	1	1	1	1	1	0	0
$\{S_4, S_6\}$	0	0	0	0	0	0	0	0	1	0

La tercera y la séptima fila forman el cubrimiento minimal. Por consiguiente, eliminando del grafo de enganche sus aristas correspondientes $\{S_1, S_8\}$ y $\{S_2, S_7\}$ es posible la coloración del grafo mediante tres colores (véase la fig. 5.25, *b*). Como resultado obtenemos la coloración de dos componentes del grafo de enganche (véase la fig. 5.25, *a, b*): S_1 : (1, 1), S_2 : (0, 0), S_3 : (1, 2), S_4 : (0, 2), S_5 : (1, 0), S_6 : (0, 1), S_7 : (2, 0), S_8 : (2, 1).

La eliminación de las aristas $\{S_1, S_8\}$ y $\{S_2, S_7\}$ significa que los estados S_1 y S_8 del grafo inicial de transiciones no deben engancharse por el vector de entrada $\mathbf{8}$ (véase la fig. 5.25, *a*) y los estados S_2 y S_7 , por el vector $\mathbf{2}$. Para desenganchar esos vectores, en las correspondientes cuatro transiciones introduzcamos órdenes complementarios, según los cuales se distinguirán estos vectores. Es posible realizarlo usando las conexiones entre los componentes (en el caso dado, el estado del primer componente) o formando un bloque desacoplador especial de memoria.

En el primer componente del espectro, a los estados S_1 y S_8 se les han puesto en correspondencia los colores 1 y 2, respectivamente. Para desenganchar estos estados, extendamos el vector $\mathbf{8}$ que pondera la transición del estado S_1 hasta el vector $\mathbf{8}' = \mathbf{8} \cdot \tilde{S}_{11}$; el vector $\mathbf{8}$ que pondera la transición del estado S_8 lo extendamos hasta el vector $\mathbf{8}'' = \mathbf{8} \cdot \tilde{S}_{12}$, donde \tilde{S}_{11} , \tilde{S}_{12} son valores de los órdenes, en los cuales se difieren los códigos de los colores 1 y 2 del primer componente del espectro (fig. 5.27, *a*). Para desenganchar los estados S_2 y S_7 tenemos $\mathbf{2}' = \mathbf{2} \cdot \tilde{S}_{10}$; $\mathbf{2}'' = \mathbf{2} \cdot \tilde{S}_{12}$, respectivamente.

En este caso, el bloque especial de memoria es un elemento de memoria α (fig. 5.27, *b*), en el cual uno de los estados (por ejemplo, nulo) se pone en correspondencia a los estados S_1 y S_2 , el otro, a los estados S_8 y S_7 . En este caso, $\mathbf{8}' = \mathbf{8}\alpha$, $\mathbf{8}'' = \mathbf{8}\alpha$, $\mathbf{2}' = \mathbf{2}\alpha$, $\mathbf{2}'' = \mathbf{2}\alpha$. El valor del bloque desacoplador especial de memoria se fija realizando la transición al estado, la salida, del cual es de carácter unívoco, lo que se determina por el estado de este bloque.

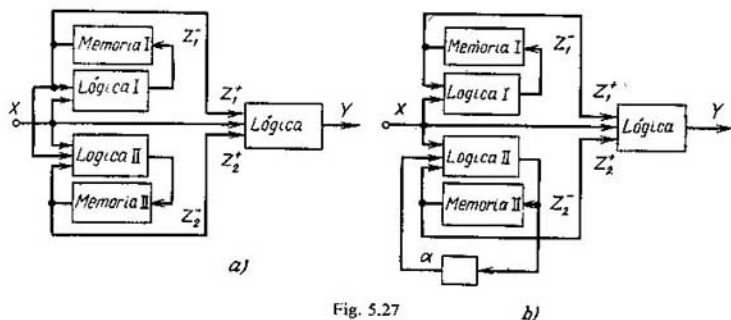


Fig. 5.27

Como resultado obtenemos las siguientes funciones de excitación del grafo de transiciones que determina el segundo factor de la descomposición:

$$\begin{aligned} \varphi_{200} &= 1 \vee 6 \vee 10, & \varphi_{201} &= 2' \vee 7 \vee 9, & \varphi_{202} &= 2'' \vee \\ & \vee 3 \vee 8, & \varphi_{210} &= 0 \vee 5 \vee 8'' \vee 9, & \varphi_{211} &= 6, \\ \varphi_{212} &= 3 \vee 4 \vee 8', & \varphi_{220} &= 7, & \varphi_{221} &= 1 \vee 4 \vee 6, \\ \varphi_{222} &= 0 \vee 5. \end{aligned}$$

La descomposición paralela abstracta obtenida del dispositivo automático prefijado con los factores conexos está representada en la fig. 5.28.

Para responder si es posible la descomposición del dispositivo automático a examinar en el producto cartesiano parcial de los factores inconexos entre ellos, consideremos la coloración del grafo de enganche teniendo en cuenta el carácter de transiciones de estados enganchados, lo que permite no tomar en consideración la conexión en el grafo de enganche, si la transición se realiza de los correspondientes estados enganchados a los vértices concoloreados. Por lo tanto, antes de extender el vector de entrada para desenganchar los estados interiores es necesario comprobar la posibilidad de eliminar esta conexión mediante la coloración igual de aquellos estados, a los cuales pasan los estados enganchados.

Construyamos la tabla de transiciones del dispositivo automático prefijado (tabla 5.14), a cada fila de la cual le corresponde biunívocamente un valor del vector de entrada, a cada columna, un estado interior, y en la célula (i, j) de la tabla se sitúa el identificador del estado interior, al cual pasa el dispositivo automático del j -ésimo estado bajo la influencia del i -ésimo vector de entrada.

La descomposición inconexa tiene lugar si, debido al cubrimiento de la tabla semántica, la conexión de los pares $\{S_1, S_8\}$ y $\{S_2, S_7\}$ no se tiene

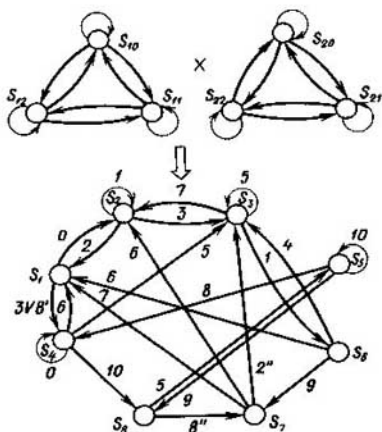


Fig. 5.28

en cuenta en virtud de que los pares de vértices $\{S_4, S_7\}$ y $\{S_1, S_3\}$ pueden ser concoloreados. Los estados S_4 y S_7 están enganchados y, para que sean concoloreados, es imprescindible que sean concoloreados los vértices S_1 y S_2 , a los cuales pasan S_4 y S_7 . Los estados S_1, S_2 están enganchados y, para que sean concoloreados, es imprescindible que sean concoloreados los vértices S_3 y S_4 . Los estados S_3 y S_4 no están enganchados; por lo tanto, pueden ser concoloreados. Coloramos los vértices S_3, S_4 con un mismo color. Entonces, según la propiedad transitiva de la relación de concoloración obtenemos que cada uno de los subconjuntos $K_0 = \{S_1, S_2, S_8\}$, $K_1 = \{S_3, S_4, S_7\}$, $K_2 = \{S_5, S_6\}$ se compone de los vértices concoloreados.

