

INSTYTUT AUTOMATYKI  
POLSKIEJ AKADEMII NAUK



PRACE  
INSTYTUTU AUTOMATYKI PAN

Zeszyt 92

RYSZARD S. MICHALSKI

SYNTEZA WYRAŻEŃ MINIMALNYCH I ROZPOZNAWANIE  
SYMETRII FUNKCJI LOGICZNYCH



WARSZAWA

1971

KOMITET REDAKCYJNY

Stefan Czarnecki, Ryszard Gawroński,  
Juliusz Lech Kulikowski, Roman Kulikowski,  
Maciej Nałęcz (przewodniczący), Andrzej Straszak,  
Jerzy Thieme (sekretarz)

Ryszard S. Michalski

SYNTEZA WYRAŹEŃ MINIMALNYCH  
I ROZPOZNAWANIE SYMETRII  
FUNKCJI LOGICZNYCH

WSTĘP

Tematem pracy są nowe koncepcje rozwiązania kilku istotnych problemów teorii syntezy automatów, a mianowicie problemu pokrycia przy syntezie minimalnych normalnych wyrażeń logicznych, problemu rozpoznawania symetrii funkcji logicznych oraz problemu wykorzystania informacji o symetrii funkcji do uproszczenia syntezy jej wyrażenia minimalnego.

Problem pierwszy, nazywany także problemem rozwiązywania tablicy prostych implikantów, ma w dziedzinie syntezy automatów cyfrowych ważne znaczenie zarówno teoretyczne jak i praktyczne. Wynika ono między innymi z tego, że problem ten ma charakter podstawowy, dzięki czemu metody rozwiązania go mogą być wykorzystane do rozwiązania szeregu innych zagadnień powstających przy syntezie automatów cyfrowych, a nawet poza tą dziedziną.

Problemy matematycznie równoważne temu problemowi (ogólnie nazywane problemami pokrycia) występują w zagadnieniach syntezy wyrażeń minimalnych funkcji wielowyjściowych /1/, wyrażeń minimalnych funkcji w ważnych dla praktyki systemach złożonych z funkcji "NOR" lub "NAND" /2/, wyrażeń minimalnych dysjunkcyjno-progowych /3/ i szeregu innych tego typu zagadnieniach (np. /4/). Problemy pokrycia występują także przy układaniu testów diagnostycznych dla maszyn cyfrowych i w zagadnieniach podziału zadanego zbioru funkcji realizującego daną funkcję logiczną na takie podzbiory mieszczące się na jednej płycie, aby zapewnić minimalną liczbę połączeń /5/. Powstają również w zagadnieniach rozpoznawania, w za-

gadzeniach sterowania sieciami łączności, w badaniach operacyjnych oraz w niektórych zagadnieniach ekonomicznych.

Uzyskane w niniejszej pracy algorytmy rozwiązania problemu pokrycia opisano w odniesieniu do syntezy minimalnych normalnych wyrażeń logicznych. Jednak, jak pokazano w pktcie 3.7.3, algorytmy te mogą być zastosowane, po odpowiedniej modyfikacji, do rozwiązywania dowolnych problemów pokrycia.

Problem pokrycia w sformułowaniu odnoszącym się do syntezy wyrażeń minimalnych był postawiony po raz pierwszy w 1952 roku przez Quine'a /6/, który opisał go stosując tzw. tablicę prostych implikantów. W sformułowaniu tym problem polega na określeniu w zbiorze wszystkich prostych implikantów danej funkcji logicznej podzbioru, którego suma składników stanowi wyrażenie minimalne funkcji. Ogólne algebrollogiczne rozwiązanie teoretyczne tego problemu podał w 1956 r. Petric /7/. Rozwiązanie to sprowadza się do utworzenia zbioru wyrażeń nieredukowalnych <sup>1)</sup> funkcji (przedstawionych tu w postaci składników dysjunkcyjnego-normalnego wyrażenia tzw. p-funkcji) i następnie na wybraniu spośród nich wyrażenia minimalnego. Rozwiązanie takie jest jednak w praktyce nierealizowalne, przy założeniu dowolności funkcji, już przy niewielkiej liczbie zmiennych. Wynika to z tego, że maksymalna liczba  $\tau(n)$  wyrażeń nieredukowalnych dowolnej funkcji  $n$  zmiennych rośnie drastycznie szybko wraz z wartością  $n$ :

$$5^{2^{n-4}} \leq \tau(n) < \left( \frac{2^{n+1}}{\sqrt{\pi n}} \right) 2^n$$

(oszacowanie podane przez Żurawlewa /8/). Dla  $n = 8$  funkcja  $\tau(n)$  jest większa od  $10^{11}$ , a dla  $n = 10$  osiąga wartość większą od  $10^{44}$ .

To ogólne rozwiązanie problemu stosuje się zwykle po zastosowaniu tzw. podstawowych technik redukcyjnych, upraszczających tablicę prostych implikantów: wydzielenie jądrowych prostych implikantów oraz wykreślenie dominujących kolumn i dominowanych wierszy

1) Wyrażeniem nieredukowalnym nazywa się wyrażenie, w którym skreślenie dowolnego składnika lub dowolnej litery (zmiennnej w postaci prostej lub zamkniętej) powoduje, że wyrażenie przestaje być ekwiwalentne danej funkcji.

/6, 9/. Różne późniejsze ulepszenia rozwiązania Patrica /10, 11 i inne/, jak na przykład metoda rozgałęzień /10/, a także niedawno opublikowane nowe techniki redukcyjne /11-16/, nie wprowadzają jednak istotnego postępu oprócz ułatwień rozwiązania w szczególnych przypadkach.

Odrębny kierunek badań nad rozwiązaniem tego problemu reprezentują prace stosujące metody programowania liniowego w liczbach całkowitych /17, 10, 18-21, 13/ oraz metody teorii gier /10/. Jak wynika z istniejących publikacji, prace te również nie przyniosły wyraźnego postępu.

W świetle ostatnich prac /22-27/ problem pokrycia przy syntezie wyrażeń minimalnych jest w praktyce nierozwiązywalny przy założeniu dowolności funkcji i w jednej klasie algorytmów (tzn. bez korzystania z dodatkowych informacji o funkcji, że na przykład jest symetryczna, liniowa itp.) już przy około 10-11 zmiennych. Ścisłe jego rozwiązanie jest bowiem w ogólnym przypadku związane, jak udowodnił Żurawlew /28/, z przeglądem elementów danego zbioru. A okazuje się, że przegląd ten już przy tej liczbie zmiennych może wymagać liczby operacji nierealizowalnej nawet na najszybszych maszynach matematycznych. W opublikowanej w 1966 r. pracy /25/ Wasiliew skonstruował na przykład funkcje, dla których określenie <sup>1)</sup> wyrażenia minimalnego wymaga już przy 11 zmiennych przeglądu  $10^{18}$  elementów (liczba wyrażeń nieredukowalnych wynosi w tym przypadku  $10^{77}$ ). Wykonanie tej pracy przez maszynę wykonującą 10 min operacji przeglądu na sekundę wymagałoby ok. 3000 lat.

Dlatego powstał problem poszukiwania; z jednej strony - metod dających dobre rozwiązania przybliżone i jednocześnie możliwych do zrealizowania w skomplikowanych przypadkach, a z drugiej strony - możliwie najdogodniejszych metod ścisłego rozwiązania, dotyczących tych przypadków, kiedy teoretycznie jest możliwe uzyskanie rozwiązania bez konieczności przeglądu elementów bardzo licznego zbioru.

Oba powyższe kierunki badań zostały podjęte w niniejszej pracy.

W chwili obecnej istnieje już szereg metod rozwiązań przybliżonych (np. /23, 29-32/), które polegają na określeniu pewnego wyrażenia nieredukowalnego. Żadna z tych metod nie pozwala jednak oszacować odległości otrzymanego rozwiązania do rozwiązania mini-

<sup>1)</sup> Za pomocą tzw. algorytmów lokalnych Żurawlewa (zob. dalej).

malnego. Brak takiego oszacowania stanowi istotną wadę tych metod, ponieważ - jak wykazano w pracy /33/ - wyrażenia nieredukowalne danej funkcji logicznej mogą znacznie różnić się złożonością już nawet przy niewielkiej liczbie zmiennych.

Jednym z wyników niniejszej pracy (pkt 3.7) jest skonstruowanie metody rozwiązania przybliżonego dającej oszacowanie maksymalnej odległości otrzymywanego rozwiązania, nazywanego quasi-minimalnym, do rozwiązania minimalnego. Odległość ta jest wyrażana liczbą składników  $\Delta$  oraz liczbą liter  $\delta$ . Okazuje się przy tym, że omawiana metoda, wymagająca znacznie zredukowanej ilości operacji, w szeregu przypadków (jak wykazały próby przeprowadzone przez autora) prowadzi do rozwiązań ściśle minimalnych (gdy  $\Delta = 0$  oraz  $\delta = 0$ ). Metodę tę rozszerza się następnie na przypadek ogólnie postawionego problemu pokrycia. Ze względu na dużą redukcję ilości operacji metoda ta może być szczególnie przydatna przy rozwiązywaniu problemów pokrycia o dużej złożoności (na przykład przy dużej liczbie zmiennych i dużej liczbie prostych implikantów).

Dla szeregu funkcji logicznych wyrażenie ściśle minimalne można otrzymać w sposób stosunkowo prosty, bez wykonywania operacji przeglądu lub co najwyżej przez wykonanie przeglądu mocno zredukowanego zbioru wyrażeń nieredukowalnych. Możliwość ta wynika z tego, że w pewnych przypadkach można stwierdzić, czy dany prosty implikant wejdzie do wyrażenia minimalnego (czy jest ekstremalą), czy też nie przez badanie implikantów tylko z pewnego jego otoczenia (rozumianego w najprostszej postaci jako zbiór implikantów przecinających się z nim /34/). Interesujące jest więc teoretyczne wyjaśnienie możliwości takiego lokalnego wyboru prostych implikantów i skonstruowanie praktycznych metod rozwiązania problemu na podstawie takiego lokalnego wyboru.

Pierwsze prace w tym kierunku były podjęte przez Rotha /29, 35/ a następnie ważne wyniki w tej dziedzinie uzyskał Żurawiew /28, 34, 36-38/. Skonstruował on pewne algorytmy lokalne (algorytm A oraz A' /34/) oraz między innymi udowodnił /28/, jak wspomniano poprzednio, że istnieją przypadki, kiedy niemożliwe jest stwierdzenie, czy dany prosty implikant jest ekstremalą, czy nie, bez określenia zbioru wyrażeń nieredukowalnych (lub przynajmniej pewnego ich podzbioru) i porównania tych wyrażeń. Wynika z tego, że rozwiązanie problemu pokrycia wymaga w ogólnym przypadku operacji przeglądu.

Prace Żurawlewa dotyczą wyjaśnienia teoretycznych możliwości lokalnego wyboru prostych implikantów, mniej natomiast uwagi zwraca się w nich na praktyczną realizację odpowiednich algorytmów.

W pktcie 3.4 formuluje się szereg twierdzeń i konstruuje operacje lokalnego wyboru prostych implikantów (oprata na innym podejściu niż stosował Żurawlew) oraz podaje się algorytmy syntezy wyrażeń minimalnych wykorzystujące te operacje. Istotną cechą tych algorytmów jest to, że przy wykonywaniu określonej operacji lokalnej badany podzbiór implikantów generuje się za każdym razem oddzielnie. Takie podejście pozwala uniknąć konieczności pamiętania implikantów zbędnych (wchodzących do wyrażeń redukowalnych), a jednocześnie nie wymaga określenia wszystkich prostych implikantów przed rozwiązywaniem problemu pokrycia. Jak wiadomo, już sam problem generowania wszystkich prostych implikantów stanowi, ze względu na szybki wzrost ich liczby ze wzrostem liczby zmiennych  $n$  (zobacz pkt 5.1.3) silne ograniczenie stosowalności wielu metod (np. metody Quine'a-McCluskeya).

Dla przypadku, gdy za pomocą przyjętych operacji logicznych nie można określić wyrażenia minimalnego, skonstruowano (pkt 3.5) pewną metodę, nazwaną metodą gwiazd rozłącznych. Podstawy teoretyczne tej metody były następnie wykorzystane do skonstruowania wspomnianej poprzednio metody rozwiązania quasi-minimalnego.

Drugi podjęty w pracy problem to problem rozpoznawania symetrii funkcji logicznych, postawiony w sposób ogólny. Chodzi tu mianowicie o rozpoznawanie całkowitej lub częściowej symetrii funkcji względem liter (zmiennych w postaci prostej lub zanegowanej), przy czym zarówno w przypadku, gdy funkcja jest określona całkowicie, jak i w przypadku, gdy jest określona częściowo.

Znajomość symetrycznych własności funkcji logicznej jest ważna dla syntezy układu logicznego z szeregu względów. Przede wszystkim, gdy nie wprowadzamy ograniczeń co do struktury układu, metody projektowania układów realizujących funkcje symetryczne są bardzo proste /90-92/, szczególnie gdy układ buduje się na takich elementach jak przekładniki /90/ lub elementy progowe /92/. Jeśli zaś jesteśmy zainteresowani w skonstruowaniu minimalnego dwuwarstwowego układu logicznego na konwencjonalnych elementach logicznych "lub", "i", "nie" (lub układu pochodnego od tego), to wówczas powstaje problem syntezy wyrażenia minimalnego.

W rozdziale 5 podaje się szereg wyników, umożliwiających znaczne uproszczenie takiej syntezy w przypadku, gdy posiada się informację o symetrii funkcji.

Powyższy fakt ma szczególne znaczenie z tego powodu, że - jak wynika z pracy /52/ - klasa funkcji logicznych posiadających maksymalną liczbę prostych implikantów wśród funkcji danej liczby zmiennych  $n$ , zawiera się w klasie funkcji symetrycznych<sup>3)</sup>. Wykorzystując informację o symetrii można dla tych funkcji, będących potencjalnie "najtrudniejszymi" dla syntezy wyrażenia minimalnego, określić takie wyrażenie w sposób znacznie ułatwiony. Funkcje o maksymalnej liczbie prostych implikantów przy liczbie zmiennych  $n = 3, \dots, 24$  podano w pktcie 5.1.3.

Zastosowanie uzyskanych wyników odnośnie do wykorzystania informacji o symetrii do syntezy wyrażeń minimalnych stwarza również - poza omówionymi poprzednio przyczynami - potrzebę istnienia wygodnej metody rozpoznawania symetrii funkcji logicznych.

Problemowi rozpoznawania symetrii poświęcono wiele prac/93-100/. Zdecydowana większość tych prac dotyczy jednak tylko rozpoznawania całkowitej symetrii funkcji, przy tym funkcji całkowicie określonych. Mukhopadhyay /98/ opisuje metodę, w której dla określenia zbiorów liter całkowitej lub częściowej symetrii (funkcji całkowicie określonej) stosuje się odpowiednio  $n$  lub  $C_n^2$  tzw. karty dekompozycyjnych. Karty dekompozycyjne pozwalają na bardzo proste sprawdzenie symetrii względem określonych par zmiennych, jednak proces rozpisywania na nich funkcji logicznej staje się kłopotliwy ze względu na szybki wzrost funkcji  $C_n^2$  wraz ze wzrostem  $n$  już przy stosunkowo niedużej wartości  $n$ .

Dietzeyer i Schneider /100/ opisują metodę rozpoznawania symetrii, obejmującą również funkcje częściowo określone i wielowyjściowe (jest to pierwsza praca uwzględniająca taki ogólny przypadek), przy czym metoda dotyczy rozpoznawania symetrii tylko względem zmiennych. Metoda ta została opracowana przede wszystkim dla realizacji maszynowej.

W rozdziale 4 opisuje się metodę rozpoznawania całkowitej albo częściowej symetrii względem liter, która obejmuje zarówno funkcje

3) Zakładając, że klasa funkcji symetrycznych obejmuje funkcje nie tylko symetryczne względem zmiennych, ale także funkcje symetryczne względem liter (zob. pkt 4).



całkowicie jak i częściowo określone. Podaje się także sposób zastosowania metody w przypadku funkcji wielowyciowych.

Problemowi wykorzystania informacji o symetrii funkcji do uproszczenia syntezy jej wyrażenia minimalnego poświęcono jak dotychczas mało uwagi /9, 10/. W rozdziale 5 niniejszej pracy podano szereg rezultatów z tego zakresu pozwalających na znaczne uproszczenie takiej syntezy, przede wszystkim w przypadku całkowitej symetrii funkcji.

Opisane w pracy rozwiązania podjętych problemów zostały oparte na wykorzystaniu pewnego dwuwymiarowego modelu topologicznego funkcji logicznej, nazywanego obrazem funkcji, który jest analogiczny do znanych konstrukcji geometrycznych stosowanych do przedstawiania funkcji logicznych. Zastosowanie tego modelu pozwala z jednej strony na łatwą realizację ręczną opisanych algorytmów (dla dowolnej funkcji do ok. 8 zmiennych, a w prostszych przypadkach do ok. 13-14 zmiennych), a z drugiej - na względną prostotę ich zaprogramowania na maszynę cyfrową celem realizacji automatycznej, koniecznej przy większej liczbie zmiennych. Przy realizacji automatycznej powyższemu modelowi odpowiada pewna macierz. Jednocześnie omawiany model umożliwił uzyskanie przejrzystości w opisie algorytmów.

W bibliografii dotyczącej pierwszego problemu, ze względu na dużą obfitość prac z tego zakresu, podano przede wszystkim prace najnowsze i tylko ważniejsze prace starsze, natomiast w odniesieniu do pozostałych podano większość znanych prac. Wzory, definicje, twierdzenia, wnioski, przykłady i rysunki ponumerowano podając numer rozdziału oraz kolejny numer w danym rozdziale.

Autor pragnie złożyć bardzo serdeczne podziękowanie prof. drowi. inż. J. Siwińskiemu za opiekę, prowadzenie tej pracy oraz za wiele cennych uwag i wskazówek. Słowa podziękowania należą się też Jego Współpracownikom z Katedry Automatyki Procesów Przemysłowych na Politechnice Śląskiej za uwagi w dyskusjach prowadzonych podczas referowania pracy na seminariach Katedry. Bardzo serdecznie autor dziękuje również prof. drowi S. Węgrzynowi za faktyczne zainicjowanie tej pracy oraz jako recenzentowi - za wiele cennych uwag o niej.

Szczególną wdzięczność autor wyraża doc. drowi R. Marczyńskiemu - recenzentowi pracy oraz Jego Współpracownikom z Centrum Obliczeniowego PAN za bardzo liczne dyskusje oraz uwagi krytyczne, które były istotnym bodźcem do ulepszenia tej pracy w czasie jej powstawania. Serdeczne podziękowanie składa autor również recenzentowi - doc. drowi A.W. Mostowskiemu za szczegółowe zapoznanie się z pracą oraz za szereg rzeczowych i wartościowych uwag.

Głęboką wdzięczność autor winien prof. M.A. Gawriłowowi, członkowi-korespondentowi AN ZSSR, kierownikowi Laboratorium w Instytucie Automatyki i Telemekhaniki w Moskwie, za zainteresowanie się rezultatami tej pracy i przesłanie o nich swojej bardzo wnikliwej opinii stanowiącej dużą zachętę dla autora.

## 1. WIADOMOSCI OGÓLNE I DEFINICJE PODSTAWOWYCH POJĘĆ

### 1.1. Oznaczenia

$\bar{x}_i$  oznacza negację zmiennej  $x_i$ . Funkcje  $f(x_i) = x_i$  oraz  $f(x_i) = \bar{x}_i$  nazywa się literami, odpowiednio prostą i zanegowaną

zmiennej  $x_i$  i oznacza się  $x_i^{\delta_i}$ ,  $\delta_i \in \{0, 1\}$ , przy czym  $x_i^{\delta_i} = \begin{cases} x_i, & \text{jeśli } \delta_i = 1 \\ \bar{x}_i, & \text{jeśli } \delta_i = 0 \end{cases}$ . Symbole  $x_i^{\delta_i} \vee x_j^{\delta_j}$ ,  $x_i^{\delta_i} \wedge x_j^{\delta_j}$  oznaczają od-

powiednio dysjunkcję i koniunkcję liter  $x_i^{\delta_i}$  oraz  $x_j^{\delta_j}$ . Dysjunkcję i koniunkcję funkcji logicznych  $f_i$ ,  $i \in I$ , oznacza się odpowiednio  $\bigvee_{i \in I} f_i$  oraz  $\bigwedge_{i \in I} f_i$ . Zbiór  $E = \{e_1, e_2, \dots, e_p\}$  oznacza

się symbolem  $\{e_i\}_{i=1}^p$ . Sumę, iloczyn, różnicę i iloczyn kartezyjański

zbiorów  $E_1$  i  $E_2$  oznacza się odpowiednio  $E_1 \cup E_2$ ,  $E_1 \cap E_2$ ,  $E_1 \setminus E_2$ ,  $E_1 \times E_2$ . Sumę i iloczyn zbiorów rodziny  $\{E_i\}_{i \in I}$  oznacza się odpowiednio  $\bigcup_{i \in I} E_i$  oraz  $\bigcap_{i \in I} E_i$ . Zbiór wszystkich ele-

mentów  $e$  spełniających funkcję zdaniową  $\varphi(e)$  oznacza się jako  $\{e : \varphi(e)\}$ . Moc zbioru  $E$  oznacza się  $e(E)$ , zbiór pusty  $\emptyset$ . Negację, sumę, iloczyn, implikację w rachunku zdań oznacza się od-

powiednio:  $\rightarrow, \vee, \wedge, \Rightarrow$ . Kwantyfikatory - szczególny i ogólny - oznaczają się odpowiednio:  $\exists, \forall$ , a ich zaprzeczenie:  $\bar{\exists}, \bar{\forall}$ .

## 1.2. Definicje pojęć podstawowych

Niech  $\Omega_j = (\omega_1, \dots, \omega_n)$ ,  $j \in \{0, 1, \dots, k^n - 1\}$ ,  $\omega_i \in \{0, 1, \dots, k-1\}$ ,  $i = 1, \dots, n$ , oznacza ciąg wartości zmiennych wejściowych

$x_1, \dots, x_n$  taki, że  $j = \sum_{i=1}^n \omega_i k^{n-i}$ .

Niech  $\Omega$  oznacza zbiór wszystkich ciągów  $\Omega_j: \Omega = \{\Omega_j\}_{j=0}^{k^n-1}$ .

Niech  $\Psi_j = (\psi_1, \dots, \psi_m)$ ,  $j \in \{0, 1, \dots, (k+1)^m - 1\}$ ,  $\psi_i \in \{0, 1, \dots, k-1, \pi\}$ ,  $i = 1, \dots, m$ , gdzie  $\pi$  nazywa się wartością obojętną, oznacza ciąg wartości zmiennych wyjściowych  $y_1, \dots, y_m$  taki, że

$j = \sum_{i=1}^m \hat{\psi}_i (k+1)^{m-i}$ , gdzie  $\hat{\psi}_i = \begin{cases} \psi_i, & \text{jeśli } \psi_i = \\ & = 1, \dots, k-1 \\ k, & \text{jeśli } \psi_i = \pi \end{cases}$

Niech  $\Psi$  oznacza zbiór wszystkich ciągów  $\Psi_j: \Psi = \{\Psi_j\}_{j=0}^{(k+1)^m-1}$

Definicja 1.1. Odwzorowanie  $f$  z zbioru  $\Omega$  w zbiór  $\Psi$

$$f: \Omega \longrightarrow \Psi \quad (1.1)$$

nazywa się k-wartościową m-wyjściową funkcją logiczną n zmiennych.

Definicja 1.2. Układy techniczne, które przy wprowadzeniu odpowiedniej interpretacji cyfrowej sygnałów wejściowych i wyjściowych realizują pewną funkcję logiczną  $f$  nazywa się kombinacyjnymi układami logicznymi lub jednostanowymi automatami cyfrowymi.

Funkcję logiczną m-wyjściową można traktować jako zbiór funkcji jednowyjściowych  $f_i(x_1, \dots, x_n)$ ,  $i = 1, \dots, m$ . Załóżmy zatem dla dalszych rozważań, że  $m = 1$ .

Niech  $\Omega^\varphi$ ,  $\varphi \in \{0, 1, \dots, k-1, \pi\}$  oznacza zbiór  $\{\Omega_j: f(\Omega_j) = \varphi\}$ . Jeśli  $\Omega^* = \emptyset$ , to funkcję  $f$  nazywa się całkowicie określoną, jeśli  $\Omega^* \neq \emptyset$ , to częściowo określoną.

Funkcja częściowo określona jest równoważna zbiorowi

$$\bar{\Phi}(f) = \{f_i^*\}_{i=0}^{\pi}$$

gdzie  $r = k^c(SF^*) - 1$ , funkcji całkowicie określonych  $f_1^*$ , powstających przez różne rozbięcia zbioru  $\Omega^*$  na podzbiory  $\Omega^{\#,0}$ ,  $\Omega^{\#,1}$ , ...,  $\Omega^{\#,k-1}$  i przyjęcie, że

$$f_1^{\#}(\Omega^{\#}, \varphi) = \{\varphi\} \quad \text{dla } \varphi = 0, 1, \dots, k-1$$

oraz

$$f_1^{\#}(\Omega \setminus \Omega^{\#}) = f(\Omega \setminus \Omega^{\#})$$

Niech

$$f^{\delta} = f + \delta + 1 \pmod{k}, \quad \delta \in \{0, 1, \dots, k-1\},$$

$$f_d(f_1, f_2) = \max(f_1, f_2), \quad f_k(f_1, f_2) = \min(f_1, f_2),$$

gdzie  $f, f_1, f_2$  - dowolne funkcje logiczne (przy  $m = 1$ ).

Funkcje  $f^{\delta}$ ,  $\delta = 0, 1, \dots, k-2$ , nazywa się funkcjami negacji, funkcję  $f^{\delta}$ ,  $\delta = k-1$  - funkcją prostą, funkcje zaś  $f_d(f_1, f_2)$  i  $f_k(f_1, f_2)$  - odpowiednio funkcją dysjunkcji i koniunkcji, przy czym oznacza się je:  $f_1 \vee f_2$  oraz  $f_1 f_2$ .

Z podanych definicji, jak łatwo dowieść, wynikają następujące równości:

$$f_1 \vee f_1 f_2 = f_1 \tag{1.2}$$

$$f_1(f_1 \vee f_2) = f_1 \tag{1.3}$$

$$\bigvee_{\delta=0}^{k-1} f_1 f^{\delta} = f_1 \tag{1.4}$$

Równości (1.2) i (1.3) nazywa się prawami pochłaniania, a równość (1.4) - prawem sklejanania.

Niniejsza praca dotyczy tylko funkcji dwuwartościowych ( $k = 2$ ) i zasadniczo jednowyściowych ( $m = 1$ ), dlatego terminu "funkcja logiczna" używa się w dalszym ciągu, gdy nie podaje się dalszych określeń, w odniesieniu do tego właśnie przypadku.

Funkcje  $f^{\delta}$ ,  $f_1 \vee f_2$ ,  $f_1 f_2$  są w tym przypadku zwykłymi funkcjami odpowiednio negacji, gdy  $\delta = 0$  (oznaczanej  $\bar{f}$ ), dysjunkcji i koniunkcji, a prawo sklejanania przybiera postać:

$$f_1 f_2 \vee f_1 \bar{f}_2 = f_1 \tag{1.5}$$

Zbiór  $\Phi(f) = \{f_i^{\#}\}_{i=0}^r$ ,  $r = 2^{\alpha(\Omega^{\#})} - 1$ , jest zbiorem funkcji całkowicie określonych  $f_i^{\#}$  powstających przez różne rozbięcia zbioru  $\Omega^{\#}$  na zbiory  $\Omega^{\#,0}$  i  $\Omega^{\#,1}$  i przyjęcie, że

$$f_i^{\#}(\Omega^{\#,0}) = \{0\}, \quad f_i^{\#}(\Omega^{\#,1}) = \{1\}, \quad f_i^{\#}(\Omega \setminus \Omega^{\#}) = f(\Omega \setminus \Omega^{\#})$$

System zupełny funkcji złożony z funkcji negacji, dysjunkcji i koniunkcji oznacza się:  $\{-, \vee, \wedge\}$ . Wyrażenia algebrollogiczne danej funkcji  $f$  w systemie  $\{-, \vee, \wedge\}$  o postaci:

$$f = \bigvee_{j=1}^a \bigwedge_{i \in I_j^d} x_i^{G_i} \quad (1.6)$$

gdzie:  $I_j^d \subseteq \{1, 2, \dots, n\}$ ; moc zbioru  $I_j^d$  nazywa się długością

$$\text{koniunkcji } \bigwedge_{i \in I_j^d} x_i^{G_i};$$

$a$  - liczba składników w wyrażeniu (długość wyrażenia)

oraz

$$f = \bigwedge_{j=1}^b \bigvee_{i \in I_j^k} x_i^{G_i} \quad (1.7)$$

gdzie  $I_j^k \subseteq \{1, \dots, n\}$ ,

$b$  - liczba czynników (członów) w wyrażeniu

nazywa się odpowiednio dysjunkcyjnym normalnym wyrażeniem (d.n.w.) i koniunkcyjnym normalnym wyrażeniem (k.n.w.) funkcji  $f$ . Jeśli moc każdego zbioru  $I_j^d$  lub każdego zbioru  $I_j^k$  jest równa  $n$ , to odpowiednio d.n.w. lub k.n.w. nazywa się kanonicznym.

K.n.w. funkcji można utworzyć znajdując najpierw d.n.w. negacji tej funkcji (tzn. traktując zbiór  $\Omega^1$  jako zbiór  $\Omega^0$  i odwrotnie), a następnie zamieniając w tym wyrażeniu znaki dysjunkcji znakami koniunkcji i odwrotnie oraz zastępując litery proste zanegowanymi, a zanegowane prostymi. Procedura ta wynika bezpośrednio z praw De Morgana:

$$\overline{f_1 \vee f_2 \vee \dots \vee f_c} = \bar{f}_1 \bar{f}_2 \dots \bar{f}_c \quad (1.8)$$

$$\overline{f_1 f_2 \dots f_c} = \bar{f}_1 \vee \bar{f}_2 \vee \dots \vee \bar{f}_c \quad (1.9)$$

gdzie  $f_i$ ,  $i = 1, \dots, c$ , dowolne funkcje logiczne.

Mówi się, że funkcja  $f_1$  implikuje funkcję  $f_2$ , jeśli:

1) w przypadku gdy obie funkcje  $f_1$  oraz  $f_2$  są całkowicie określone, zachodzi relacja

$$\forall (\Omega_j \in \Omega), f_1(\Omega_j) \leq f_2(\Omega_j)$$

2) w pozostałych przypadkach w zbiorze  $\Phi(f_1) \times \Phi(f_2)$  istnieje para funkcji  $(f_1^*, f_2^*)$  spełniających warunek 1.

Mówi się, że koniunkcja  $\alpha_1 = \bigwedge_{i \in I_1} x_i^{6_1^1}$  zawiera w sobie ko-

niunkcje  $\alpha_2 = \bigwedge_{i \in I_2} x_i^{6_1^1}$ , jeśli  $I_1 \supset I_2$  oraz  $6_1^1 = 6_1^2$  dla  $i \in I_2$ .

Definicja 1.3. Koniunkcję  $\alpha = \bigwedge_{i \in I} x_i^{6_1^1}$ ,  $I \subseteq \{1, \dots, n\}$  na-

zywa się prostym implikantem funkcji  $f$ , jeśli implikuje ona funkcję  $f$  oraz nie zawiera w sobie żadnej koniunkcji, która również implikuje funkcję  $f$ .

Koniunkcje  $\alpha^1$  oraz  $\alpha^2$  nazywa się sąsiednimi, jeśli mają postać:

$$\alpha^1 = x_s^{6_s} \alpha, \quad \alpha^2 = x_s^{1-6_s} \alpha$$

gdzie:  $\alpha = \bigwedge_{i \in I} x_i^{6_1^1}$ ,  $I \subseteq \{1, \dots, n\}$

$s \in \{1, \dots, n\}$ ,  $s \notin I$

Koniunkcje sąsiednie, zgodnie z prawem sklejanania (1.5), spełniają równość

$$\alpha^1 \vee \alpha^2 = \alpha \quad (1.10)$$

Definicja 1.4. Funkcję  $f(x_1, \dots, x_n)$  nazywa się symetryczną względem zbioru liter  $X = \{x_i^{6_1^1}\}_{i \in I}$ ,  $I \subseteq \{1, \dots, n\}$ , jeśli:

1) w przypadku gdy  $\Omega^{\#} = \emptyset$  jest inwariantna względem dowolnej permutacji liter zbioru  $X$ ,

2) w przypadku gdy  $\Omega^{\#} \neq \emptyset$ , w zbiorze  $\Phi(f) = \{f_i^{\#}\}$  funkcji

całkowicie określonych istnieje przynajmniej jedna funkcja  $f \in \mathcal{F}_I$  spełniająca warunek 1.

Zbiory liter, względem których dana funkcja jest symetryczna, nazywa się zbiorami liter symetrii lub, krótko, zbiorami symetrii. Zbiory symetrii maksymalne względem inkluzji <sup>4)</sup> nazywa się maksymalnymi zbiorami symetrii. Maksymalny zbiór symetrii posiadający największą moc wśród wszystkich maksymalnych zbiorów symetrii danej funkcji  $f$  oznacza się  $X^m$ . Jeśli  $c(X^m) = n$ , to funkcję  $f$  nazywa się całkowicie symetryczną, a jeśli  $1 < c(X^m) < n$  - częściowo symetryczną.

Definicja 1.5. Elementarnymi funkcjami symetrycznymi nazywa się funkcje o postaci:

$$\begin{aligned}
 S_0(x_1^{b_1}, x_2^{b_2}, \dots, x_n^{b_n}) &= x_1^{1-b_1} x_2^{1-b_2} \dots x_n^{1-b_n} \\
 S_1(x_1^{b_1}, x_2^{b_2}, \dots, x_n^{b_n}) &= x_1^{b_1} x_2^{1-b_2} \dots x_n^{1-b_n} \vee x_1^{1-b_1} x_2^{b_2} x_3^{1-b_3} \dots x_n^{1-b_n} \vee \dots \\
 S_2(x_1^{b_1}, x_2^{b_2}, \dots, x_n^{b_n}) &= x_1^{b_1} x_2^{b_2} x_3^{1-b_3} \dots x_n^{1-b_n} \vee x_1^{b_1} x_2^{1-b_2} x_3^{b_3} x_4^{1-b_4} \dots \\
 &\dots \vee x_n^{1-b_n} \dots \\
 &\vdots \\
 S_n(x_1^{b_1}, x_2^{b_2}, \dots, x_n^{b_n}) &= x_1^{b_1} x_2^{b_2} \dots x_n^{b_n}
 \end{aligned}$$

Definicja 1.5. podaje uogólnienie elementarnych funkcji symetrycznych względem zmiennych na elementarne funkcje symetryczne względem liter. Poniżej podaje się analogiczne uogólnienie dwóch znanych twierdzeń dotyczących funkcji całkowicie symetrycznych względem zmiennych.

Twierdzenie 1.1. Każda funkcja całkowicie symetryczna może być przedstawiona w postaci dysjunkcji elementarnych funkcji symetrycznych posiadających ten sam zbiór symetrii.