Para la coloración hallada del grafo de enganche del segundo componente de descomposición, los estados S_2, S_7 son no concoloreados. Por consiguiente, está también liquidada la segunda contradicción que condiciona el carácter indeterminado del grafo de transiciones del segundo factor. En definitiva obtenemos las siguientes funciones de excitación del segundo componente de descomposición: $\varphi_{200} = 0 \vee 1 \vee 2$, $\varphi_{201} = 3 \vee 8$, $\varphi_{202} = 5$, $\varphi_{210} = 6 \vee 7 \vee 10$, $\varphi_{211} = 0 \vee 2 \vee 5$, $\varphi_{212} = 1$, $\varphi_{220} = 6 \vee 9$, $\varphi_{221} = 4 \vee 8$, $\varphi_{222} = 10$.

De este modo, tenemos la descomposición inconexa del dispositivo automático prefijado que se determina por los sistemas obtenidos y la coloración de dos componentes de la forma $S_1 - (1, 0)$, $S_4 - (0, 1)$, $S_7 - (2, 1)$, $S_2 - (0, 0)$, $S_5 - (1, 2)$, $S_8 - (2, 0)$, $S_3 - (1, 1)$, $S_6 - (0, 2)$.

En el caso general, para construir la descomposición abstracta paralela de la conexión absolutamente minimal, es necesario estimar cada coloración por el primer componente mediante las coloraciones por el segundo

Tabla 5.14

x_i	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
0								
1	S_2			S_4				
2		S_2	S_6				S_3	
3	S_4	S_1		S_3				
4		S_3				S_3		
5			S_3					S_5
6				S_1		S_1	S_2	
7			S_2			S_7	S_1	
8	S_4				S_4		S_7	
9					S_6			
10				S_8	S_5			

y elegir una coloración de componentes múltiples que satisfaga las cantidades prefijadas de vértices de los grafos de factores y sus condiciones de conexión. La descomposición del grafo de transiciones en n componentes es análoga.

Examinemos la descomposición abstracta paralela límite de los dispositivos automáticos cuando el dispositivo subatómico es un elemento de memoria. La semántica de esta descomposición será semántica de la conexión funcional de los elementos de memoria. Teniendo en cuenta la estructura de los grafos casi completos y el carácter de dos signos de la lógica de Boole, la semántica reflexiva de la conexión funcional de los elementos de memoria del dispositivo automático se determina por la siguiente afirmación.

Teorema 5.5. *Los grafos de enganche que no contienen ciclos de longitud impar determinan la codificación, en la cual los elementos de memoria son funcionalmente inconexos.*

Este criterio permite determinar sucesivamente los valores de los órdenes en los códigos de estados interiores de modo análogo a cómo se hacía durante la búsqueda de la descomposición abstracta paralela.

§ 5.5. Caracterización y métodos del emplazamiento óptimo de los datos en la memoria del ordenador

Los sistemas modernos de información se caracterizan no sólo por grandes capacidades, sino también por la complejidad de los datos almacenados, la que consiste en que los datos están en diferentes interrelaciones. De tal modo, los datos complejos se representan en forma de un juego de ciertos objetos elementales y un conjunto de relaciones que unen los objetos de datos. En otras palabras, los sistemas complejos de información se formalizan

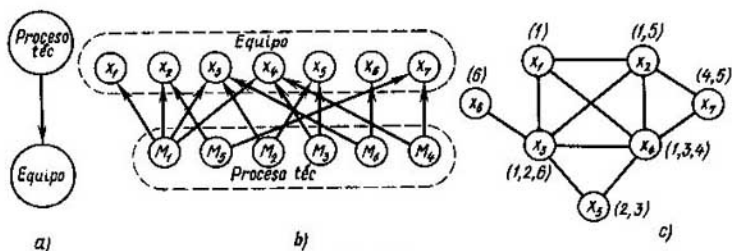


Fig. 5.29

zan mediante los conceptos tales como *grafo* y *mografo*. La rica experiencia práctica de construcción de sistemas de información lo corrobora.

A menudo surge la necesidad de almacenar estructuras de grafos en "forma pura", por ejemplo, cuando en las bases de datos se usan los modelos de datos basados en grafos. Los objetos de datos se almacenan separados de sus relaciones que se representan por grafos. Se conocen varios procedimientos de prefijación de los grafos: empleando matrices de incidencias y de adyacencia, enumerando entornos de los vértices. Se puede mostrar que el último procedimiento es el más ahorrativo para los grafos de grandes dimensiones, lo que es típico para la práctica. Pero la prefijación del grafo mediante la enumeración de entornos es equivalente a la prefijación del mografo, cuyo portador es el del grafo y las palabras son los entornos de sus vértices. La prefijación del grafo puede tener variantes, pero en cualquier caso un mografo puede ser una abstracción de representación. Por ejemplo, en las bases de datos que parten de un modelo de la red de datos (un fragmento del esquema de la red de datos se refleja en la fig. 5.29, a), la abstracción de los datos es un grafo orientado. Tratando las demandas, la búsqueda de información se realiza en el sentido indicado por los arcos (de los datos sobre los procesos a los de equipo), por esta razón hace falta almacenar solamente los entornos positivos (los del nivel inferior del grafo dado en la fig. 5.29, b). Esta información se da por el mografo representado en la fig. 5.29, c.

Como se ha mostrado anteriormente, el mografo es una abstracción del sistema de búsqueda informativa con un conjunto fijado de demandas. En calidad de otro ejemplo de la representación de un sistema informativo por un mografo sirve una organización de ficheros con varias claves de acceso. El fichero es una sucesión de inscripciones compuestas de campos idénticos (las inscripciones pueden ser de longitud desigual). Llámase *clave* un campo o conjunto de campos, cuyos valores identifican las inscripciones. El acceso al fichero se realiza indicando el valor de la clave. Para acelerar el procesamiento de las demandas se organizan índices (tablas en las cuales para cada valor de la clave se indican las direcciones de las inscripciones con este valor). Si el índice almacena las direcciones de todas las inscripciones con el valor dado, esta organización se denomina *listas invertidas*. La información almacenada en las listas invertidas se representa por un mografo, cuyo portador se compone de un conjunto