Dzięki twierdzeniu 1.1 każdą funkcję całkowicie symetryczną względem liter można zapisać w postaci

$$f = \bigvee_{a_j \in A} S_{a_j}(X) = S_A(X)$$

4)

Przez "zbiór o własności  $W$  maksymalny (minimalny) względem inkluzji" rozumie się zbiór, którego żaden nadzbiór (podzbiór) nie posiada własności  $W$ .

gdzie  $X = \{x_i\}_{i=1}^n$  - zbiór liter symetrii, a  $A = \{a_j\}_{j=1,2,\dots}$ ,  
 $A \subseteq \{0, 1, \dots, n\}$  - zbiór indeksów funkcji całkowicie symetrycz-  
 nych, z których składa się dana funkcja (tzw. zbiór  $a$ -indeksów  
 funkcji). Funkcję  $S_A(x_1, x_2, \dots, x_n)$  zapisuje się w postaci  $S_A^n$ .

Twierdzenie 1.2:

$$S_A(X) = S_{\hat{A}}(\hat{X})$$

gdzie  $\hat{X} = \{\hat{x}_i\}_{i=1}^n$ ,  $\hat{\delta}_i = 1 - \delta_i$

$$\hat{A} = \{\hat{a}_j\}_{j=0}^n, \quad \hat{a}_j = n - a_j$$

Definicja 1.6. Ciągiem charakterystycznym funkcji całkowicie symetrycznej  $S_A(X)$  nazywa się ciąg  $C(A) = (c_0, c_1, \dots, c_n)$ , ta-  
 ki, że

$$c_a = \begin{cases} 1, & \text{jeśli } a \in A \\ 0, & \text{jeśli } a \notin A \end{cases}$$

$a = 0, 1, 2, \dots, n$ .

1.3. Kryteria minimalności wyrażeń funkcji logicznych

Przyjmując określony system zupełny funkcji można w tym syste-  
 mie utworzyć zwykle szereg różnych ekwiwalentnych wyrażeń algebra-  
 icznych danej funkcji logicznej. Każde takie wyrażenie opisuje  
 jednoznacznie pewien układ logiczny realizujący tę funkcję. Ze  
 względów realizacyjnych najbardziej pożądane są wyrażenia opisują-  
 ce układy możliwie najbliższe "optymalnym" w sensie różnych wyma-  
 gań narzucanych przez praktykę. Wymagania te dotyczą zwykle szere-  
 gu różnych parametrów układu, są z reguły przeciwstawne i często  
 mało sformalizowane. Najczęściej jednak sprowadzają się do kilku  
 podstawowych, jak na przykład do wymagań odnośnie do minimalnej  
 liczby funktorów, minimalnej głębokości układu, maksymalnej niez-  
 wodności, przy spełnieniu określonych ograniczeń (na przykład co  
 do maksymalnej ilości wejść na funktory i ich obciążalności).

Powyższym wymaganiom odpowiadają pewne określone parametry wy-  
 rażenia algebraicznego funkcji (na przykład liczba składników,



liczba liter, liczba nawiasów itp.) i dlatego problem syntezy optymalnego układu logicznego sprowadza się do syntezy odpowiedniego wyrażenia algebraicznego. Ze względu na różnorodność kryteriów praktycznych (pomijając brak ich wzajemnego ścisłego formalnego powiązania) określenie takiego wyrażenia wymaga z reguły bardzo wielkiej, praktycznie nierealizowalnej, ilości obliczeń. W związku z tym wprowadza się pewne proste matematyczne kryteria optymalności wyrażen algebraicznych, dotyczące jednego lub dwóch parametrów wyrażenia i dokonuje się syntezy wyrażenia optymalnego w sensie tylko tych kryteriów. Tak otrzymane wyrażenie może służyć następnie jako podstawa do dalszych modyfikacji.

W niniejszych rozważaniach ograniczymy się do krótkiego omówienia najbardziej rozpowszechnionych kryteriów odnoszących się do syntezy wyrażen w konwencjonalnym systemie  $\{-, \vee, \wedge\}$ . Są to następujące kryteria:

1. Kryterium absolutnie minimalnej liczby liter.

Wyrażenie spełniające to kryterium nazywa się wyrażeniem absolutnie minimalnym (lub minimalnym wyrażeniem nawiasowym); oznacza się je  $M^a(f)$ .

2. Kryterium minimalnej liczby liter w d.n.w. funkcji.

Wyrażenie spełniające to kryterium nazywa się minimalnym d.n.w. funkcji  $f$ ; oznacza się je  $M^1(f)$ .

3. Kryterium minimalnej liczby liter przy minimalnej liczbie składników w d.n.w. funkcji.

Wyrażenie spełniające to kryterium nazywa się minimalnym najkrótszym d.n.w. funkcji  $f$ ; oznacza się je  $M^k(f)$ .

Kryterium 1 jest szczególnie przydatne dla równoległoszeregowych układów przekaźnikowych, ponieważ ilość styków w takim układzie jest określona ilością liter w wyrażeniu funkcji w systemie  $\{-, \vee, \wedge\}$ . Kryterium to wprowadził Shannon /39/, który podał też pierwsze oszacowanie funkcji  $L(n)$ , określającej najmniejszą ilość liter, przy użyciu których może być w systemie  $\{-, \vee, \wedge\}$  wyrażona dowolna funkcja  $n$  zmiennych. Ostateczne asymptotyczne oszacowanie funkcji  $L(n)$  podał Lupanow /40/; wyraża je wielkość  $2^n/n$ .

Burkhart /41/ udowodnił, że wyrażenie  $M^a(f)$  nie zawsze może być otrzymane (przez wynoszenie przed nawias) z wyrażenia  $M^1(f)$ .

Kryterium 2 ma również zastosowanie przede wszystkim odnośnie do układów przekaźnikowych. Wyrażenie  $M^1(f)$  opisuje bowiem ukła-

dy przekątnikowe o najmniejszej ilości styków, wśród wszystkich układów realizujących funkcję  $f$  i składających się z równoległych gałęzi zawierających szeregowo połączone styki (tzw. układy klasy  $\Pi$  /42/).

Dla układów elektronicznych złożonych z funkcyj realizujących funkcje systemu  $\{-, \vee, \wedge\}$  największe zastosowanie ma kryterium 3. Wyrażenie  $M^k(f)$  opisuje bowiem układ realizujący funkcję  $f$ , który (w ogólnym przypadku) posiadając głębokość 2 (a więc minimalną, jaka jest potrzebna do realizowania dowolnej funkcji) posiada minimalną liczbę funkcyj i przy tej liczbie funkcyj posiada minimalną liczbę wejść na nie (zakłada się, że sygnały wejściowe odpowiadają literom).

Ze względu na istniejące zwykle ograniczenia (na przykład co do ilości wejść na funkcyj) układy opisywane wyrażeniem  $M^k(f)$  nie są praktycznie realizowane na funkcyj elektronicznych już przy stosunkowo niewielkich wartościach  $n$ . Praktyczne znaczenie wyrażenia  $M^k(f)$  polega jednak na tym, że zwykle stanowi ono podstawę do syntezy wyrażen uwzględniających zadane ograniczenia.

Istniała hipoteza, że kryteria 2 i 3 są równoważne, co oznaczałoby, że zbiór  $M = \{M_1^1(f)\}$  równoważnych wyrażen minimalnych  $M_1^1(f)$  zawiera się zawsze w zbiorze wyrażen najkrótszych  $K = \{K_2(f)\}$ . W 1960 roku Żurawlew udowodnił /43/ niesłuszność tej hipotezy przedstawiając przykład funkcji 11 zmiennych <sup>5)</sup>, dla której  $M \cap K = \emptyset$ . Ostateczne rezultaty badań tego problemu uzyskał w 1967 r. Lin-Sin-Lian /44/, który udowodnił, że

$$\begin{array}{ll} \text{dla } n = 3 & M = K \\ \text{dla } n = 4 & M \subseteq K \\ \text{dla } n \geq 5 & \exists f(x_1, \dots, x_n), M \cap K = \emptyset \end{array}$$

#### 1.4. Ogólna charakterystyka procesu syntezy wyrażen minimalnych $M^1(f)$ albo $M^k(f)$

W procesie syntezy wyrażenia minimalnego  $M^1(f)$  lub  $M^k(f)$  danej funkcji  $f$  wyróżnia się zwykle dwa etapy:

5)

Po zapoznaniu się z wynikiem Żurawlewa autor znalazł przykład funkcji 9 zmiennych mającej taką samą własność (rezultat nie opublikowany).

I. Określenie zbioru  $P(f) = \{\alpha_i\}_{i=1}^P$  wszystkich prostych implikantów danej funkcji.

II. Określenie podzbioru  $P^1 \subseteq P(f)$  lub  $P^k \subseteq P(f)$  takiego, że

$$M^1(f) = \bigvee_{\alpha_i \in P^1} \alpha_i \quad (1.11)$$

$$M^k(f) = \bigvee_{\alpha_i \in P^k} \alpha_i \quad (1.12)$$

W ogólnym przypadku funkcja może posiadać nie jedno wyrażenie  $M^1(f)$  lub  $M^k(f)$ , lecz zbiór  $M^1 = \{M_i^1(f)\}$  albo  $M^k = \{M_i^k(f)\}$  wyrażen równoważnych. Dla ich znalezienia należy zatem określić odpowiednie rodziny podzbiorów  $P_i^1$  lub  $P_i^k$ .

Każdy z powyższych etapów można rozważać jako oddzielny problem.

Problem I. Istnieją dwie podstawowe metody określania zbioru  $P(f)$ :

1. Przez generowanie kolejnych koniunkcji liter i sprawdzanie, czy implikują one zadaną funkcję logiczną /45-47 i inne/.

2. Przez wielokrotne stosowanie prawa sklejanja względem koniunkcji wyrażenia kanonicznego /6, 9, 48/.

Dla przypadków istnienia dużej ilości ciągów  $\Omega_j$ , dla których funkcja posiada wartość obojętną ( $c(\Omega_j) \geq 0$ ), zostały opracowane specjalne algorytmy /49, 50 i inne/. Istnieje również metoda określania prostych implikantów, której podstawą jest teoria consensusu /51/.

Maksymalna liczba prostych implikantów  $p(n)$ , jaką może posiadać dowolna funkcja  $n$  zmiennych

$$p(n) = \max_{f_i} p(f_i) \quad (1.13)$$

gdzie  $\{f_i\}$  - zbiór wszystkich funkcji  $n$  zmiennych,

$$p(f_i) = c(P(f_i))$$

została określona (dla  $n \leq 20$ ) przez Kazakowa /52/.

Rozszerzoną tablicę wartości funkcji  $p(n)$  dla  $n \leq 24$  podano w pktcie 5.1.3.

Problem II. Jest to tzw. problem pokrycia (w sformułowaniu od-

noszącym się do syntezy minimalnych d.n.w.). Przy większej liczbie zmiennych ( $n > 4$ ) problem ten stanowi podstawowy problem syntezy wyrażenia minimalnego ( $M^1(f)$  lub  $M^k(f)$ ).

Krótką charakterystyką niektórych ważniejszych prac poświęconych temu problemowi oraz trudności jego rozwiązania (odnoszących się zarówno do poszukiwania podzbioru  $P^c$  jak i  $P^k$ ) była podana we wstępie. Oprócz omówionych tam prac w spisie literatury podano niektóre inne prace dotyczące tego problemu (lub ogólnie problemu syntezy minimalnych d.n.w.) /53-78, 81-88/. Prace /81-88/ opisują graficzne metody syntezy wyrażeń minimalnych (lub w przybliżeniu minimalnych).

W metodach graficznych problem pokrycia posiada prostą interpretację geometryczną i jest rozwiązywany na podstawie wizualnego połączenia pól pewnej tablicy w podzbiory pól odpowiadające prostym implikantom. Ponieważ w metodach tych (z wyjątkiem metody /88/) proces ten odbywa się nie na podstawie ścisłego algorytmu, lecz drogą prób i błędów, dlatego metody te przy liczbie zmiennych większej niż około 4 prowadzą, poza niektórymi przypadkami szczególnymi, tylko do rozwiązań przybliżonych (zob. pkt 3.2).

W niniejszej pracy problem pokrycia rozważa się w sensie poszukiwania podzbioru  $P^k$ , to znaczy poszukiwania wyrażenia minimalnego  $M^k(f)$ . Dlatego terminu "wyrażenie minimalne" będzie się używać dalej (w przypadku braku bliższego określenia) w odniesieniu do wyrażenia  $M^k(f)$ , zapisując go przy tym prościej jako  $M(f)$ .

## 2. O POJĘCIU OBRAZU FUNKCJI LOGICZNEJ

### 2.1. Konstruowanie tablicy logicznej

Dowolna funkcja logiczna  $f(x_1, \dots, x_n)$  może być zapisana w postaci następującej macierzy:

$$T(f) = \begin{bmatrix} t_{v,h} \end{bmatrix}_{\substack{v=0,1,\dots,2^{\lfloor n/2 \rfloor} \\ h=0,1,\dots,2^{n-\lfloor n/2 \rfloor}-1}} \quad (2.1)$$

gdzie  $t_{v,h} = f(\Omega_j)$ ,  $j = v \cdot 2^{n-\lfloor n/2 \rfloor} + h$   
 $\lfloor n/2 \rfloor$  - część całkowita liczby  $n/2$   
 $\Omega_j$  - ciąg  $j$ -ty wartości zmiennych  $x_1, \dots, x_n$ .

Macierzy  $\hat{T}(f)$  można wzajemnie jednoznacznie przyporządkować pewien dwuwymiarowy model topologiczny funkcji  $f$ . Przed zdefiniowaniem tego modelu wprowadzimy najpierw pojęcie tablicy logicznej  $n$  zmiennych.

Podzielmy dowolny prostokąt na  $2^{\lfloor n/2 \rfloor}$  warstw i  $2^{n-\lfloor n/2 \rfloor}$  kolumn według następujących reguł:

1. W pierwszym kroku prostokąt dzieli się prostą poziomą na 2 warstwy. W kroku  $m$  każdą warstwę otrzymaną w kroku  $m-1$  dzieli się prostymi poziomymi na 2 warstwy. Wykonuje się  $\lfloor n/2 \rfloor$  kroków.

2. Kroki  $\lfloor n/2 \rfloor + 1, \dots, n$  wykonuje się tak jak w punkcie 1, lecz dzieląc prostokąt prostymi pionowymi na kolumny.

Proste dzielące prostokąt w kroku  $i$  ( $i = 1, \dots, n$ ) nazywa się osiami zmiennej  $x_i$ . O powstałych w prostokącie warstwach i kolumnach zakłada się, że są zbiorami punktów nie zawierającymi punktów ani osi zmiennych, ani konturu prostokąta. Przecięcie dowolnej warstwy z dowolną kolumną nazywa się polem. Zbiorowi pól leżących nad osią  $x_1$  przyporządkowuje się literę  $\bar{x}_1$ , a zbiorowi pól leżących pod osią  $x_1$  - literę  $x_1$ . Zbiorowi pól składającemu się z podzbiorów pól leżących nad osiami  $x_i$ ,  $i = 2, \dots, \lfloor n/2 \rfloor$ , i zawartych w podzbiorach pól, na które dzieli prostokąt osie  $x_1, \dots, x_{\lfloor n/2 \rfloor}$ , przyporządkowuje się literę  $\bar{x}_1$ , zbiorowi zaś pól składającemu się z podobnych podzbiorów pól, lecz leżących pod osiami  $x_i$ , przyporządkowuje się literę  $x_1$ .

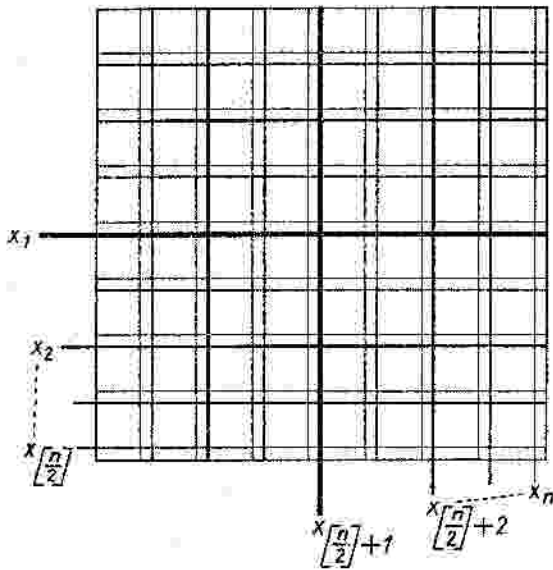
Analogicznym zbiorom pól, składającym się z podzbiorów pól leżących na lewo i na prawo osi zmiennej  $x_i$ ,  $i = \lfloor n/2 \rfloor + 1, \dots, n$ , przyporządkowuje się litery odpowiednio  $\bar{x}_i$  oraz  $x_i$ .

Definicja 2.1. Figurę z opisanym powyżej przyporządkowaniem zbiorów pól liter  $x_1^{\delta_1}, \dots, x_n^{\delta_n}$ ,  $\delta_i \in \{0, 1\}$ , nazywa się tablicą logiczną  $n$  zmiennych.

Na rys. 2.1 przedstawiono tablicę logiczną  $n$  zmiennych, przy czym daną zmienną  $x_i$ ,  $i = 1, 2, \dots, n$ , wpisano obok tylko jednej z osi tej zmiennej.

Na rys. 2.2 zaznaczono jako przykład sposób przyporządkowania zbiorom pól niektórych liter  $x_i^{\delta_i}$  w tablicy logicznej 6 zmiennych.

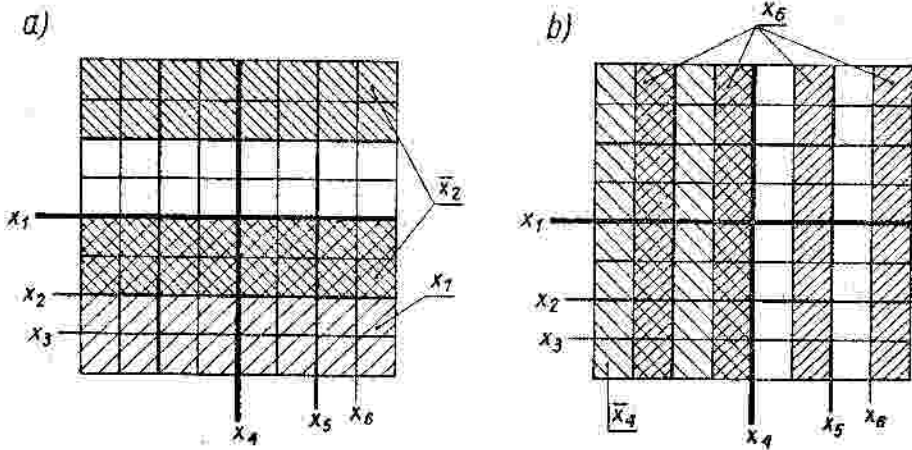
Zdefiniowana powyżej tablica logiczna  $n$  zmiennych jest analogiczna do istniejących już konstrukcji geometrycznych służących do



Rys. 2.1  
Tablica logiczna  
n zmiennych

graficznego przedstawienia funkcji logicznych, a mianowicie takich jak tablice Marquanda /79/, tablice Vanna <sup>6)</sup> /80/, Veitcha /81/ i Swobody /83/. Różni się ona od nich inną interpretacją (litery przyporządkowuje się zbiorom pól,

nie zaś zbiorom punktów, co ułatwia zapis działań w tablicy), wprowadzeniem pojęcia osi zmiennych oraz sposobem oznaczenia.



Rys. 2.2. Zbiory pól w tablicy logicznej 6 zmiennych, którym przyporządkowano: a) litery  $x_1$  i  $\bar{x}_2$ , b) litery  $\bar{x}_4$  i  $x_6$

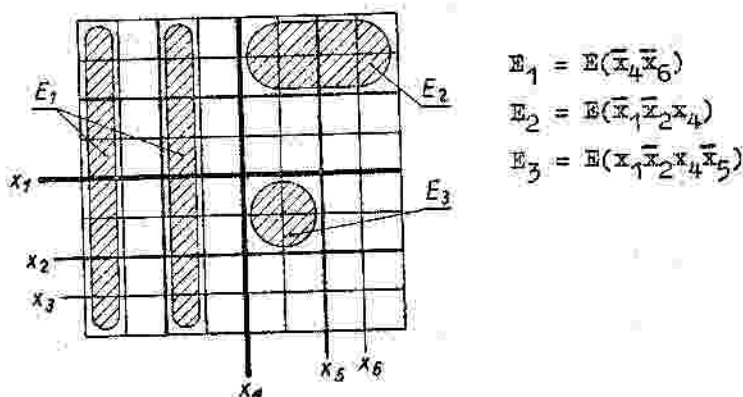
Logicznym operacjom na literach odpowiadają w tablicy logicznej

6) Nie mylić z diagramami Venna.

teoriomnogościowe operacje na zbiorach pól przyporządkowanych tym literom: negacji odpowiada dopełnienie do zbioru wszystkich pól tablicy, dysjunkcji - suma (połączenie) zbiorów pól, koniunkcji - iloczyn (przecięcie) zbiorów pól.

## 2.2. O pojęciu zespołu pól

Definicja 2.2. Zbiór pól tablicy logicznej  $n$  zmiennych utworzony przez przecięcie zbiorów pól przyporządkowanych literom dowolnej koniunkcji  $\alpha = \bigwedge_{i \in I} x_i^{\epsilon_i}$ ,  $I \subseteq \{1, 2, \dots, n\}$ , nazywa się zespołem pól i oznacza się  $E(\alpha)$ .



Rys. 2.3. Przykłady zespołów pól w tablicy logicznej 6 zmiennych

Twierdzenie 2.1. Jeśli koniunkcja  $\alpha$  posiada długość 1, to zespół  $E(\alpha)$  zawiera  $2^{n-1}$  pól.

Dowód. Tablica logiczna  $n$  zmiennych zawiera  $2^n$  pól. Zespół odpowiadający koniunkcji o długości 1 zawiera  $2^{n-1}$  pól. Załóżmy, że zespół odpowiadający koniunkcji  $\alpha$  o długości 1 zawiera  $2^{n-1}$  pól.

Pomnożeniu koniunkcji  $\alpha$  przez dowolną literę  $x_i^{\epsilon_i}$ , której ona nie zawiera, odpowiada zbiór pól o dwukrotnie mniejszej mocy, a więc o mocy równej  $2^{n-1-1}$ . Stąd przez indukcję wynika teza twierdzenia.

Q.E.D.

Definicja 2.3. Liczbę  $r = n - 1$ , gdzie  $l$  - długość koniunkcji  $\alpha$ , nazywa się rozmiarem zespołu  $E(\alpha)$ .

Wniosek 2.1. Zgodnie z twierdzeniem 2.1 zespół rozmiaru  $r$  zawiera  $2^r$  pól.

Przyjmuje się, że jeśli  $\alpha = 0$ , to zespół  $E(\alpha)$  jest zbiorem pustym, a jeśli  $\alpha = 1$ , to jest zbiorem wszystkich pól tablicy logicznej.

Jeśli pewien zbiór pól  $E$  jest zespołem, to koniunkcję jemu odpowiadającą oznacza się  $\alpha(E)$  i mówi się, że  $E$  wyraża się koniunkcją  $\alpha(E)$ . Jeśli  $e \in E$ , to mówi się, że zespół  $E$  pokrywa pole  $e$ .

Twierdzenie 2.2. Maksymalna liczba różnych zespołów w tablicy logicznej  $n$  zmiennych wynosi  $3^n$ .

Dowód. Maksymalna liczba różnych zespołów jest równa, zgodnie z definicją 2.2, maksymalnej liczbie różnych koniunkcji.

$$\alpha = \bigwedge_{i \in I} x_i^{\delta_i}, \quad I \subseteq \{1, 2, \dots, n\}$$

Każdej takiej koniunkcji można przyporządkować ciąg

$$a(\alpha) = (a_1, \dots, a_n)$$

gdzie

$$a_i = \begin{cases} 0, & \text{jeśli } \alpha \text{ zawiera literę } \bar{x}_i \\ 1, & \text{jeśli } \alpha \text{ zawiera literę } x_i \\ 2, & \text{jeśli } \alpha \text{ nie zawiera żadnej litery z indeksem } i \end{cases}$$

( $i = 1, \dots, n$ ).

Maksymalna liczba różnych tego rodzaju ciągów jest równa, jak wiadomo,  $3^n$ . Q.E.D.

Koniunkcji  $n$  liter  $\alpha = x_1^{\delta_1} \dots x_n^{\delta_n}$  odpowiada wzajemnie jednoznacznie pewne pole  $e$  tablicy logicznej  $n$  zmiennych. Koniunkcja  $\alpha$  posiada wartość 1 wtedy i tylko wtedy, kiedy zmienne  $x_i$ ,  $i = 1, \dots, n$ , przyjmują wartości  $\omega_i = \delta_i$ . Pole  $e$  odpowiada więc wzajemnie jednoznacznie ciąg

$$\Omega_j = (\omega_1, \dots, \omega_n), \quad \text{gdzie } \omega_i = \delta_i, \quad j = \sum_{i=1}^n \omega_i 2^{n-i}$$

Definicja 2.4. Numerem pola  $e$  w tablicy logicznej  $n$  zmien-



nych nazywa się liczbę  $\gamma(e) = j$ , gdzie  $j$  jest indeksem ciągu  $\Omega_j$  odpowiadającego polu  $e$ .

0	1	2	3	4	5	6	7
8	9	10	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	53	54	55
56	57	58	59	60	61	62	63

$x_1$  (row 4),  $x_2$  (row 5),  $x_3$  (row 6),  $x_4$  (col 4),  $x_5$  (col 5),  $x_6$  (col 6)

Rys. 2.4. Rozkład numerów pól w tablicy logicznej 6 zmiennych

0	1	1	2	1	2	2	3
1	2	2	3	2	3	3	.
1	2	2	3	2	3	.	.
2	3	3	4	3	.	.	.
1	2	2	3	2	3	3	.
2	3	3	.	3	.	.	.
2	3	.	.	3	.	.	5
3	.	.	.	.	.	5	6

$x_1$  (row 4),  $x_2$  (row 5),  $x_3$  (row 6),  $x_4$  (col 4),  $x_5$  (col 5),  $x_6$  (col 6)

Rys. 2.5. Rozkład wag pól w tablicy logicznej 6 zmiennych

Pole  $e$  o numerze  $\gamma(e)$  oznacza się  $e^{\gamma(e)}$ . Numery  $\gamma(e)$  pól tablicy logicznej  $n$  zmiennych układają się w niej w sposób uporządkowany. Jako przykład, na rys. 2.4 przedstawiono rozkład numerów pól w tablicy logicznej 6 zmiennych.

Definicja 2.5. Waga pola  $e$  tablicy logicznej  $n$  zmiennych nazywa się liczbę  $\delta(e) = \sum_{i=1}^n \omega_i$ , gdzie  $\omega_i, i = 1, 2, \dots, n$ , są elementami ciągu  $\Omega_{\gamma(e)}$ .

Wagi pól posiadają w tablicy logicznej charakterystyczny rozkład (rys. 2.5).

### 2.3. Definicja obrazu $T(f)$

Założmy, że każdemu polu  $e$  tablicy logicznej  $n$  zmiennych została przypisana wartość  $f(\Omega_{\gamma(e)})$ , gdzie  $f(x_1, \dots, x_n)$  jest pewną funkcją logiczną.

Definicja 2.6. Zbiór wszystkich pól tablicy logicznej  $n$  zmiennych z przypisanymi wartościami  $f(\Omega_{\gamma(e)})$  nazywa się obrazem funkcji logicznej  $f(x_1, \dots, x_n)$  i oznacza się  $T(f)$ .

Jeśli polu  $e$  w obrazie  $T(f)$  jest przypisana wartość  $p$ ,  $p \in \{0, 1, \ast\}$ , to mówi się, że pole  $e \in T(f)$  posiada wartość  $p$ . Symbolami  $F^0, F^1, F^\ast$  oznacza się podzbiory pól obrazu  $T(f)$  składające się z pól o wartości odpowiednio  $0, 1, \ast$ . Ze względu na wzajemnie jednoznaczność między działaniami logicznymi na literach a działaniami na odpowiadających im zbiorach pól, obraz  $T(f)$  można łatwo określić na podstawie dowolnego wyrażenia algebrollogicznego funkcji  $f$  w systemie  $\{\neg, \vee, \wedge\}$ . Określanie zaś równoważnych algebrollogicznych wyrażeń funkcji  $f$  na podstawie obrazu  $T(f)$  wymaga umiejętności rozpoznania, które podzbiory zbioru  $F^1$  są zespolami. W pktcie 2.5 sformułowano twierdzenie umożliwiające rozpoznanie zespolów pól przy użyciu zdefiniowanych w pktcie 2.4 pojęć geometrycznych.

#### 2.4. Opis teoretyczny tablicy logicznej

Definicja 2.7. Zbiór pól  $E = E(\alpha)$ , gdzie  $\alpha = \bigwedge_{i=1}^a x_i^{\delta_i}$ ,

$a \in \{1, 2, \dots, \lfloor n/2 \rfloor\}$ ,  $\delta_i \in \{0, 1\}$ , nazywa się warstwą prawidłową o grubości  $h = \lfloor n/2 \rfloor - a$ . Połączenie warstw prawidłowych o grubości  $\lfloor n/2 \rfloor - 1$  nazywa się warstwą prawidłową o grubości  $\lfloor n/2 \rfloor$ .

Definicja 2.8. Zbiór pól  $E = E(\alpha)$ , gdzie  $\alpha = \bigwedge_{i=\lfloor n/2 \rfloor + 1}^b x_i^{\delta_i}$ ,  $b \in \{\lfloor n/2 \rfloor + 1, \lfloor n/2 \rfloor + 2, \dots, n\}$ ,  $\delta_i \in \{0, 1\}$ , nazywa się kolumną prawidłową o grubości  $v = n - b$ . Połączenie kolumn prawidłowych o grubości  $n - \lfloor n/2 \rfloor - 1$  nazywa się kolumną prawidłową o grubości  $n - \lfloor n/2 \rfloor$ .

Definicja 2.9. Przecięcie warstwy prawidłowej o grubości  $h$  z kolumną prawidłową o grubości  $v$  nazywa się prostokątem prawidłowym stopnia  $(h, v)$ .

Twierdzenie 2.3. Prostokąt prawidłowy  $P$  stopnia  $(h, v)$  wyraża się koniunkcją  $\alpha(P)$  o długości  $l_p = n - h - v$ .

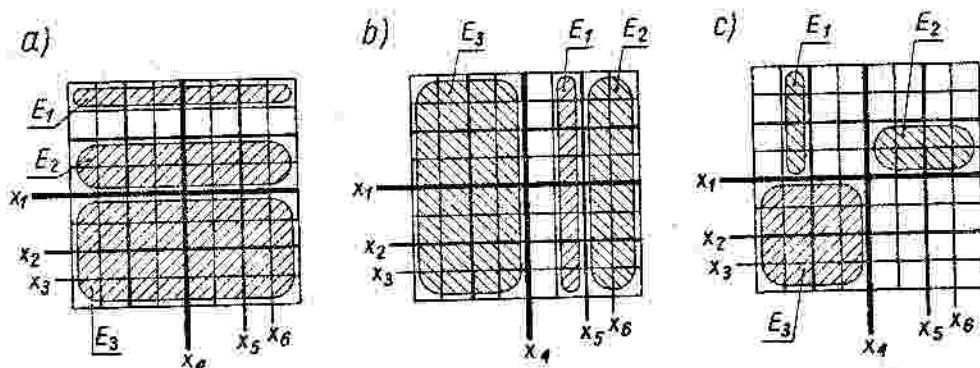
Dowód. Niech  $H$  oraz  $V$  będą odpowiednio pewną warstwą prawidłową o grubości  $h$  i pewną kolumną prawidłową o grubości  $v$ . Zależy równość  $\alpha(P) = \alpha(H) \alpha(V)$ . Ponieważ zbiory liter koniunkcji  $\alpha(H)$  i  $\alpha(V)$  są rozłączne, więc zgodnie z definicjami 2.7 i 2.8:

$$l_p = a + b - \left\lfloor \frac{n}{2} \right\rfloor = n - h - v$$

Q.E.D.

Zgodnie z definicjami 2.2, 2.3 i 2.7-2.9 oraz twierdzeniami 2.1 i 2.3 warstwa prawidłowa o grubości  $h$ , kolumna prawidłowa o grubości  $v$  oraz prostokąt prawidłowy stopnia  $(h, v)$  są zespolami o rozmiarach odpowiednio  $n - \lfloor n/2 \rfloor + h$ ,  $\lfloor n/2 \rfloor + v$  oraz  $h + v$ .

Przykłady powyżej zdefiniowanych elementów tablicy logicznej przedstawiono na rys. 2.6.



$E_1, E_2, E_3$  - warstwy prawidłowe o grubości odpowiednio 0, 1, 2

$E_1, E_2, E_3$  - kolumny prawidłowe o grubości odpowiednio 0, 1, 2

$E_1, E_2, E_3$  - prostokąty prawidłowe stopnia odpowiednio  $(2,0)$ ,  $(1,2)$ ,  $(2,2)$

Rys. 2.6. Przykłady: a) warstw prawidłowych, b) kolumn prawidłowych oraz c) prostokątów prawidłowych

Niech  $E$  oraz  $E^0$  będą dowolnymi zespolami takimi, że  $E \subseteq E^0$ .

Definicja 2.11. Najkrótszą koniunkcję  $\beta$  taką, że  $\beta \alpha(E^0) = \alpha(E)$ , nazywa się koniunkcją adresującą zespół  $E$  w zespole  $E^0$  i oznacza się ją  $\alpha(E, E^0)$ .

Niech dane są zespoły  $E_1$  oraz  $E_2$  takie, że

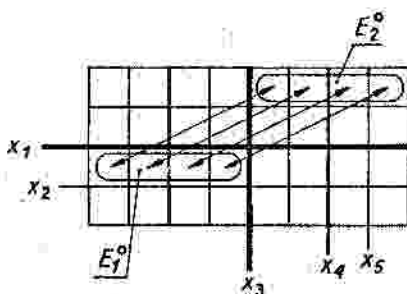
$$\alpha(E_1) = \bigwedge_{i \in I} x_i^{\delta_i^1}, \quad \alpha(E_2) = \bigwedge_{i \in I} x_i^{\delta_i^2}$$

gdzie  $I \subseteq \{1, 2, \dots, n\}$ ,  $\delta_i^1, \delta_i^2 \in [0, 1]$ , oraz zespoły  $E_1^0$  i  $E_2^0$  takie, że

$$\alpha(E_1^0) = \bigwedge_{i \in I^0} x_i^{\delta_i^1}, \quad \alpha(E_2^0) = \bigwedge_{i \in I^0} x_i^{\delta_i^2}$$

gdzie  $I^0 \subset I$ .

Wynika z tego, że zachodzą relacje:  $E_1 \subseteq E_1^0$  i  $E_2 \subseteq E_2^0$ .



$$E_1^0 = E(x_1 \bar{x}_2 \bar{x}_3)$$

$$E_2^0 = E(\bar{x}_1 \bar{x}_2 x_3)$$

Rys. 2.7

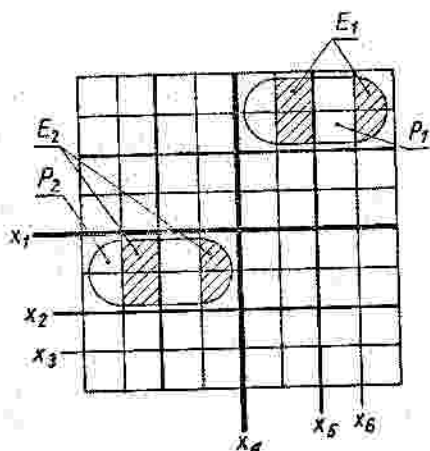
Odpowiadające sobie pola w zespołach  $E_1^0$  oraz  $E_2^0$

**Definicja 2.12.** Jeśli koniunkcje adresujące  $\alpha(E_1, E_1^0)$  oraz  $\alpha(E_2, E_2^0)$  są identyczne, to mówi się, że zespoły  $E_1$  i  $E_2$  mają podobne położenie w zespołach  $E_1^0$  i  $E_2^0$ . Gdy zespoły  $E_1$  i  $E_2$  są pojedynczymi polami, to przy spełnieniu powyższego warunku pola te nazywa się odpowiadającymi sobie polami w zespołach  $E_1^0$  i  $E_2^0$ .