<i>dirección</i>	<i>Nombre y apellido</i>	<i>Título</i>	<i>Datos de salida</i>			<i>Palabras clave</i>	
x_1	A.A. Samarski	<i>Teoría de esquemas en diferencias</i>	M	Nauka	1977	<i>Métodos matemáticos</i>	
x_2	G.I. Marchuk	<i>Métodos de la matemática de cálculos</i>	M	Nauka	1977	<i>Métodos matemáticos, simulación matemática</i>	
x_3	I.M. Makárov (red.)	<i>Fundamentos de la automatización del control de la producción</i>	M	Vischaya shkola	1983	<i>Métodos matemáticos, procesos de producción, automatización</i>	
x_4	Swamy M., Thulasiramank.	<i>Gratos, redes, algoritmos</i>	M	Mir	1984	<i>Métodos matemáticos, modelos de la teoría de grafos, algoritmos sobre grafos</i>	
x_5	Breuer M.	<i>Teoría y métodos de automatización de la proyección de cálculo</i>	M	Mir	1977	<i>Automatización, algoritmos sobre grafos</i>	
x_6	V.V. Rzhhevski	<i>Procesos de trabajo a cielo</i>	M	Nedra	1978	<i>Procesos de producción</i>	
x_7	Peterson J.	<i>Teoría de redes de Petri, y simulación de sistemas</i>	M	Mir	1984	<i>Simulación matemática, modelos, de la teoría de grafos</i>	

Fig. 5.30

de las direcciones de inscripciones y las palabras, de los conjuntos de las direcciones de inscripciones que tienen valores idénticos de la clave.

Examinemos un fichero de un sistema informativo de biblioteca (fig. 5.30) que tiene varias claves de acceso: el apellido del autor, el nombre de la Editorial, la palabra clave, etc. Para acelerar el acceso se puede ordenar el fichero por una de las claves (habitualmente, por el apellido). Por otras claves, las inscripciones estarán completamente desordenadas. En el ejemplo a examinar, el índice por la clave del acceso "palabra clave" se representa por el mografo dado anteriormente (véase fig. 5.29, c), a la palabra M_1 , le corresponde el valor "métodos matemáticos", a la M_2 , "automatización", etc.

El mografo puede fijar tanto las listas invertidas, como otras organizaciones de lista para los índices: listas múltiples, listas de secciones. En fin, una matriz binaria sencilla puede representarse como la matriz de incidencia del mografo.

Para el emplazamiento de los datos en la memoria del ordenador los criterios principales son la minimización de la capacidad de memoria y la del tiempo de acceso. Los elementos del portador del mografo corresponden unívocamente a los objetos de los datos almacenados en la memoria. Por lo tanto, el criterio de la minimización de la capacidad de memoria determina la funcional de la calidad $\varphi(\Psi_b)$, en la equivalentización

semántica, como el mínimo de la potencia del portador Ψ_b . El mografo Ψ_b determina el emplazamiento, en el cual se puede hallar todas las palabras lo más posible. Si el mografo Ψ_a no admite este emplazamiento, la búsqueda de cierta palabra es equivalente a la de varias palabras que significa la desintegración de la palabra dada en varias palabras. De tal modo, el criterio de minimización del tiempo de acceso determina la funcional de la calidad $\varphi(\Psi_b)$ como el mínimo de la potencia de la signatura Ψ_b .

En los ejemplos considerados de la organización de los datos representados por el mografo es importante minimizar la capacidad de memoria en condiciones del tiempo mínimo de acceso. Es posible el criterio inverso: la minimización del tiempo de acceso contando con la capacidad constante de memoria. Por ejemplo, es importante en el problema de una ordenación de las inscripciones del fichero (fig. 5.31), con la cual se realizaría más rápido la búsqueda por la clave.

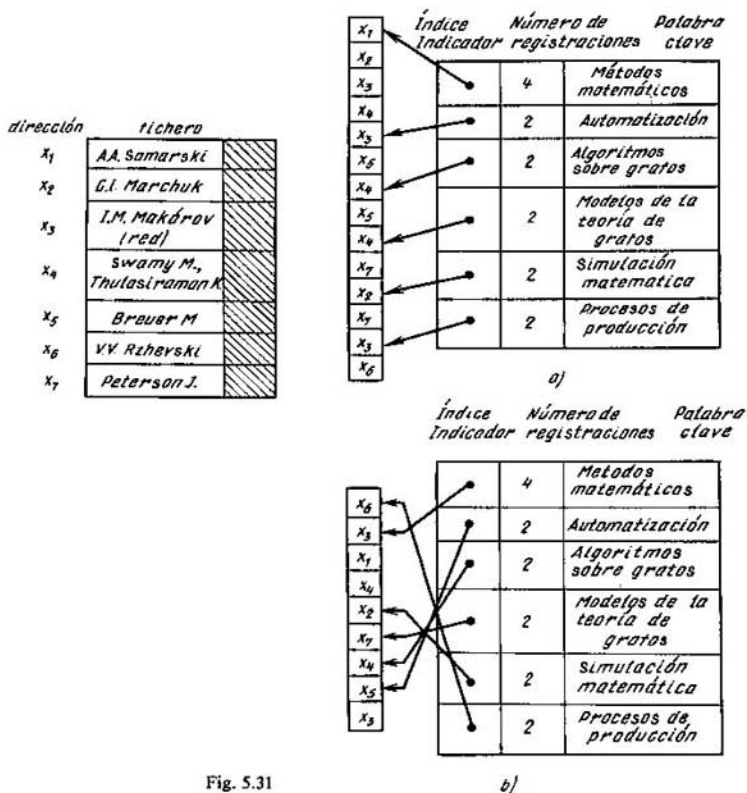
El acceso a los objetos de datos se realiza con ayuda de un procedimiento estándar de búsqueda, en la cual se cumple la transición de un elemento de memoria a otro. La transición se realiza unívocamente, por eso se puede formalizar este procedimiento de búsqueda mediante una función de examen de la memoria S que es una función parcial sobre el conjunto de los elementos del portador de un mografo $S: X \rightarrow X$; $S(x)$ es un elemento, al cual se realiza la transición después del elemento x . Analicemos tal emplazamiento de los datos y tal función de examen de la memoria S (modelo Ψ_b), para los cuales existe un elemento $x_0 \in M$ tal que

$$M = \{x_0, S(x_0), S^2(x_0), \dots, S^{|M|-1}(x_0)\}$$

para cada palabra M del modelo Ψ_b , es decir, cada palabra se busca con ayuda de la función S basándose solamente en la información de la potencia de la palabra y del elemento inicial x_0 .

Semejantes mografos se denominan *admisibles*. Es obvio que en la práctica la mayoría de los mografos no son admisibles. Para su realización en la memoria del ordenador se necesita desintegrar los elementos del portador o las palabras. Al mismo tiempo, se aumenta ora la capacidad de memoria, ora el tiempo de acceso. En la memoria del ordenador, los mografos admisibles se representan de modo sin exceso y requieren el tiempo mínimo de acceso.

De tal modo, el problema del emplazamiento óptimo de los datos consiste en la transformación del mografo en el admisible y en la construcción de la función de examen de la memoria. Con ello, la funcional de la calidad es la extensión minimal del portador sin cambiar la signatura o la extensión minimal de la signatura sin cambiar el portador. Este problema puede interpretarse por grafos. Consideremos la función de examen de la memoria $S: X \rightarrow X$ de un mografo admisible Ψ_b como la relación de adyacencia $S \subseteq X \times X$ de los vértices en el grafo orientado $G_f = \langle X, S \rangle$ construido



sobre el conjunto de vértices X . El grafo G_f es funcional (f -grafo), o sea, de cada uno de sus vértices no sale más que un arco.

Teorema 5.6. *Un f -grafo conexo es acíclico o contiene exactamente un ciclo.*

De esta afirmación se desprende que en la práctica son importantes las siguientes clases de f -grafos y mografos admisibles que se representan por los f -grafos correspondientes: lineales (camino), cíclicos (circuitos), acíclicos (árboles orientados).

En la práctica, la función de examen de la memoria se realiza habitualmente empleando las marcas o mediante la transición al elemento adyacente

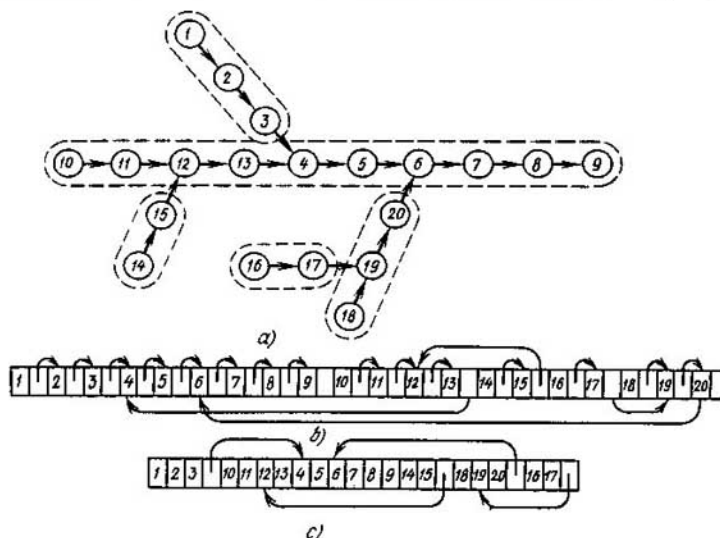


Fig. 5.32

de la memoria. Para representar en la memoria los f -grafos lineales se puede utilizar la segunda variante de la realización de la función de examen de la memoria. La representación de los f -grafos cíclicos se basa también en la transición al elemento adyacente de la memoria, excepto $S(x_n) = x_1$, donde x_1, x_n son los elementos primero y último, respectivamente, en el emplazamiento. Para representar los f -grafos acíclicos son necesarias las marcas. Sin embargo, la representación de un f -grafo acíclico mediante los pares $(x, S(x))$ para cada vértice x es excedente (fig. 5.32, a, b). Para la representación sin exceso, partamos el f -grafo en fragmentos lineales, dentro de los cuales emplearemos como la función de examen la transición al elemento adyacente de la memoria e indicaremos la relación entre los fragmentos mediante las marcas (fig. 5.32, c).

Para resolver los problemas de caracterización de la prefijación de los mografos por f -grafos de diferentes clases, introduzcamos relaciones de subordinación. Un mografo Ψ_1 se subordina a otro Ψ_2 en cuanto a una relación de subordinación P_s^1 si se obtiene del Ψ_2 mediante una sucesión de limitaciones, contracciones, extensiones y convoluciones. Como una limitación del mografo se comprende la eliminación de algunas de sus palabras, como una contracción se comprende la eliminación de unos elementos del

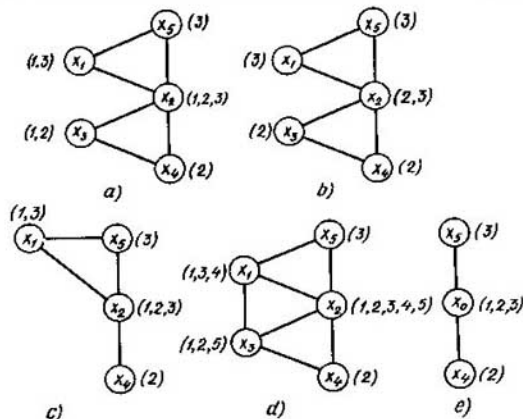


Fig. 5.33

portador, como una extensión se comprende la introducción de una palabra nueva que es la intersección de las ya existentes, como una convolución se comprende el encolamiento de todos los vértices de cierta palabra con la unión de sus pesos. La figura 5.33 (a, b, c, d, e) ilustra las operaciones introducidas. La relación de subordinación P_1^1 se usa para caracterizar morfatos lineales y acíclicos.

La relación de subordinación P_2^2 , salvo las operaciones usadas en P_1^1 , incluye otra operación: la adherencia de la palabra, es decir, su sustitución por el complemento en el conjunto de los elementos del portador del

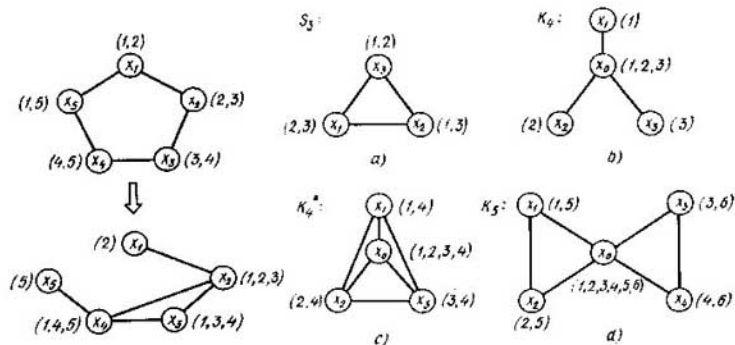


Fig. 5.34

Fig. 5.35

mografo (fig. 5.34). Además, impone la siguiente restricción sobre la operación de extensión P_s^2 : se puede incluir solamente aquella palabra M que es la intersección de palabras M_i, M_j tales que su unión no forma todo el portador del mografo (la extensión posible). Esta relación de subordinación se usa para caracterizar los mografos cíclicos.

El principio de localidad se cumple para las propiedades de linealidad y de aperiodicidad cuando se tiene la relación P_s^1 y para las propiedades de periodicidad cuando se tiene la relación P_s^2 .

Teorema 5.7. (teorema de V. L. Tórvov). *Para las clases de admisibilidad de los mografos las figuras prohibidas son los siguientes mografos: 1) S_3 y K_4 , para la propiedad de linealidad y para la relación de subordinación P_s^1 ; 2) K_4 , para la propiedad de periodicidad y para la relación de subordinación P_s^2 ; 3) S_3, K_4^*, K_5 , para la propiedad de aperiodicidad y para la relación de subordinación P_s^1 .*

En las fig. 5.35 (a, b, c, d) se aducen los mografos S_3, K_4, K_4^*, K_5 .