Można łatwo pokazać, że odpowiadające sobie pola w zespołach  $E_1^0$  i  $E_2^0$  pokryją się, jeśli zespoły te nałożymy na siebie drogą równoległego przesunięcia (rys. 2.7). Podobnie, w ogólnym przypadku, pokryją się zespoły  $E_1$  i  $E_2$  po analogicznym nałożeniu na siebie zespołów  $E_1^0$  i  $E_2^0$ , w których mają one podobne położenie (rys. 2.8).

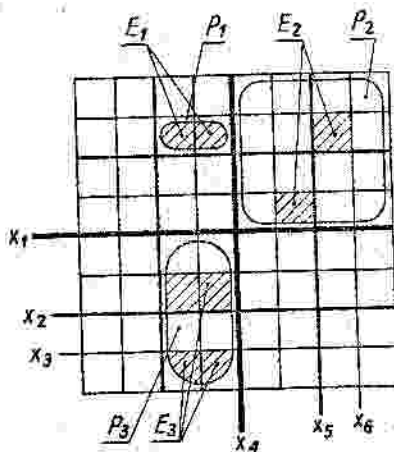
**Definicja 2.13.** Prostokatem prawidłowym obejmującym zbiór pól E nazywa się minimalny względem inkluzji prostokąt prawidłowy P spełniający warunek  $E \subseteq P$ . Prostokąt prawidłowy obejmujący zbiór E oznacza się  $P(E)$ .

Na rys. 2.9 podane przykłady prostokątów prawidłowych obejmujących pewne zbiory pól.



$$P_1 = P(E_1), P_2 = P(E_2), P_3 = P(E_3)$$

Rys. 2.8. Zespoły  $E_1$  i  $E_2$  mają podobne położenie w prostokątach prawidłowych  $P_1$  oraz  $P_2$



Rys. 2.9. Prostokąty prawidłowe  $P_1, P_2, P_3$  obejmujące zbiory pól odpowiednio  $E_1, E_2, E_3$

## 2.5. Twierdzenie o rozpoznawaniu zespołów pól

Na podstawie zdefiniowanych w pktcie 2.4 pojęć sformulujemy twierdzenie pozwalające stwierdzać, kiedy zbiór pól dwóch zespołów jest również zespołem.

Niech  $E_1$  i  $E_2$  będą dowolnymi zespołami tego samego rozmiaru  $r$  oraz takimi, że  $P(E_1) \cap P(E_2) = \emptyset$ .

Twierdzenie 2.4. Warunkiem koniecznym i dostatecznym, aby zbiór  $E = E_1 \cup E_2$  był zespołem rozmiaru  $r + 1$  jest, aby zespoły  $E_1$  i  $E_2$  miały podobne położenie w prostokątach prawidłowych  $P_1 \supseteq E_1$  i  $P_2 \supseteq E_2$  takich, że połączenie ich jest prostokątem obejmującym zbiór  $E : P_1 \cup P_2 = P(E)$ .

Dowód.

Konieczność. Ze względu na warunek  $P(E_1) \cap P(E_2) = \emptyset$  istnieje zawsze co najmniej jedna para prostokątów prawidłowych  $(P_1, P_2)$  takich, że  $P_1 \supseteq E_1, P_2 \supseteq E_2, P_1 \cup P_2 = P(E)$ . Załóżmy, że  $E_1$  i  $E_2$

nie mają podobnego położenia w pewnych prostokątach  $P_1$  i  $P_2$  spełniających powyższe warunki. Oznaczmy koniunkcje adresujące

$$\alpha(E_1, P_1) \text{ oraz } \alpha(E_2, P_2)$$

odpowiednio symbolami  $\alpha_1$  oraz  $\alpha_2$ . Mamy więc  $\alpha_1 \neq \alpha_2$ , a zatem również  $E(\alpha_1) \neq E(\alpha_2)$ . Zbiory  $E(\alpha_1)$  i  $E(\alpha_2)$  spełniają następujące równości:

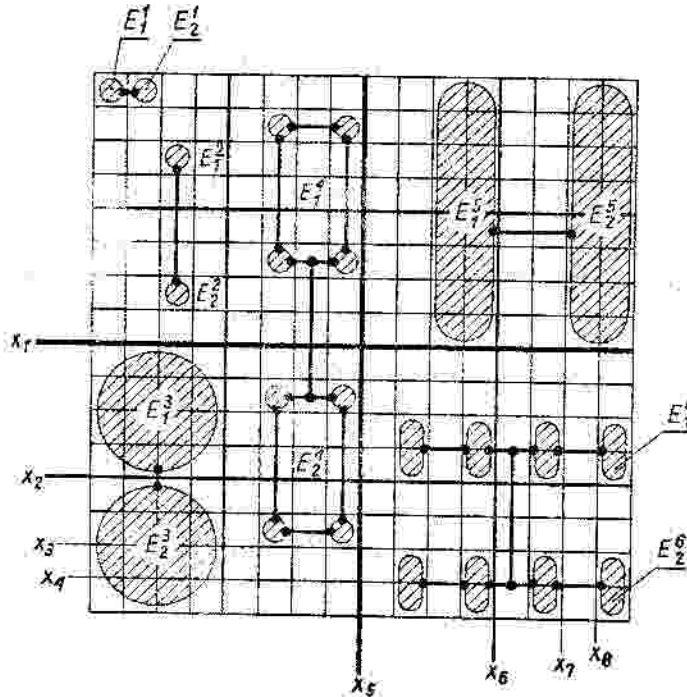
$$E_1 = P_1 \cap E(\alpha_1) \text{ oraz } E_2 = P_2 \cap E(\alpha_2)$$

Jeśli  $E = E_1 \cup E_2$  ma być zespolem, to musi być spełniona równość

$$P_1 \cap E(\alpha_1) \cup P_2 \cap E(\alpha_2) = P(E) \cap E_p$$

gdzie  $E_p$  jest zbiorem wyrażającym się koniunkcją adresującą  $E$  w  $P(E)$ . Ponieważ  $P(E) = P_1 \cup P_2$ , więc

$$P_1 \cap E(\alpha_1) \cup P_2 \cap E(\alpha_2) = P_1 \cap E_p \cup P_2 \cap E_p$$



Rys. 2.10  
 $(E_1^i, E_2^i)$ ,  
 $i = 1, 2, 3,$   
 $4, 5, 6$ , pa-  
 ry zespo-  
 lów speł-  
 niających  
 warunki  
 twierdze-  
 nia 2.4

Uwzględniając że  $E_1 \cap E_2 = \emptyset$ , otrzymujemy, że

$$E(\alpha_1) = E(\alpha_2) = E_P$$

co jest sprzeczne z poprzednim założeniem.

Dostateczność. Jeśli  $E_1$  i  $E_2$  mają podobne położenie w  $P_1$  i  $P_2$ , to

$$\alpha_1 = \alpha_2 \quad \text{oraz} \quad E(\alpha_1) = E(\alpha_2)$$

Zatem

$$E_1 = P_1 \cap E(\alpha) \quad \text{oraz} \quad E_2 = P_2 \cap E(\alpha)$$

gdzie  $E(\alpha) = E(\alpha_1) = E(\alpha_2)$ . Stąd

$$E = E_1 \cup E_2 = (P_1 \cup P_2) \cap E(\alpha)$$

Ponieważ  $P_1 \cup P_2 = P(E)$ , więc  $E = P(E) \cap E(\alpha)$ . Zatem  $E$  jest zespołem. Ponieważ  $P(E_1) \cap P(E_2) = \emptyset$ , więc  $E_1 \cap E_2 = \emptyset$  i  $E$  zawiera  $2 \cdot 2^r = 2^{r+1}$  pól, co zgodnie z wnioskiem 2.1 oznacza, że  $E$  jest zespołem rozmiaru  $r + 1$ . Q.E.D.

Na rys. 2.10 podano przykłady par zespołów spełniających warunki twierdzenia 2.4. Łatwo zauważyć, że pary zespołów spełniających to twierdzenie odpowiadają parom koniunkcji sąsiednich.

Twierdzenie 2.4 pozwala łatwo rozpoznawać, czy dwa dane zespoły tworzą zespół większego rozmiaru, czy też nie. Jeśli więc na początku założyć, że danymi zespołami są pola zbioru  $F^1 \cup F^\#$ , to stosując kolejno twierdzenie 2.4 można określać najpierw pary, następnie czwórki, ósemki itd. pól zbioru  $F^1 \cup F^\#$  tworzące zespoły.

### 3. SYNTEZA MINIMALNYCH I QUASI-MINIMALNYCH POKRYĆ OBRAZU $T(f)$

#### 3.1. Pojęcie pokrycia obrazu $T(f)$ oraz definicje pojęć pomocniczych

Niech dany jest obraz  $T(f)$  i niech  $F^0, F^1, F^\#$  będą, jak definiowano poprzednio, zbiorami pól posiadających wartości odpowiednio 1, 0,  $\#$ .

Definicja 3.1. Pokryciem obrazu  $T(f)$  nazywa się dowolny zbiór

$$D(T) = \left\{ E_i \right\}_{i=1}^P \quad \text{zespołów} \quad E_i \in F^1 \cup F^\# \quad \text{takich, że}$$

$$\bigcup_{i=1}^P E_i \supseteq F^1 \quad (3.1)$$

Definicja 3.2. Złożonością  $z(D(T))$  pokrycia  $D(T) = \{E_i\}_{i=1}^p$  nazywa się sumę

$$z(D(T)) = \sum_{i=1}^p (n - r_i) \quad (3.2)$$

gdzie  $r_i$  oznacza rozmiar zespołu  $E_i$ .

Definicja 3.3. Pokryciem minimalnym najkrótszym obrazu  $T(f)$  nazywa się pokrycie  $M^k(T)$ , które ma minimalną moc i wśród pokryć o takiej samej mocy ma minimalną złożoność.

W niniejszej pracy rozważa się tylko pokrycia minimalne najkrótsze  $M^k(T)$ . Dlatego w dalszym ciągu nazywa się je krótko pokryciami minimalnymi i oznacza symbolem  $M(T)$ . Jeśli dany obraz  $T(f)$  ma więcej niż jedno pokrycie minimalne, to zbiór wszystkich takich pokryć oznacza się  $M(T) = \{M_i(T)\}$ .

Definicja 3.4. Maksymalne względem inkluzji zespoły pól  $E$  spełniające warunki:  $E \subseteq F^1 \cup F^*$ ,  $E \cap F^1 \neq \emptyset$  nazywa się zespołami maksymalnymi w obrazie  $T(f)$ .

Zespoły maksymalne w obrazie  $T(f)$  oznaczają się  $L_i$ ,  $i = 1, 2, \dots$ .

Twierdzenie 3.1. Każde pokrycie minimalne obrazu  $T(f)$  składa się wyłącznie z zespołów maksymalnych.

Dowód. Twierdzenie uważa się za oczywiste.

Q.E.D.

Definicja 3.5. Pokrycie obrazu  $T(f)$  składające się z zespołów maksymalnych i minimalne względem inkluzji nazywa się pokryciem nieredukowalnym.

Pokrycia nieredukowalne obrazu  $T(f)$  oznaczają się symbolem  $N_1(T)$  lub - jeśli wiadomo, o jaki obraz chodzi - symbolem  $N_1$ . Zbiór wszystkich pokryć nieredukowalnych obrazu  $T(f)$  oznaczają się  $N(T)$ .

Z twierdzenia 3.1 oraz definicji 3.3 i 3.5 wynika

Wniosek 3.1. Pokrycia minimalne są elementami zbioru pokryć nieredukowalnych:  $M(T) \subseteq N(T)$ .

Pojęciom pokrycia, złożoności pokrycia, pokrycia minimalnego najkrótszego, zespołu maksymalnego i pokrycia nieredukowalnego odpowiada na gruncie algebriologicznym kolejno: d.n.w. danej funkcji, liczba liter w wyrażeniu, minimalne najkrótsze d.n.w., prosty implikant i nieredukowalne d.n.w.

Jeśli zbiór  $M(T) = \{L_i\}_{i=1}^m$  jest pokryciem minimalnym, to



wyrażenie minimalne funkcji  $f$  jest równe:

$$M(f) = \bigvee_{i=1}^m \alpha(L_i) \quad (3.4)$$

Definicja 3.6. Zespół maksymalny  $L_i$  w obrazie  $T(f)$  nazywa się ekstremalą jądrową, jeśli istnieje pole  $e \in L_i \cap F^1$ , którego nie pokrywa żaden inny zespół maksymalny w obrazie  $T(f)$ .

Definicja 3.7. Zbiór wszystkich ekstremali jądrowych w obrazie  $T(f)$  nazywa się jądrem pokrycia obrazu  $T(f)$ .

Z definicji 3.1, 3.5 i 3.6 wynika

Wniosek 3.2. Jądro pokrycia zawiera się w każdym pokryciu nie-redukowalnym obrazu  $T(f)$ .

Definicja 3.8. Zbiór pól  $e$  należących do zbioru  $F^1$  i zawartych w zespołach jądra pokrycia obrazu  $T(f)$  nazywa się jądrem obrazu  $T(f)$ .

Definicja 3.9. Zespół maksymalny  $L_i$  w obrazie  $T(f)$  nazywa się ekstremalą, jeśli istnieje pokrycie minimalne  $M(T)$  zawierające ten zespół.

Definicja 3.10. Ekstremalę  $L_i$  należącą do pewnego określonego pokrycia  $M(T)$  nazywa się ekstremalą pokrycia  $M(T)$ .

Definicja 3.11. Gwiazdą  $G(e)$  pola  $e \in F^1$  nazywa się zbiór wszystkich zespołów maksymalnych  $L_i$  pokrywających pole  $e$ .

Symbolem  $G^U(e)$  oznacza się zbiór wszystkich pól zbioru  $F^1$ , które są pokrywane przez zespoły gwiazdy  $G(e)$ :

$$G^U(e) = \bigcup_{L_i \in G(e)} L_i \cap F^1$$

Znakiem  $:=$  oznacza się operację podstawienia (jak w języku Algol), która oznacza, że zmienna z lewej strony tego znaku przyjmuje nową wartość wynikającą z operacji zapisanej po prawej stronie znaku. Na przykład wyrażenie  $F^1 := F^1 \setminus E$  oznacza, że nowa wartość zmiennej  $F^1$  będzie zbiorem powstałym przez odjęcie od zbioru będącego jej poprzednią wartością zbioru  $E$ .

Przez  $O^P(E; e_1, \dots, e_k)$  oznacza się operację, która polom  $e_i \in E$  przypisuje indeksy  $i=1, \dots, k$  takie, że  $i_1 < i_2$ , jeśli

$$\gamma(e_{i_1}) < \gamma(e_{i_2})$$

(to znaczy porządkuje pola  $e$  w ten sposób by ich numery były uporządkowane relacją  $\leq$ ). Przez  $O^D(E; e_1)$  oznacza się operację wybrania ze zbioru  $E$  pola o najniższym numerze i przypisania mu oznaczenia  $e_1$ .

### 3.2. Określanie pokryć minimalnych obrazu $T(f)$ w sposób wizualny metodą prób i błędów

Przekształćmy wzór (3.2) określający złożoność pokrycia:

$$z(D(t)) = \sum_{i=1}^p (n - r_i) = pn - \sum_{i=1}^p r_i \quad (3.5)$$

Aby pokrycie  $D(t)$  było pokryciem minimalnym, musi ono posiadać minimalną moc (a więc minimalną wartość  $p$ ), a następnie minimalną złożoność  $z(D(t))$ . Ze wzoru (3.5) widać, że złożoność pokrycia jest tym mniejsza, im większe będą rozmiary zespołów wchodzących do tego pokrycia.

Na podstawie twierdzenia 2.4 łatwo można tworzyć zespoły maksymalne w obrazie  $T(f)$ . Wybiera się pewne pole  $e \in F^1$  i poszukuje się innego pola  $e_2 \in F^1 \cup F^*$ , z którym pole  $e_1$  tworzy zespół rozmiaru 1. Następnie poszukuje się innej pary pól  $(e_3, e_4)$ ,  $(e_3, e_4) \in F^1 \cup F^*$ , tworzącej zespół rozmiaru 1, z którą para  $(e_1, e_2)$  tworzy zespół rozmiaru 2 itd. dopóty, dopóki nie utworzy się pewien zespół maksymalny. Tym sposobem tworzy się zbiór zespołów maksymalnych pokrywających wszystkie pola zbioru  $F^1$ , to znaczy pokrycie obrazu  $T(f)$ .

Posługując się taką metodą można w szeregu przypadków, drogą prób i błędów, ustalić pokrycie minimalne obrazu  $T(f)$ . Tego rodzaju metoda jest stosowana we wszystkich znanych metodach graficznych. W ten sposób, jak wskazuje praktyka, udaje się względnie łatwo określić pokrycie minimalne przy funkcjach do około 4-5 zmiennych (jak przy innych tego typu metodach). Przy funkcjach większej liczby zmiennych, ze względu na brak ścisłego algorytmu wyboru zespołów maksymalnych (których liczba może w tym przypadku osiągnąć już duże wartości, na przykład 92 przy  $n=6$  i 218 przy  $n=7$ ), metoda ta umożliwia - poza niektórymi szczególnymi przypad-

kami - określać tylko pokrycia w przybliżeniu minimalne.