Los precedimientos de transformación de estas figuras prohibidas en las permitidas consisten o sea en la desintegración de un elemento del portador x y la correspondiente partición del conjunto de las palabras $E(x)$, que lo comprendían, en dos conjuntos $E_1(x)$ y $E_2(x)$ (designaremos este procedimiento mediante $x(E_1, E_2)$), o sea en la desintegración de la palabra M_i en dos M_i' y M_i'' con la correspondiente distribución de los elementos por estas palabras (lo designaremos mediante $i(M_i', M_i'')$). Los procedimientos de la transformación de las figuras prohibidas en las designaciones aceptadas se unen en la tabla 5.15.

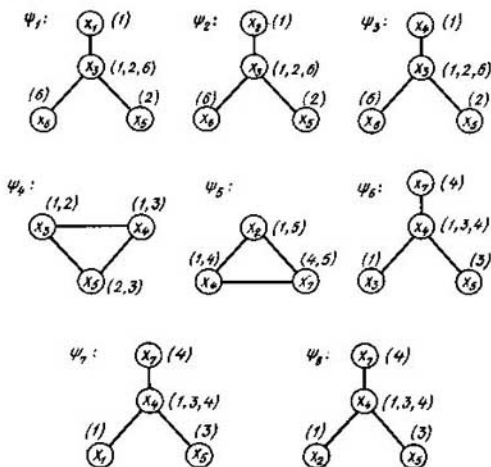


Fig. 5.36

Tabla 5.15

Figura prohibida	Procedimiento de transformación	
	extensión del portador	extensión de la signatura
S_3	$x_1(2, 3)$ $x_2(1, 3)$ $x_3(1, 2)$	$1(x_2, x_3)$ $2(x_1, x_3)$ $3(x_1, x_2)$
K_4	$x_0(3, (1, 2))$ $x_0(2, (1, 3))$ $x_0(1, (2, 4))$	$1(x_0, x_1)$ $2(x_0, x_2)$ $3(x_0, x_3)$
K_4^*	$x_1(1, 4)$ $x_2(2, 4)$ $x_3(3, 4)$ $x_0(1, (2, 3, 4))$ $x_0(2, (1, 3, 4))$ $x_0(3, (1, 2, 4))$	$1(x_1, x_0)$ $2(x_2, x_0)$ $3(x_3, x_0)$ $4(x_1, x_2, x_3, x_0)$ $4(x_2, (x_1, x_3, x_0))$ $4(x_3, (x_1, x_2, x_0))$
K_5	$x_1(1, 5)$ $x_2(2, 5)$ $x_3(3, 6)$ $x_4(4, 6)$ $x_0(1, (2, 3, 4, 5, 6))$ $x_0(2, (1, 3, 4, 5, 6))$ $x_0(3, (1, 2, 4, 5, 6))$ $x_0(4, (1, 2, 3, 5, 6))$	$1(x_1, x_0)$ $2(x_2, x_0)$ $3(x_3, x_0)$ $4(x_4, x_0)$ $5(x_1(x_0, x_2))$ $5(x_2(x_0, x_1))$ $6(x_3(x_0, x_4))$ $6(x_4(x_0, x_3))$
	$x_0(1, 2, 5), (3, 4, 6)$	

Se puede mostrar que estos procedimientos son básicos, o sea, cualquier otro procedimiento de transformación es una superposición de estos procedimientos. Para el modelo en total, los procedimientos dados de las figuras prohibidas no son unívocos. La equivalentización semántica de un mografo Ψ_a en un mografo Ψ_b con las propiedades dadas de admisibilidad se realiza con ayuda de un procedimiento habitual: construimos la tabla semántica; hallamos los cubrimientos; estimamos estos cubrimientos construyendo y colorando grafos especiales. Como resultado obtenemos un mografo que posee la propiedad dada de admisibilidad. Según ella, construimos el f -grafo correspondiente. Consideremos más detalladamente el algoritmo de la construcción de un f -grafo lineal según un mografo lineal (aceptemos que el mografo es conexo).

Determinemos la relación de subordinación P_E sobre el portador de un mografo que $(x_i, x_j) \in P_E$, si $E(x_i) \subset E(x_j)$ (al mismo tiempo, no consideremos las palabras de un elemento). Hallemos el conjunto de elementos minimales X_i de la relación de subordinación P_E . En él dejemos un elemento de cada uno de aquellos que integran las palabras iguales y, después, solamente tales elementos x_i , cuya eliminación junto con $E(x_i)$ no hace inconexo el mografo. (Se puede mostrar que siempre no hay más de dos elementos de este tipo). Sea que se queden los elementos x_i y x_n . Fijemos uno de ellos, por ejemplo, x_n , como un vértice final del f -grafo. Eliminemos otro elemento x_j del mografo y lo introduzcamos en el f -grafo. Si el f -grafo ya tiene vértices, unamos el anterior vértice introducido x_j con x_i por medio de un arco. Continuemos realizando este procedimiento hasta que en el mografo se quede un vértice x_n . Lo tratamos de modo análogo.

Examinemos el proceso de la equivalentización semántica del mografo Ψ_a (véase la fig. 5.29, c) en un f -grafo lineal. Las figuras, prohibidas para la propiedad de linealidad, presentes en Ψ_a , se dan en la fig. 5.36. Si el criterio es la minimización de la funcional $\varphi(\Psi_b)$ igual a la potencia del portador Ψ_b , bajo la condición de no incremento de la potencia de la signatura Ψ_a , se puede aplicar sólo la transformación basada en la desintegración de los elementos del portador. La tabla semántica tiene forma (tabla 5.16):

Tabla 5.16

ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7	ψ_8	
1	1	1	1					$x_3(1, (2, 6))$
1	1	1	1					$x_3(2, (1, 6))$
1	1	1						$x_3(6, (1, 2))$
			1					$x_3(1, 2)$
			1					$x_3(2, 3)$
			1					$x_4(1, 3)$
				1				$x_2(1, 5)$
				1				$x_4(1, 4)$
				1				$x_7(4, 5)$
			1	1	1	1	1	$x_4(1, (3, 4))$
			1		1	1	1	$x_4(3, (1, 4))$
				1	1	1	1	$x_4(4, (1, 3))$

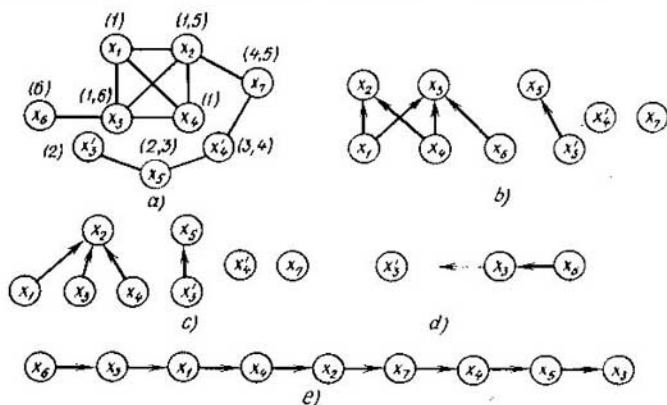


Fig. 5.37

Examinemos el cubrimiento $\pi = \{x_3(2, (1, 6)), x_4(1, (3, 4))\}$. El cumplimiento de estas transformaciones en Ψ_a conducirá al mografo lineal Ψ_b (fig. 5.37, a). Construimos según el algoritmo propuesto, un f -grafo lineal que representa Ψ_b . Definamos la relación de subordinación P_E (fig. 5.37, b). Sus elementos minimales son x_1, x_4, x_6, x'_5 . Eliminando de la consideración x_1 ó x_4 , con la palabra M llegamos al mografo inconexo. Fijamos el elemento x'_5 como el vértice final del f -grafo. En el f -grafo introducimos el vértice x_6 , eliminándolo de Ψ_b . Luego, volvemos a construir la relación de ordenación P_E (fig. 5.37, c). Sus elementos minimales son x_1, x_3, x_4, x'_5 . Los tres primeros integran un conjunto de palabras; escogemos el elemento x_3 y eliminamos los demás de la consideración. Introducimos x_3 en el f -grafo (fig. 5.37, d). Continuando este procedimiento, construimos el f -grafo (fig. 5.37, e).

El mografo Ψ_a representa un sistema de listas invertidas para el fichero de un sistema informativo de biblioteca (véase la fig. 5.30). Si empleamos la organización normal de lista con la función lineal de examen, las listas invertidas (véase la fig. 5.31, a) ocupan 14 elementos de la memoria; si

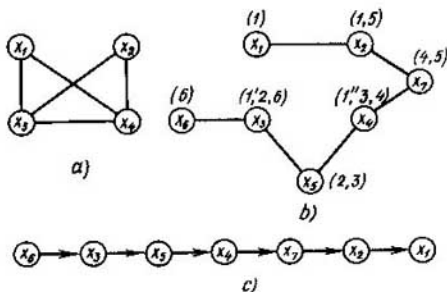


Fig. 5.38

empleamos el emplazamiento óptimo que representa Ψ_b (véase la fig. 5.31, *b*) se ocupan 9 elementos. De tal modo, el ahorro de la memoria sobre las listas invertidas aproximadamente es igual al 35%.

Si el criterio es la minimización de la funcional $\varphi(\Psi_b)$, igual a la potencia de la signatura Ψ_b bajo la condición de no aumento de la potencia del portador Ψ_a , se puede aplicar sólo las transformaciones de las figuras prohibidas que desintegran las palabras. La tabla semántica tiene forma (tabla 5.17):

Tabla 5.17

ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7	ψ_8	
1								$1(x_1, x_3)$
1	1	1	1					$2(x_3, x_5)$
1	1	1						$6(x_3, x_6)$
	1							$1(x_2, x_3)$
		1	1		1			$1(x_3, x_4)$
			1		1	1	1	$3(x_4, x_5)$
				1			1	$1(x_2, x_4)$
					1	1	1	$4(x_4, x_7)$
				1				$5(x_2, x_7)$
						1		$1(x_1, x_4)$

Uno de los cubrimientos que determinan una solución minimal es el cubrimiento no minimal (según el número de transformaciones) $\pi = \{1(x_1, x_3), 1(x_2, x_3), 1(x_3, x_4), 1(x_2, x_4), 1(x_1, x_4)\}$. Un grafo especial para la palabra M (todos los procedimientos desintegran sólo esta palabra) se da en la figura 5.38, *a*. Su coloración con tres colores determina la desintegración minimal $M_1: M_1^l = \{x_1, x_2\}$, $M_1^r = \{x_3\}$, $M_1^m = \{x_4\}$; después de esto el mografo se hace lineal (fig. 5.38, *b*) y se representa mediante el f -grafo lineal (fig. 5.38, *c*).

La solución de este problema permite, conforme al f -grafo construido, ordenar las inscripciones del fichero del sistema informativo de biblioteca (fig. 5.39) de tal modo que las inscripciones con los valores idénticos de la palabra clase se agrupan en el número mínimo de cadenas. Si en el fiche-

dirección Nombre y apellido

x_6	V.V. Rzhewski	
x_3	I.M. Makárov /red.	
x_5	Breuer M.	
x_4	Swamy M., Thulasiramank	
x_7	Peterson J.	
x_2	Б.И. Marchuk	
x_1	A.A. Samarski	

Fig. 5.39

ro inicial (fig. 5.30) había 10 cadenas de este tipo (para la palabra clave “métodos matemáticos” se tiene 1 cadena, para la palabra clave “automatización”, 2 cadenas, etc.), en el emplazamiento obtenido (fig. 5.39) hay 8 cadenas de este tipo (para la palabra clave “métodos matemáticos” se tiene 3 palabras, para las demás, una para cada una). De este modo, como resultado de la optimización durante el procesamiento del fichero, la acción rápida aumentó en el 20%. Para ambos criterios se ha obtenido la solución minimal.

§ 5.6. Problemas y ejercicios

5.1. Determinar la densidad de trabajo y la complejidad capacitiva del algoritmo de la equivalentización sintáctica de un grafo no orientado en el de dos partes mediante la eliminación de aristas, si la funcional de la calidad es el mínimo de las aristas eliminadas.

5.2. Comprobar el cumplimiento del principio de localidad para el problema de caracterización de la transformación de grafo en uno de dos partes y para la relación de subordinación “ser subgrafo” (recordemos que un subgrafo se diferencia de un subgrafo parcial en que si no tiene algún vértice del grafo, no tiene aristas que le son incidentes en el grafo). ¿Forman un conjunto de figuras prohibidas los ciclos de longitud impar?

5.3. Determinar una relación de subordinación que satisfice el principio de localidad para el problema de caracterización de los grafos de Hamilton.

5.4. Determinar una relación de subordinación que satisfice el principio de localidad para el problema de caracterización de los grafos de Euler.

5.5. Proponer algoritmos de la equivalentización sintáctica, heurística y semántica de los grafos no orientados en los de Euler. Comparar la densidad de trabajo y la complejidad capacitiva de algoritmos.

5.6. Cumplir la equivalentización semántica de un grafo G_1 , que es el complemento del grafo dado en la fig. 5.4(b), en el de dos partes $G_2 = \langle V_2, U_2 \rangle$ para los siguientes procedimientos de transformaciones de las figuras prohibidas en las permitidas y de las funcionales de la calidad: a) eliminación de una arista de un ciclo impar, $\varphi(G_2) = \max |U_2|$; b) desintegración de un vértice de un ciclo impar, $\varphi(G_2) = \min |V_2|$.

5.7. Demostrar que al eliminar las filas y columnas que se absorben (véase el § 5.2), lo que se emplea para disminuir la densidad de trabajo para hallar el cubrimiento de la tabla semántica la solución minimal no se pierde.

5.8. Determinar si son inestables las figuras prohibidas los tipos A y B , presentes en el mografo dado en la fig. 5.10, a .