Metodę tę opisano ze względów porządkowych. Okazuje się bowiem, że istnieje pewien algorytm postępowania, który charakteryzuje podobna prostota określania pokrycia, ale jednocześnie w sposób ścisły pozwala stwierdzić, czy otrzymane pokrycie jest minimalne lub - przy braku takiej pewności - dostarcza informację o maksymalnej możliwej różnicy między pokryciem otrzymanym a pokryciem minimalnym (wyrażanej liczbą zespołów maksymalnych i złożonością). Powyższy algorytm jest opisany w pktcie 3.7.

### 3.3. Algorytmy generowania gwiazd $G(e)$

#### 3.3.1. Uwagi wstępne

Poniżej będą sformułowane ścisłe algorytmy tworzenia pokryć minimalnych oraz tzw. pokryć quasi-minimalnych. Jedną z podstawowych operacji, którą stosuje się w tych algorytmach, jest generowanie gwiazd  $G(e)$  pewnych pól  $e \in \mathbb{F}^1$ . W niniejszym punkcie sformułowano dwa algorytmy generowania gwiazd  $G(e)$ ; są one przeznaczone przede wszystkim do realizacji automatycznej. W przypadku gdy liczba zmiennych nie przekracza około 8, gwiazdy  $G(e)$  można łatwo określać ręcznie na podstawie twierdzenia 2.4. Przy tej liczbie zmiennych gwiazdy  $G(e)$  zawierają na ogół - oprócz rzadkich skrajnych przypadków - nie więcej niż kilka zespołów maksymalnych. Dlatego formułowanie ścisłego algorytmu określającego sposób i kolejność tworzenia tych zespołów wydaje się zbędne.

#### 3.3.2. Twierdzenie o maksymalnej liczbie zespołów maksymalnych, jakie może posiadać gwiazda $G(e)$ .

Zanim przejdziemy do opisu algorytmów generacji gwiazd sformułujemy twierdzenie, określające maksymalną możliwą liczbę zespołów maksymalnych, jaką może mieć gwiazda  $G(e)$  w obrazie funkcji logicznej  $n$  zmiennych.

Twierdzenie 3.2. Maksymalna liczba zespołów maksymalnych, jaką może mieć gwiazda  $G(e)$  w obrazie  $T(f)$  dowolnej funkcji  $n$  zmiennych, wynosi  $g(n) = C_n^{\lfloor n/2 \rfloor}$ . Jeśli  $n$  jest liczbą parzystą, to gwiazda  $G(e)$  o mocy  $c(G(e)) = g(n)$  składa się z zespołów maksymalnych rozmiaru  $n/2$ , jeśli  $n$  jest liczbą nieparzystą, to

gwiazda  $G(e)$  składa się z zespołów maksymalnego rozmiaru  $(n-1)/2$  lub  $(n+1)/2$ .

Dowód. Liczba  $g(n)$  jest równa maksymalnej liczbie różnych zespołów pokrywających dowolnie wybrane pole  $e$  tablicy logicznej  $n$  zmiennych, przy czym spełniających warunek, że żadnych dwu zespołów nie można połączyć znakiem inkluzji. Załóżmy, bez straty na ogólności, że wybranym polem  $e$  jest pole  $e^0$ , to znaczy pole o numerze 0. Koniunkcja  $\alpha(e^0) = \bar{x}_1 \bar{x}_2 \dots \bar{x}_n$ . Zespoły  $E_1$  pokrywające pole  $e^0$  odpowiadają koniunkcjom powstającym przez usuwanie różnych liter z koniunkcji  $\alpha(e^0)$ . Istnieje  $2^n - 2$  różnych koniunkcji zawartych w  $\alpha(e^0)$ . Oznaczmy zbiór tych koniunkcji symbolem  $\alpha$ ,  $\alpha = \{\alpha_j\}$ . Parom zespołów, których nie można połączyć znakiem inkluzji, odpowiadają pary koniunkcji  $\alpha_j$  nie spełniających relacji, że jedna z koniunkcji tej pary zawiera się w drugiej. Określmy w zbiorze  $\alpha$  podzbiór koniunkcji o największej mocy, w którym dowolne dwie koniunkcje spełniają powyższy warunek. Oznaczmy symbolem  $X(\alpha_j)$  zbiór liter koniunkcji  $\alpha_j$ . Przyporządkujmy każdej koniunkcji  $\alpha_j \in \alpha$  wektor  $A_j = [a_1^j, a_2^j, \dots, a_n^j]$  gdzie

$$a_i^j = \begin{cases} 1, & \text{jeśli } \bar{x}_i \in X(\alpha_j) \\ 0, & \text{jeśli } \bar{x}_i \notin X(\alpha_j) \end{cases} \quad i = 1, \dots, n$$

Dwa wektory  $A_{j_1}$  i  $A_{j_2}$  nazywa się wektorami porównywalnymi, jeśli:

$$(a) \quad \forall i \in \{1, \dots, n\}, \quad a_i^{j_1} \geq a_i^{j_2}$$

lub

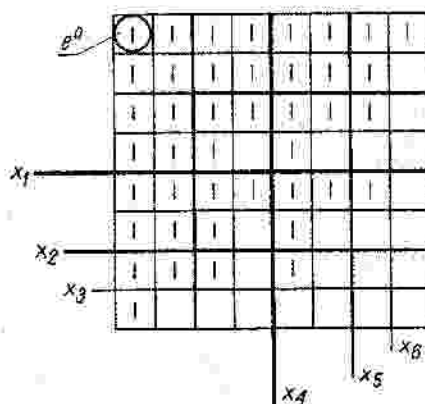
$$(b) \quad \forall i \in \{1, \dots, n\}, \quad a_i^{j_1} \leq a_i^{j_2}$$

Jeśli zachodzi (a), to pisze się, że  $A_{j_1} \geq A_{j_2}$ ; jeśli (b), to  $A_{j_1} \leq A_{j_2}$ . Warunkowi, aby koniunkcja  $\alpha_{j_1}$  nie zawierała się w koniunkcji  $\alpha_{j_2}$  lub odwrotnie, odpowiada warunek, aby przyporządkowane tym koniunkcjom wektory  $A_{j_1}$  i  $A_{j_2}$  nie były porównywalne. Tak więc określenie liczby  $g(n)$  sprowadza się do określenia maksymalnej liczby wektorów nieporównywalnych o długości  $n$ .

Problem ten został rozwiązany w pracy [86], gdzie udowodnio-

no, że wektorów tych jest  $C_n^{\lfloor n/2 \rfloor}$ . Jeśli  $n$  jest liczbą parzystą, to zbiór wektorów nieporównywalnych składa się z wektorów o  $n/2$  współrzędnych różnych od zera, a jeśli  $n$  jest liczbą nieparzystą, to są dwa takie zbiory, a mianowicie zbiory wektorów o  $(n-1)/2$  i  $(n+1)/2$  współrzędnych różnych od zera. Wektorowi  $A_j$  o  $l$  współrzędnych różnych od zera odpowiada koniunkcja  $\alpha_j$  o długości  $l$ , a koniunkcji takiej, zgodnie z twierdzeniem 2.1. i definicją 2.3, odpowiada zespół rozmiaru  $n-l$ . Aby zakończyć dowód, wystarczy zauważyć, że  $C_n^{n-\lfloor n/2 \rfloor} = C_n^{\lfloor n/2 \rfloor}$ ,  $n + \frac{n-1}{2} = \frac{n+1}{2}$  oraz  $n - \frac{n+1}{2} = \frac{n-1}{2}$ . Q.E.D.

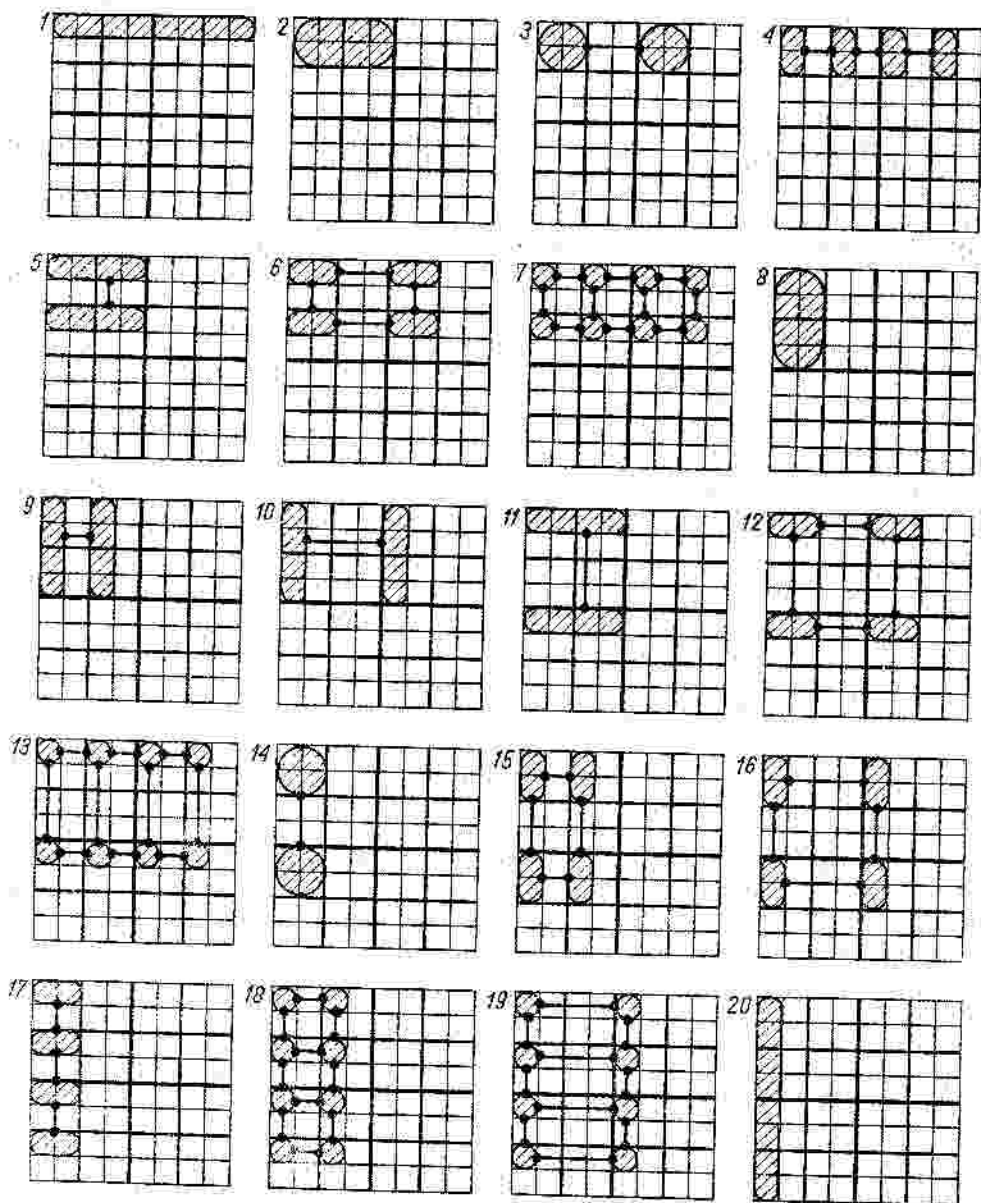
Rys. 3.1. Obraz  $T(f)$ , w którym gwiazda pola  $e^0$  ma maksymalną liczbę zespołów maksymalnych przy 6 zmiennych



Na rys. 3.1 przedstawiono przykład obrazu  $T(f)$  funkcji 6 zmiennych, w którym gwiazda  $G(e^0)$  ma  $g(6) = C_6^3 = 20$  zespołów maksymalnych. Funkcja  $f$  jest funkcją całkowicie symetryczną  $S_{0,1,2,3}^6$ . Zespoły gwiazdy  $G(e^0)$  przedstawiono na rys. 3.2. Są one ekstremalami jądrowymi, a ponieważ pokrywają wszystkie pola zbioru  $F^1$  w obrazie  $T(f)$ , przeto gwiazda  $G(e^0)$  jest jednocześnie pokryciem minimalnym  $M(T)$ .

### 3.3.3. Algorytmy $G^1$ i $G^2$ generowania gwiazdy $G(e)$

Załóżmy, że chcemy określić gwiazdę  $G(e)$  pewnego pola  $e \in F^1 \subseteq \mathbb{T}(f)$ , któremu odpowiada koniunkcja  $\alpha(e) = \bigwedge_{i=1}^n x_i^{\epsilon_i}$ .



Rys. 3.2. Zespoły maksymalne gwiazdy pola  $e^0$   
w obrazie  $T(f)$  przedstawionym na rys. 3.1.

Sformułujemy dwa algorytmy  $G^1$  i  $G^2$  określania gwiazdy  $G(e)$ . Oba algorytmy są uniwersalne, ale algorytm  $G^2$  jest wygodniej stosować wówczas, gdy  $c(F^0)$  nie osiąga dużych wartości, w przeciwnym natomiast przypadku wygodniej jest stosować algorytm  $G^1$ .

Algorytm  $G^1$ . Niech  $\alpha = \{\alpha_j\}_{j=1}^{2^n-1}$  oznacza zbiór zawierający wszystkie koniunkcje zawarte w  $\alpha(e)$  oraz koniunkcję  $\alpha(e)$ , przy czym  $\alpha(e) = \alpha_{2^n-1}$ . Oznaczmy symbolem  $X(\alpha_j)$  zbiór liter koniunkcji  $\alpha_j$ . Przypiszmy każdej koniunkcji  $\alpha_j$  wektor  $A_j = [a_1, \dots, a_n]$ , gdzie

$$a_i = \begin{cases} 1, & \text{jeśli } x_i^{\delta_j} \in X(\alpha_j) \\ 0, & \text{jeśli } x_i^{\delta_j} \notin X(\alpha_j) \end{cases} \quad i = 1, 2, \dots, n$$

Niech  $A = \{A_j\}_{j=1}^{2^n-1}$  oznacza zbiór wszystkich wektorów  $A_j$ , przy czym niech  $j = \sum_{i=1}^n a_i 2^{n-i}$ , gdzie  $a_i$  - elementy wektora  $A_j$ . Oznaczmy symbolem  $L(A_j)$  zespół  $L(\alpha_j)$ . Wykres operacyjny algorytmu  $G^1$  przedstawiono na rys. 3.3.

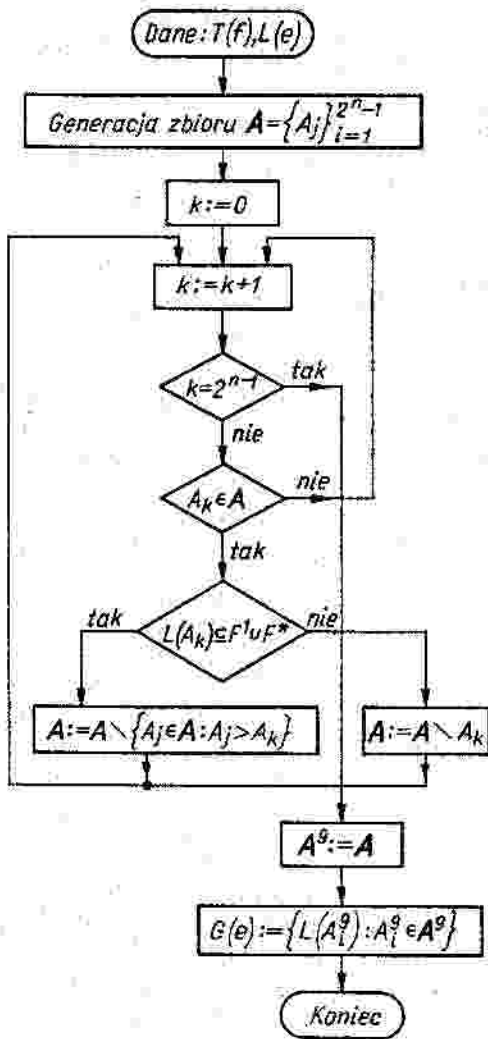
Generuje się zbiór  $A$ , następnie wybiera wektor  $A_1$  i sprawdza się, czy

$$L(A_1) \subseteq F^1 \cup F^* \quad (a)$$

Jeśli warunek (a) jest spełniony, to ze zbioru  $A$  wykreśla się wszystkie wektory  $A_j > A_1$ . Jeśli warunek ten nie jest spełniony, wykreśla się wektor  $A_1$ . Następnie wybiera się kolejny (w przypadku gdy warunek (a) był spełniony, pierwszy z nie skreślonych) wektor  $A_j$  i znowu sprawdza się czy spełniony jest warunek (a), w którym zamiast  $A_1$  podstawiono wektor  $A_j$ . Operacje te kontynuuje się dopóty, dopóki nie zostaną sprawdzone wszystkie wektory zbioru  $A$  (oprócz wykreślonych). Zbiór wektorów  $A_j$ , które pozostały w zbiorze  $A$  po zrealizowaniu powyższych operacji, oznacza się  $A^E = \{A_j^E\}$ . Ze względu na przyjętą kolejność wyboru wektorów  $A_j$  ze zbioru  $A$  dla każdego wektora  $A_j$  zachodzi relacja

$$\nexists A_j, (A_j < A_j^E) \wedge (L(A_j) \subseteq F^1 \cup F^*)$$

Z relacji tej wynika, że zespoły  $L(A_j^{\mathcal{E}})$  są zespołami maksymalnymi w obrazie  $T(f)$ . Tak więc  $G(e) = \{L(A_j^{\mathcal{E}})\}$ .



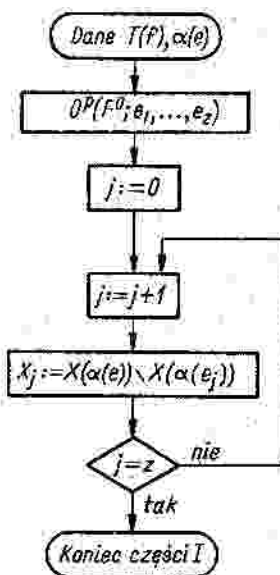
Rys. 3.3. Algorytm  $G^1$  generowania gwiazdy  $G(e)$ .

Algorytm  $G^2$ . Wykres operacyjny części I algorytmu  $G^2$  przedstawiono na rys. 3.4.

W rezultacie zrealizowania części I otrzymuje się rodzinę  $\mathbf{X} = \{X_j\}_{j=1}^2$  zbiorów  $X_j$ .



Rys. 3.4. Część I algorytmu  $G^2$  generowania gwiazdy  $G(e)$



Określmy proste implikanty  $\alpha_k$ ,  $k = 1, 2, \dots$ , funkcji  $f^S$ , której k.n.w. jest następujące:

$$f^S = \bigwedge_{j=1}^z \bigvee_{\substack{6_i \\ x_i \in X_j}} x_i \quad (3.6)$$

Zachodzi

Twierdzenie 3.3. Gwiazda  $G(e)$  jest zbiorem

$$G(e) = \{L(\alpha_k)\}_{k=1,2,\dots}$$

Dowód. Zbiory  $X_j$ ,  $j = 1, \dots, z$ , są zbiorami tych liter koniunkcji  $\alpha(e)$ , którymi różni się ona od koniunkcji  $\alpha_j$ . Jakakolwiek koniunkcja  $\alpha_k$  taka, że  $L(\alpha_k) \in G(e)$ , musi zawierać przynajmniej jedną z liter każdego zbioru  $X_j$ , aby mógł być spełniony warunek  $L(\alpha_k) \cap F^0 = \emptyset$ , równoważny warunkowi  $L(\alpha_k) \subseteq F^1 \cup F^*$ . Warunek ten spełnia każda koniunkcja d.n.w. funkcji  $f^S$ . Zespołom maksymalnym gwiazdy  $G(e)$  będą zatem odpowiadać składniki nieredukowalnego d.n.w. funkcji  $f^S$ . Wyrażenie takie otrzymuje się przez wymnożenie poszczególnych członów wyrażenia (3.6) przy



$$f^G = \bigvee_{i=1}^6 \alpha_i$$

$$\begin{aligned} \text{gdzie: } \alpha_1 &= x_4 \bar{x}_6 & \alpha_4 &= \bar{x}_5 \bar{x}_6 \\ \alpha_2 &= \bar{x}_1 x_6 & \alpha_5 &= x_3 \bar{x}_5 \\ \alpha_3 &= \bar{x}_2 x_6 & \alpha_6 &= \bar{x}_1 x_3 \bar{x}_4 \end{aligned}$$

skąd

$$G(e) = \{L_i\}_{i=1}^6, \quad \text{gdzie } L_i = L(\alpha_i).$$

### 3.4. Teoretyczne możliwości określania ekstremal za pomocą operacji lokalnych

#### 3.4.1. Określenie operacji upraszczającej nad gwiazdą $G(e)$ .

Dany jest obraz  $T(f) = F^0 \cup F^1 \cup F^*$ . Przez proces syntezy pokrycia obrazu  $T(f)$  będziemy rozumieć proces włączania do pewnego zbioru  $M_1$  kolejnych zespołów maksymalnych dopóty, dopóki otrzymany zbiór nie będzie pokryciem obrazu  $T(f)$ . Zbiór  $M_1$  traktuje się w tym procesie jako wartość pewnej zmiennej, ulegająca zmianie po dołączeniu każdego nowego zespołu maksymalnego. Jeśli pewien zespół maksymalny  $L_i$  zostaje włączony do zbioru  $M_1$ , to pola  $e_j \in L_i \cap F^1$  mogą być od tej chwili traktowane jako pola zbioru  $F^*$ , to znaczy jako pola o wartości  $*$ . Następne zespoły mogą je bowiem pokrywać, ale nie muszą.

Obecnie przejdziemy do rozważenia możliwości określenia pokrycia minimalnego  $M(T)$  przez realizację tego rodzaju procesu.

Uporządkujmy zbiór  $F^1$  za pomocą operacji  $O^D(F^1; e_1, \dots, e_w)$ . Otrzymujemy  $F^1 = \{e_j\}_{j=1}^w$ , gdzie  $w = c(F^1)$ . Określmy kolejno dla każdego pola  $e_j \in F^1$  gwiazdę  $G(e_j)$ , przy czym założmy, że zespołom  $L_i$  należącym do przecięcia gwiazd przypisuje się identyczną wartość indeksu  $i$ , natomiast zespołom różnym przypisuje się

kolejne wartości  $i = 1, 2, \dots$ . Zbiór  $F(T) = \bigcup_{j=1}^w G(e_j) = \{L_i\}_{i=1}^P$

$p = c(P(T))$ , jest zbiorem wszystkich zespołów maksymalnych w  $T(f)$ , to znaczy odpowiadającym zbiorowi  $P(f)$ . Przypiszmy każdemu polu  $e_j \in F^1$  zbiór indeksów  $I(e_j) \subseteq \{1, 2, \dots, p\}$  taki, że  $G(e_j) = \{L_i\}_{i \in I(e_j)}$ .

Jeśli kolejno z każdej gwiazdy  $G(e_j)$ ,  $j = 1, \dots, w$ , wybierać po jednym elemencie, przy czym z gwiazd przecinających się można wybrać tylko element wspólny, to uzyskany zbiór zespołów maksymalnych będzie pewnym pokryciem obrazu  $T(f)$ . Jeśli chcemy, aby tak uzyskany zbiór był pokryciem minimalnym, to wybór zespołu z kolejnej gwiazdy (wykonywany wówczas, gdy pole tej gwiazdy nie jest pokryte przez żaden zespół wybrany dotychczas) będzie w ogólnym przypadku zależał zarówno od historii poprzednich wyborów jak i od zależności między zespołami danej gwiazdy a zespołami pozostałych gwiazd (tych, z których zespoły nie były jeszcze wybrane).

Założmy, że uzyskaliśmy informację, że zespół  $L_1 \in G(e_1)$  jest ekstremalą pewnego pokrycia minimalnego. Przez  $M_1$  będziemy obecnie rozumieć zbiór, który uzupełnia się kolejno określanymi ekstremalami tego pokrycia. Tak więc  $M_1 := M_1 \cup \{L_1\}$ , gdzie  $M_1$  po prawej stronie znaku  $:=$  jest zbiorem pustym lub zbiorem już określonych ekstremal omawianego pokrycia. Pola  $e_j \in L_1 \cap F^1$  można od tej chwili traktować jako pola o wartości  $*$ , co oznacza, że można je wyłączyć ze zbioru i włączyć do zbioru  $F^*$ :

$$F^1 := F^1 \setminus L_1, \quad F^* := F^* \cup L_1.$$

Przy tym zapisie  $F^1$  i  $F^*$  traktuje się jako oznaczenia zmiennych, których wartościami są odpowiednie zbiory. Po wykonaniu powyższej operacji, przez  $T(f)$  będziemy rozumieć zbiór, w którym polami o wartości 1 i 0 są pola zbiorów aktualnych wartości zmiennych  $F^1$  i  $F^0$ . Umowę powyższą przyjmuje się we wszystkich opisanych dalej algorytmach syntezy pokryć obrazu  $T(f)$ .

Założmy obecnie, że o pewnym zespole  $L_{12} \in G(e_1)$  uzyskaliśmy informację, że nie jest ekstremalą poszukiwanego pokrycia minimalnego. Wówczas zespół  $L_{12}$  można wyłączyć z gwiazdy  $G(e_1): G(e_1) := G(e_1) \setminus \{L_{12}\}$ , a także z każdej innej gwiazdy, do której zespół

ten należy. Operacje te realizuje się przez wykreślenie indeksu  $i_2$  ze wszystkich zbiorów  $I(e_j)$  zawierających ten indeks.

Działanie polegające na wyłączeniu niektórych pól ze zbioru  $F^1$  i włączeniu ich do zbioru  $F^*$  lub wyłączeniu pewnych zespołów maksymalnych z gwiazd  $G(e_j)$ ,  $e_j \in F^1$ , nazywa się działaniami upraszczającymi obraz  $T(f)$ .

Założmy, że w danej chwili opisywanego procesu rozważa się gwiazdę  $G(e_k)$ ,  $e_k \in F^1$ .

Niech  $O_i = O_i(G(e_k), T(f), \Lambda)$ ,  $i \in \{1, 2, \dots\}$ , oznacza operację polegającą na badaniu zależności między zespołami maksymalnymi pewnego zbioru  $\Lambda$  i wykonaniu w wyniku tego badania jednego z czterech typów działań:

I. Włączenie pewnego zespołu  $L_i^e \in G(e_k)$  do zbioru  $M^1$ :  $M_1 := M_1 \cup \{L_i^e\}$  i wykonanie działań:

$$F^1 := F^1 \setminus L_i^e, \quad F^* := F^* \cup L_i^e$$

II. Wyłączenie pewnych zespołów  $L_i$ ,  $i \in I$ , z gwiazdy  $G(e_k)$

$$G(e_k) := G(e_k) \setminus \{L_i\}_{i \in I}$$

oraz pozostałych gwiazd zawierających ten zespół.

III. Wyłączenie pewnego zbioru pól  $E \in G^u(e_k)$  ze zbioru  $F^1$  i włączenie do zbioru  $F^*$ :

$$F^1 := F^1 \setminus E, \quad F^* := F^* \cup E$$

IV. Niewykonanie żadnego działania.

Operację  $O_i$  nazywa się operacją upraszczającą nad gwiazdą  $G(e_k)$ . Operację  $O_i$  nad gwiazdą  $G(e_k)$  nazywa się operacją o zasięgu 1, jeśli

$$\Lambda = \Lambda^1 = G(e_k) \quad (3.7)$$

Operację  $O_i$  nad gwiazdą  $G(e_k)$  nazywa się operacją o zasięgu 1,  $l = 2, 3, \dots$ , jeśli:

$$\Lambda = \Lambda^l = \bigcup_{e_j \in \mathcal{E}_1} G(e_j) \quad (3.8)$$

gdzie  $G_1^U$  jest zbiorem wszystkich pól, których gwiazdy mają nie-puste przecięcie z gwiazdą przynajmniej jednego pola  $e_j \in G_{1-1}^U$ , przy czym  $G_1^U = \{e_k\}$ .

Pojęcie zasięgu operacji jest więc związane ze zbiorem pól, dla których należy wygenerować gwiazdy biorące udział przy wykonywaniu danej operacji.

Operacje upraszczające o zasięgu  $1 \leq 2$  nazywa się operacjami lokalnymi.

Jeśli pewne operacje  $O_i$ ,  $i = 1, 2, \dots$ , nad gwiazdą każdego pola  $e_j \in P^i$  sprowadzają się do wykonania działania IV, to obraz  $T(f)$  nazywa się obrazem nieupraszczalnym za pomocą operacji  $O_i$ .

Algorytmy syntezy pokrycia obrazu  $T(f)$ , polegające na wykonywaniu operacji upraszczających o zasięgu nie większym niż 1, nazywa się algorytmami stopnia 1. Algorytmy, w których wykonuje się operacje lokalne, nazywa się algorytmami lokalnymi.

Okazuje się, że za pomocą odpowiednio skonstruowanych algorytmów lokalnych można dla szeregu funkcji logicznych uzyskać pokrycie minimalne ich obrazu lub przynajmniej jego niepusty podzbiór.

Istotną cechą przedstawionego tutaj podejścia jest to, że określenie ekstremal i uproszczenia obrazu  $T(f)$  odbywa się na podstawie pewnego ograniczonego podzbioru zespołów maksymalnych  $\Lambda \subset P(T)$ . Podzbiór ten można generować doraźnie przed wykonaniem danej operacji upraszczającej nad określoną gwiazdą. Synteza pokrycia minimalnego (lub - ogólnie - nieredukowalnego) nie wymaga więc przy tej metodzie uprzedniego określenia wszystkich zespołów maksymalnych w obrazie  $T(f)$  (oznacza to więc połączenie etapu I z etapem II w procesie syntezy wyrażenia minimalnego).

Podejście to umożliwia uniknięcie konieczności pamiętania (co oznacza redukcję potrzebnej objętości pamięci przy realizacji maszynowej) tych zespołów maksymalnych, o których - w wyniku wykonania operacji upraszczającej - uzyskuje się informację, że nie mogą być ekstremalami, a także konieczności generowania szeregu zespołów maksymalnych, ponieważ potrzeba ich generowania zostanie z góry wykluczona wskutek wykonania operacji upraszczających.

### 3.4.2. Twierdzenia o możliwościach lokalnego wydobywania ekstremal

Obecnie sformułujemy szereg twierdzeń, na podstawie których

skonstruujemy następnie pewne lokalne operacje upraszczające.

Twierdzenie 3.4. Jeśli gwiazda  $G(e)$  pewnego pola  $e \in F^1$  składa się tylko z jednego zespołu maksymalnego, to zespół ten jest ekstremalną jądrową.

Dowód. Wynika bezpośrednio z definicji 3.6.

Q.E.D.

Niech  $L^p = \{L_i^p\}$  oznacza jądro pokrycia obrazu  $T(f)$ .

Twierdzenie 3.5. Zespoły maksymalne  $L_j$  w obrazie  $T(f)$  spełniające warunek:

$$L_j \setminus \left\{ i: L_i^p \in L^p \right\} L_i^p \in F^* \quad (a)$$

nie wchodzą do żadnego pokrycia nieredukowalnego tego obrazu.

Dowód. Zgodnie z wnioskiem 3.2, jądro  $L^p$  zawiera się w każdym pokryciu nieredukowalnym obrazu  $T(f)$ . Ponieważ zespoły  $L_j$  spełniające warunek (a) można wykluczyć z każdego pokrycia zawierającego jądro bez naruszenia warunku pokrycia (3.1), więc ich istnienie w jakimkolwiek pokryciu nieredukowalnym przeczyłoby definicji 3.5.

Q.E.D.

Warunek (a) w twierdzeniu 3.5 jest warunkiem dostatecznym, aby dany zespół maksymalny nie wchodził do żadnego pokrycia nieredukowalnego, ale nie jest warunkiem koniecznym. Ten ostatni sformułowano w twierdzeniu 3.8.

Definicja 3.12. Pole  $e_r \in F^1$  nazywa się polem redukowalnym w obrazie  $T(f)$ , jeśli istnieje takie pole  $e_j \in F^1$ , że  $G(e_j) \subseteq G(e_r)$ , przy czym w przypadku, gdy  $G(e_j) = G(e_r)$ , wówczas  $\gamma(e_j) < \gamma(e_r)$ .

Twierdzenie 3.6. Zbiór wszystkich pokryć nieredukowalnych  $N(T)$  obrazu  $T(f)$  jest równy zbiorowi wszystkich pokryć nieredukowalnych  $N(\tilde{T})$  obrazu  $\tilde{T}(f) = \tilde{F}^1 \cup \tilde{F}^* \cup F^0$ , gdzie  $\tilde{F}^1 = F^1 \setminus E_r$ ,  $\tilde{F}^* = F^* \cup E_r$ , natomiast  $E_r$  - zbiór wszystkich pól redukowalnych w obrazie  $T(f)$ .

Dowód. Zgodnie z definicją 3.12, dla każdego  $e_r \in E_r$  istnieje  $e_k \in \tilde{F}^1$  takie, że  $G(e_k) \subseteq G(e_r)$ ,  $e_r \in F^1$ . Dlatego jakiekolwiek pokrycie nieredukowalne obrazu  $\tilde{T}(f)$  jest jednocześnie pokryciem nieredukowalnym obrazu  $T(f)$ , a więc  $N(\tilde{T}) \subseteq N(T)$ .

Założmy teraz, że  $N(T) \setminus N(\tilde{T}) \neq \emptyset$ . Istnieje więc pokrycie nieredukowalne  $N_i(T)$ , które nie jest pokryciem obrazu  $\tilde{T}(f)$ . Oznacza to, że  $N_i(T)$  zawiera zespoły maksymalne pokrywające wyłącza-

nie pola zbioru  $E_T \cup F^*$ . Zespoły te można jednak usunąć z  $N_1(T)$  bez naruszenia warunku pokrycia, ponieważ pola  $E_T$ , jak wykazano wyżej, będą pokryte przez pewne inne zespoły maksymalne pokrycia  $N_1(T)$ . Wynika z tego, że  $N_1(T)$  nie jest pokryciem nieredukowalnym, co jest sprzeczne z poprzednim założeniem. Wynika z tego, że  $N(\tilde{T}) = N(T)$ . Q.E.D.

Dane jest pewne pole  $e_k \in F^1 \in T(f)$ .

Twierdzenie 3.7. Jeśli

$$\nexists e_j \in F^1, G(e_j) \subset G(e_k) \quad (a)$$

to

$$\forall L_i \in G(e_k) \exists N_1(T), L_i \in N_1(T)$$

Dowód. Niech  $L_k$  oznacza pewien zespół maksymalny gwiazdy  $G(e_k)$ . Wyłączmy pola zbioru  $L_k \cap F^1$  ze zbioru  $F^1$  i włączmy do zbioru  $F^*$ :  $F^1 := F^1 \setminus L_k$ ,  $F^*_k := F^* \cup L_k$ . Otrzymany obraz  $T(f)$  oznaczmy  $\tilde{T}(f)$ . Niech  $N_1(\tilde{T})$ , to jest pewne nieredukowalne pokrycie obrazu  $\tilde{T}(f)$ , spełnia warunek

$$N_1(\tilde{T}) \cap G(e_k) = \emptyset \quad (b)$$

Istnienie takiego pokrycia jest zagwarantowane warunkiem (a). Z warunku (b) wynika, że pokrycie  $N_1(\tilde{T})$  nie zawiera zespołu maksymalnego pokrywającego pole  $e_k$ . W takim razie zbiór  $N_1(\tilde{T}) \cup L_k$  jest pokryciem nieredukowalnym obrazu  $T(f)$ . Analogiczne rozumowanie można przeprowadzić względem każdego  $L_i \in G(e_k)$ . Q.E.D.