5.9. Demostrar que la solución del problema de la minimización teórica y estructural de la función

$$f(x_1, x_2, x_3, x_4)|_1 = \vee(2, 3, 4, 5, 8, 9, 11, 12, 14, 15)$$

por medio de un procedimiento propuesto en el § 5.4 y basado en la funcional (5.1) es absolutamente minimal.

5.10. Demostrar que la solución del problema de la minimización teórica y estructural del sistema de funciones booleanas, dado por la tabla 5.12, mediante el procedimiento propuesto en el § 5.4 y basado en la funcional (5.2), es absolutamente minimal.

5.11. Determinar la descomposición paralela abstracta del dispositivo automático (véase la fig. 5.24, a), partiendo de la construcción del segundo dispositivo subautomático (que tiene cuatro estados). ¿Cambia el resultado de la descomposición?

5.12. Cumplir la equivalentización semántica del mografo Ψ_a (véase la fig. 5.30, c) en el cíclico Ψ_b con la funcional de la calidad (el mínimo de las desintegraciones) y mediante los siguientes procedimientos de la transformación de las figuras prohibidas en las permitidas: a) desintegración de un elemento del portador, b) desintegración de una palabra.

Comentarios

La lucha contra el sondeo de variantes en la solución de los problemas de la matemática discreta es uno de los más actuales del apoyo matemático moderno de los sistemas de tratamiento de la información. Se puede lograr éxito, sólo resolviendo el problema de caracterización de las transformaciones modelo en realización. Si el problema de caracterización no está resuelto se usa el enfoque heurístico de la optimación de los algoritmos combinatorios.

Bibliografía

- Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, *The design and analysis of computer algorithms*, Reading, Mass., USA, 1976.
- S. E. Goodman, S. T. Hedetniemi., *Introduction to the design and analysis of algorithms*, New York, 1977.
- Michael R. Garey, David S. Johnson., *Computers and Intractability*, San Francisco, 1979.
- Edward Minieka., *Optimization algorithms for networks and graphs*, New York — Basel, 1978.
- Views on general systems theory*, New York — London — Sydney, 1964.
- Edward M. Reingold, Jurg Nievergelt, Narsingh Deo., *Combinatorial algorithms*, Englewood Cliffs, N. J., 1977.
- M. N. S. Swamy, K. Thulasiraman., *Graphs, networks and algorithms*, New York etc.
- Горбатов В. А., Павлов П. Г., Четвериков В. Н., *Логическое управление информационными процессами*. — М.: Энергоатомиздат, 1984. (V. A. Gorbátov, P. G. Pávlov, V. N. Chetverikov., *Gestión lógica de los procesos de información*).
- Левин Д. Я., *Язык сверхвысокого уровня СЕТЛ и его реализация*. Новосибирск: Наука, Сибирское отделение, 1983. (D. Ya. Levin., *Lenguaje de nivel superalto SETL y su realización*).
- Мальцев А. И., *Алгебраические системы*. — М.: Наука, 1970. (A. I. Máltsev., *Sistemas algebraicos*).
- C. Berge, M. Las Vergnas., *Sur un theoreme du type König pour hipergraphes*, Ann. V. Y. Acad. Sci., 1975, № 1, 1970.
- D. Hilbert, P. Bernays., *Grundlagen der Mathematik*, Berlin, 1968.
- Новиков П. С., *Конструктивная математическая логика и точки зрения классической*. — М.: Наука, 1977. (P. S. Nóvikov., *Lógica matemática constructiva desde el punto de vista de la lógica clásica*).
- Автоматизация проектирования сложных логических структур/Под ред. В. А. Горбатова*. — М.: Энергия, 1978 (*Automatización de proyección de las estructuras lógicas complejas*. Dirigido por V. A. Gorbátov).
- Горбатов В. А. *Семантическая теория проектирования автоматов*. — М.: Энергия, 1979 (V. A. Gorbátov., *Teoría semántica del diseño de los dispositivos automáticos*).
- Горбатов В. А., Кафаров В. В., Павлов П. Г., *Логическое управление технологическими процессами*. — М.: Энергия, 1978 (V. A. Gorbátov, V. V. Kafárov, P. G. Pávlov., *Gestión lógica de los procesos tecnológicos*).
- Горбатов В. А., Останков Б. Л., Фролов С. А., *Регулярные структуры автоматного управления/Под ред. В. А. Горбатова*. — М.: Машиностроение, 1980 (V. A. Gorbátov, B. L. Ostánkov, S. A. Frolov., *Estructuras regulares del control de autómatas*. Dirigido por V. A. Gorbátov).
- Лазарев В. Г., Пийль Е. И., *Синтез управляющих автоматов*. — М.: Энергия, 1978 (V. G. Lázarev, E. I. Piil., *Síntesis de los dispositivos automáticos de mando*).
- Поспелов Д. А., *Логико-лингвистические модели в системах управления*. — М.: Энергия, 1981 (D. A. Pospélov., *Modelos lógicos y lingüísticos en los sistemas de mando*).
- Зыков А. А., *Основы теории графов*. — М.: Наука, 1987 (A. A. Zýkov., *Fundamentos de la teoría de los grafos*).

- conexión
 — por aristas 112
 conjunto 11
 — cociente 19, 20
 — finito 11
 — ordenado 22
 — linealmente ordenado 22
 — parcialmente ordenado 22
 — vacío 11
 conjuntos equivalentes 12
 — ordenados isomorfos 24
 constituyente 43
 — co-operación 240
 corte 113
 cota
 — inferior 23
 — — máxima 25
 cota superior 23
 — — mínima 25
 cuantificador
 — existencial 92
 — universal 92
 cubrimiento 49, 128
 cuerda 117
 cuerpo 18

Densidad 134
 derivada
 — de un grafo 122
 — — orden k de una función booleana 76
 — — primer orden de una función booleana 75
 — — una función booleana 75
 — mixta de una función booleana 76
 — en tiempo 250
 descomposición de Shannon 59
 diagrama
 — de Euler 12
 — de Hasse 22
 diámetro de un grafo 110
 dispositivo automático (véase *autómata*)
 distancia 110

Elemento 11
 — comparable 23
 — complementario 28
 — maximal 23
 — máximo 23
 — minimal 23
 — mínimo 23
 — neutral 16
 — — bilateral 16
 — — derecho 16
 — — izquierdo 16
 elemento unidad 16
 entorno 102, 131
 — de radio unidad de un elemento (sección) 20
 enunciación 56
 — falsa 56
 — verdadera 56
 espacio 12
 — booleano 144
 — de ciclos de un grafo 116
 — n -dimensional 116
 espesor 142
 estado de un dispositivo automático 182
 — enganchado 312
 — equivalente 213
 — inestable 231
 — no encadenado 220
 — pseudoequivalente 224

Familia de un conjunto (booleano) 12
 fase
 — de una variable 88
 forma
 — normal
 — — abreviada de Cantor 48
 — — conjuntiva (FNCj) de una función
 — — — perfecta 61
 — — de Cantor 47
 — — disyuntiva de una función
 — — — abreviada 62
 — — — perfecta 58
 — — — tope 64
 — — minimal de Cantor 47
 — — paréntesis 51
 — — perfecta de Cantor 47
 — — tope de Cantor 49
 frecuencia
 — de orden k 127
 — recíproca 122, 127
 — propia 122, 127
 función 13
 — booleana 57
 — — débilmente definida 62
 — — residual 65
 — — completamente definida 13
 — de una lógica de k signos 87
 — n -ádica 14
 — parcialmente definida (incompletamente definida) 13

Género

- de un grafo 141
- de una superficie 141
- grafo 19
 - casi completo 165, 168
 - completo 33
 - conexo 112
 - fuertemente conexo 113
 - no fuertemente conexo 114
 - cubicable 145
 - de borrado 225
 - de dos partes 34, 119
 - de encadenamiento 220
 - de tipo π_0 241
 - de transiciones 205
 - hamiltoniano 111
 - homeomorfo 141
 - isomorfo 105
 - lineal 248
 - parcial 21, 102, 276
 - planar 141
 - ponderado 102
 - reducible 276
 - regular 112
 - trivial 111
- gramática 181
 - de contexto 184
 - de un número finito de estados 181
 - lineal 185
 - metalineal 186
 - unilateralmente lineal 186
 - sin contexto 185
- grupo 17
 - de sustituciones (grupo de Galois) 17
- grupoide 16
 - aditivo 16
 - asociativo 17
 - conmutativo (abeliano) 17
 - idempotente 16
 - multiplicativo 16
- Hipercubo (cubo n -dimensional) 46
- Implicante simple 48, 62
- Intervalo 28
 - de un conjunto 47
 - de una función booleana 62
 - maximal de un conjunto 48
 - — de una función booleana 62
 - nulo de una función booleana 62
 - unitario de una función booleana 62
- isomorfismo 24, 30

Lazo 21

- lenguaje
 - con el número finito de estados 182
 - automático finito 186
 - equivalente 181
- letra
 - proposicional 86
- longitud
 - de una cadena 24
 - de un camino 113
 - — — conjunto ordenado 24
 - — — microrayo 218

Matriz

- básica
 - — ciclomática 118
 - — de cortes (cociclomática) 119
- ciclomática 117
 - de adyacencia 19, 105
 - — incidencia 32
 - — incidencias 105
 - — k células 111
 - modificada de adyacencia 137
- N -dimensional de frecuencia de relaciones 127
- mayorante de un subconjunto 23
- microoperación 206
- microprograma 207
- microrrayo 218
- modelo 35
 - subordinado 275
- modelización de un grafo 33
- mografo 34
 - lineal 279

No entorno 131

- número de un grafo
 - arista de la estabilidad exterior 128
 - — — independencia 128
 - ciclomático 117, 116
 - cromático 156
 - de cubrimiento arista 128
 - — — vértice 128
 - vértice de la estabilidad exterior 128
 - — negativo de la estabilidad exterior 138
 - — positivo de la estabilidad exterior 137
 - — — independencia 128

Orden 91

- de un grafo 165

- Partición de un conjunto** 25
peso de la derivada de una función booleana 78
portador 16
potencia
 — de un conjunto 12
 — una relación 33
 — un vértice 105
problema de caracterización 275
producto
 — cartesiano
 — — de conjuntos 13
 — — de grafos 108
 — — parcial 216
 — de sustituciones 17
Red 115
 — de Petri 254
 — marcada de Petri 255
relación 19
 — binaria 19
 — — de ordenación 21
 — — — estricta 22
 — compatible 35
 — de equivalencia-25
 — de suborden 22
 — *n*-aria 32
 — — simétrica (*S*-relación) 32
 — reflexiva 20
 — simétrica 21
 — transitiva 21
retículo 26
 — completo 26
 — con complementos 29
 — de Dedekind (modular) 28
 — distributivo 28
Sección 20
semigrupo 17
signatura 16
 — completa 88
 — independiente 88
 — no contradictoria 88
sistema
 — de numeración 195
 — — — aditivo 196
 — — — asimétrico 197
 — — — en sentido negativo 197
 — — — en sentido positivo 197
 — — — de la base *S* 196
 — — — de tipo de valores ponderables 196
 — — — multiplicativo 196
 — — — lógico 196
 — — — no posicional 196
 — — — posicional 196
 — — — simétrico 197
subconjunto 11
 — propio 12
subgrafo 33, 102
 — bincidente 217
 — de esqueleto 117
 — parcial 102
 — vacío 128
sustitución 181
Tabla
 — de distinciones 62
 — — Quine (implicante) 49
 — — veracidad 57
terma 92
 — primaria 43, 58
tipo-ádico de una función 13
tipo-ario de una relación 35
transformación
 — equivalente 271
 — idéntica 93
Universo 12
Zung 216

A nuestros lectores:

Mir edita libros soviéticos traducidos al español, inglés, francés, árabe y otros idiomas extranjeros. Entre ellos figuran las mejores obras de las distintas ramas de la ciencia y la técnica, manuales para los centros de enseñanza superior y escuelas tecnológicas, literatura sobre ciencias naturales y médicas. También se incluyen monografías, libros de divulgación científica y ciencia-ficción.

Dirijan sus opiniones a la Editorial Mir, 1 Rizhski per., 2, 129820, Moscú, U-110, GSP, URSS.

Próximas obras de nuestro sello editorial
V. N. Litvinenko, A. G. Mordkóvich
Prácticas para resolver problemas matemáticos
Álgebra y trigonometría

El objetivo de este manual consiste en prestar ayuda concreta a los estudiantes en el desarrollo de los hábitos para resolver problemas matemáticos del curso escolar de álgebra y trigonometría. La presencia del material teórico y ejemplos analizados detalladamente ofrece la posibilidad de utilizar el manual por aquellos que estudian dicho curso individualmente.

K. A. Ríbnikov

Análisis combinatorio

Los matemáticos, ingenieros, así como los especialistas en otras ramas de la ciencia, saben que al solucionar los problemas prácticos con más frecuencia, se ven obligados a ocuparse de las estructuras discretas. Entre éstas citemos grafos, matrices, esquemas-bloque, redes eléctricas, flujos de transporte, sistemas de organización de la producción, flujos de información y muchos otros. Además, como es sabido, el funcionamiento de la mayoría de ordenadores se basa en el principio del cálculo directo.

El presente libro ofrece al lector los fundamentos de esta teoría que mantiene el título históricamente formado, análisis combinatorio.

V. A. Gúsiev, V. N. Litvinenko, A. G. Mordkóvich

Prácticas para resolver problemas matemáticas

Geometría

Este manual se dirige a los estudiantes con el fin de prestarles ayuda concreta en la elaboración y el desarrollo de los conocimientos y hábitos para resolver problemas geométricos según el curso escolar de matemáticas. Contiene material teórico informativo y se examinan ejemplos de resolución de los problemas, lo que permite hacer uso del presente manual para el estudio individual.