Twierdzenie 3.8. Warunkiem koniecznym i dostatecznym, aby zespół maksymalny  $L_i$  nie wchodził do żadnego pokrycia nieredukowalnego obrazu  $T(f)$ , jest warunek

$$L_i \in E_T \cup F^* \quad (a)$$

gdzie  $E_T$  jest zbiorem wszystkich pól redukowalnych obrazu  $T(f)$ .

Dowód. **D o s t a t e c z n o ść :** Wynika bezpośrednio z twierdzenia 3.6. **K o n i e c z n o ść :** Założmy, że warunek (a) nie jest spełniony. Wówczas, zgodnie z definicją 3.12, istnieje pole  $e_k \in L_i$



takie, że

$$\bigoplus (e_j) \in P^1, \quad G(e_j) \subset G(e_k)$$

Z twierdzenia 3.7 wynika, że każdy zespół gwiazdy  $G(e_k)$ , a więc i zespół  $L_1$ , wchodzi do pewnego pokrycia nieredukowalnego  $N_1(T)$ . W takim razie, aby zespół nie wchodził do żadnego pokrycia nieredukowalnego, konieczne jest spełnienie warunku (a). Q.E.D.

Niech  $G^S(e) = \{L_i^S\}_{i=1}^a$  oznacza pewien podzbiór gwiazdy  $G(e)$ ,  $e \in P^1$ , w obrazie  $T(f)$  i niech  $G^W(e) = \{L_i^W\}_{i=1}^b = G(e) \setminus G^S(e)$ . Oznaczmy symbolem  $r_i^S$ ,  $i = 1, \dots, a$ , rozmiary zespołów  $L_i^S$ , symbolem zaś  $r_i^W$ ,  $i = 1, \dots, b$ , rozmiary zespołów  $L_i^W$ .

Twierdzenie 3.9. Jeśli

$$\forall L_i^S \in G^S(e) \exists L_i^W \in G^W(e), \quad \left[ (L_i^S \setminus L_i^W \subseteq P^*) \wedge (r_i^S \leq r_i^W) \right] \quad (a)$$

to istnieje pokrycie minimalne  $M(T)$  nie zawierające żadnego zespołu  $L_i^S$ ,  $i = 1, \dots, a$ .

Dowód. Niech  $L_k^S$  oznacza pewien zespół podzbioru  $G^S(e)$ . Jeśli zachodzi (a), to istnieje  $L_k^W \in G^W(e)$  taki, że  $L_k^S \setminus L_k^W \subseteq P^*$  (b) oraz  $r_k^S \leq r_k^W$  (c). Z warunku (b) wynika, że  $L_k^S \cap P^1 \subseteq L_k^W \cap P^1$ . W takim razie w każdym pokryciu  $N_1$  zawierającym  $L_k^S$  można zespół  $L_k^S$  zastąpić zespołem  $L_k^W$  bez zmiany długości tego pokrycia. Zmiana ta nie spowoduje również zwiększenia złożoności pokrycia, ponieważ ze względu na (c) zachodzi nierówność  $n - r_k^W \leq n - r_k^S$ . Wynika z tego, że jeśli pewne pokrycie  $N_1$ , zawierające  $L_k^S$ , byłoby pokryciem minimalnym, to również pokrycie utworzone przez zmianę zespołu  $L_k^S$  na zespół  $L_k^W$  będzie pokryciem minimalnym. Ponieważ rozumowanie powyższe nie zależy od tego, który zespół  $L_i^S$  przyjęto jako zespół  $L_k^S$ , więc to kończy dowód. Q.E.D.

W dalszym ciągu  $G^S(e) \subseteq G(e)$  oznacza podzbiór o maksymalnej mocy, dla którego jest spełniony warunek (a) twierdzenia 3.9 (jeśli istnieje kilka takich podzbiórów, to  $G^S(e)$  oznacza dowolny z nich).

Ze względu na występujący w warunku (a) tego twierdzenia zbiór  $P^*$ , podzbiór  $G^S(e)$  zależy od aktualnego obrazu  $T(f)$ .

Definicja 3.13. Zbiór  $G^W(e) = G(e) \setminus G^S(e)$  nazywa się gwiazdą

zredukowaną pola  $e$  w obrazie  $T(f)$ .

Niech  $L, L^1, L^2, \dots, L^a, a > 1$ , oznaczają pewne zespoły maksymalne w obrazie  $T(f)$ , posiadające rozmiary  $r, r^1, \dots, r^a$  i spełniające warunek  $E_t = L \cap \bigcup_{i=1}^a L^i \cap F^1 \neq \emptyset$ . Niech  $M_U^k$  i  $M_\Omega^k$  oznaczają odpowiednio sumę mnogościową i przecięcie wszystkich pokryć  $M_i^k(T)$  obrazu  $T(f)$ . Niech  $M_U^1$  i  $M_\Omega^1$  oznaczają odpowiednio sumę mnogościową i przecięcie wszystkich pokryć  $M_i^1(T)$ .

Twierdzenie 3.10. Jeśli:

$$I. L \setminus E_t \subseteq F^* \quad (a)$$

$$\forall (L^i \in \{L^i\}_{i=1}^a), (L^i \setminus E_t \subseteq F^*) \wedge (L^i \cap E_t \neq \emptyset) \quad (b)$$

$$\exists e \in E_t, G(e) \subseteq \{L, L^1, L^2, \dots, L^a\} \quad (c)$$

to  $L \in M_\Omega^k, L^1, L^2, \dots, L^a \notin M_U^k$ ;

II. są spełnione warunki (a) - (c) oraz dodatkowo warunek

$$\sum_{i=1}^a (n - r^i) < n - r \quad (d)$$

to  $L \notin M^1, L^1, L^2, \dots, L^a \in M_\Omega^1$ .

Dowód. Z (a) wynika, że

$$L \cap F^1 = E_t. \quad (e)$$

Z warunku (b) wynika, że

$$\left( \bigcup_{i=1}^a L^i \right) \cap F^1 = E_t \quad (f)$$

oraz że

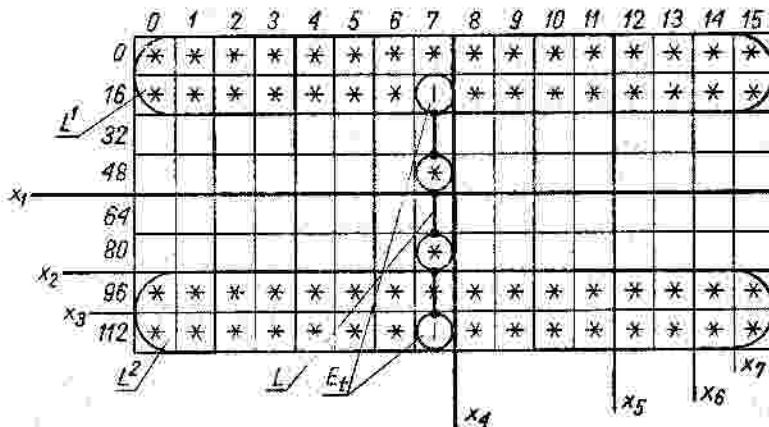
$$\nexists L' \in \{L^i\}_{i=1}^a, E_t \subseteq L' \quad (g)$$

Z (c), (e) i (f) wynika, że  $E_t$  może być pokryty albo przez zespół  $L$ , albo przez zespoły  $\{L^i\}_{i=1}^a$ . Ponieważ  $a > 1$ , więc zespół  $L$  wejdzie do każdego pokrycia minimalnego najkrótszego

obrazu  $T(f)$ , to znaczy  $L \in M_n^k$ , natomiast zespoły  $L^i$ ,  $i = 1, \dots, a$ , nie mogą wejść do żadnego takiego pokrycia, to znaczy  $L \notin M_n^k$ . To dowodzi część I twierdzenia.

Złożoność  $z(\{L^i\}_{i=1}^a) = \sum_{i=1}^a (n - r^i)$ , natomiast  $z(L) = n - r$ .

Z warunku (d) wynika, że  $z(\{L^i\}_{i=1}^a) < z(L)$ . W takim razie w każdym pokryciu minimalnym  $M^1(T)$  będzie zawarty zbiór  $\{L^i\}_{i=1}^a$ , natomiast w żadnym z nich nie może być zawarty zespół  $L$ . Q.E.D.



$$\begin{aligned}
 M^k(T) &= \{L\}, & z(M^k(T)) &= 5, & c(M^k(T)) &= 1 \\
 M^1(T) &= \{L_1, L_2\}, & z(M^1(T)) &= 4, & c(M^1(T)) &= 2 \\
 r &= 2 \\
 r^1 &= 5 \\
 r^2 &= 5
 \end{aligned}$$

Rys. 3.6. Ilustracja twierdzenia 3.10

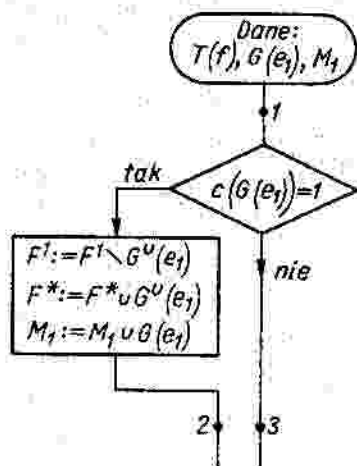
Twierdzenie 3.10 można łatwo uogólnić na przypadek, gdy zamiast jednego zespołu  $L$  występuje zbiór  $\{L_1, L_2, \dots, L_b\}$ ,  $b < a$ . W

tym przypadku  $E_t = \bigcup_{i=1}^a L^i \cap \bigcup_{i=1}^b L_i^1 \cap F^1$ .

### 3.4.3. Operacje lokalne $O_1, O_2, O_3$ .

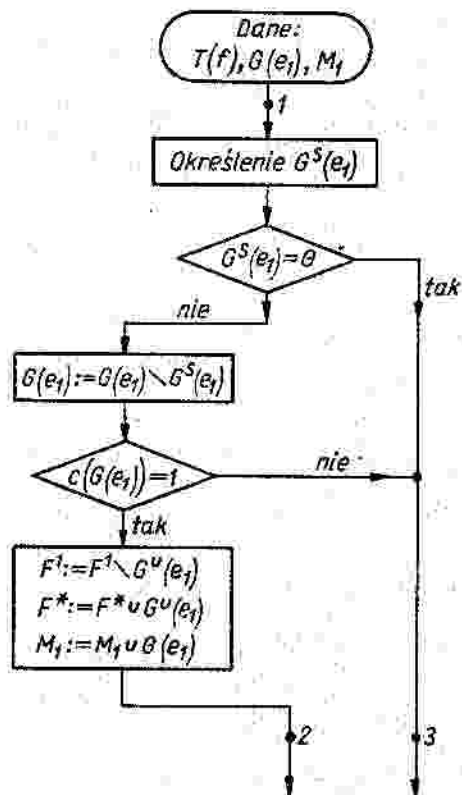
Na podstawie sformułowanych w punkcie 3.4.2 twierdzeń skon-

struujemy obecnie pewne lokalne operacje uproszczające  $O_1, O_2, O_3$ .



Rys. 3.7. Wykres operacyjny operacji  $O_1$ .

Rys. 3.8. Wykres operacyjny operacji  $O_2$ .



Operacje  $O_1$  i  $O_2$  są operacjami o zasięgu 1 i polegają na wykonaniu działań typu I, II lub IV w zależności od spełnienia warunków twierdzeń 3.4 i 3.9. Operacja  $O_3$  jest operacją o zasięgu 2 i polega na wykonaniu działań typu III lub IV zgodnie z twierdzeniem 3.6, w zależności od spełnienia warunków wynikających z definicji 3.12.

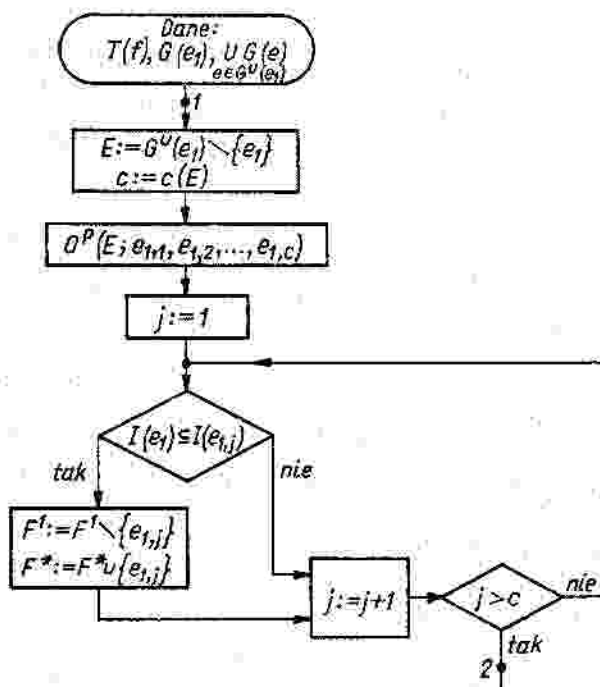
1. Operacja  $O_1 = O_1(G(e), T(f), G(e_1))$  (rys. 3.7).

W operacji  $O_1$  wykorzystuje się twierdzenie 3.4. Zespół  $L_1$  jest, zgodnie z tym twierdzeniem, ekstremalą jądrową.

2. Operacja  $O_2 = O_2(G(e_1), T(f), G(e_1))$  (rys. 3.8).

W operacji  $O_2$  wykorzystuje się twierdzenia 3.4 oraz 3.9.

3. Operacja  $O_3 = O_3(G(e_1), T(f), \bigcup_{e \in G^U(e_1)} G(e))$  (rys. 3.9).



Rys. 3.9. Wykres operacyjny operacji  $O_3$ .

W operacji  $O_3$  wykorzystuje się twierdzenie 3.6. Ponieważ każdemu polu  $e_j \in \mathbb{R}^1$  został przypisany podzbiór indeksów  $I(e_j) =$

$= \{i : I_i \in G(e_j)\}$ , więc zamiast sprawdzać relację  $G(e_i) \subseteq G(e_{r,j})$  sprawdza się tu równoważną jej relację  $I(e_i) \subseteq I(e_{r,j})$ . Zmienna  $E$  pełni funkcję pomocniczą.

Oprócz powyższych operacji, można skonstruować jeszcze inną operację lokalną o zasięgu 2, której podstawą jest twierdzenie 3.10. Sprawdzanie warunków tego twierdzenia jest jednak dość kłopotliwe, a przy tym warunki te są stosunkowo rzadko spełnione w praktyce, dlatego operacji tej nie wykorzystuje się w dalszej opisanych algorytmach. W pktcie 5.2 sformułowano twierdzenie, które umożliwia skonstruowanie operacji lokalnej (o zasięgu 1), upraszczającej obraz  $T(f)$  w pewnym przypadku symetrii funkcji względem liter.

#### 3.4.4. Ograniczoność metod, których podstawą jest lokalne wydobywanie ekstremal

Wykonywanie operacji  $O_1, O_2, O_3$  nad gwiazdami poszczególnych pól  $e_j \in F^1$  prowadzi do uproszczenia obrazu  $T(f)$ , jeśli są spełnione warunki odpowiednich twierdzeń, będących podstawą tych operacji. Jeśli obraz  $T(f)$  nie zawiera pól redukowalnych, a gwiazda zredukowana każdego pola  $e_j \in F^1$  składa się z więcej niż jednego zespołu maksymalnego, to wówczas obraz  $T(f)$  jest nieupraszczalny za pomocą operacji  $O_1, O_2, O_3$ . Nie jest więc upraszczalny żadnym algorytmem lokalnym zbudowanym na podstawie tych operacji. Tego rodzaju obraz nazywa się obrazem cyklicznym.

Jeśli dany jest pewien obraz  $T(f)$ , to może być on od razu obrazem cyklicznym lub może stać się nim po wykonaniu operacji upraszczających  $O_1, O_2, O_3$  nad gwiazdami poszczególnych pól  $e_j \in F^1$ . Jak wynika z twierdzenia 3.7, każdy zespół maksymalny w obrazie cyklicznym wchodzi do pewnego pokrycia nieredukowalnego.

W szczególnych przypadkach obraz cykliczny może być uproszczony za pomocą operacji, których podstawą jest na przykład twierdzenie 3.10. W ogólnym jednak przypadku informacja, jaką dostarczają zbiory  $\Lambda^1$  lub  $\Lambda^2$ , nie wystarcza do stwierdzenia, który z zespołów rozważanej gwiazdy jest ekstremalą. Powstaje pytanie, czy można skonstruować takie operacje o większym zasięgu, które w każdym przypadku doprowadzą do określenia ekstremali w danej gwiazdzie. Odpowiedź na to pytanie jest negatywna.

Można udowodnić w taki sposób, jak to uczynił Żurawlew [28] (a

mianowicie na podstawie takiej samej funkcji, jaką w tej pracy skonstruowałam), że dla każdej operacji skończonego zasięgu można znaleźć funkcję, której obraz jest nieupraszczalny za pomocą tej operacji. Wynika z tego, że nie istnieje algorytm skończonego stopnia, dokonujący syntezy pokrycia minimalnego obrazu każdej funkcji logicznej.

Algorytmy wyższych stopni mogą określić pokrycia minimalne tylko dla pewnych konkretnych obrazów, dla których nie udało się tego zrobić algorytmami niższego stopnia. W ogólnym natomiast przypadku określenie pokrycia minimalnego będzie wymagało realizacji algorytmu, w którym operacje upraszczające mają zasięg obejmujący cały zbiór  $P(T)$ . Oznacza to - innymi słowy, że uzyskanie informacji, czy dany zespół maksymalny jest ekstremalą czy też nie wymaga określenia pokryć nieredukowalnych i porównania ich ze sobą.

Sformułowane w punkcie 4.4.3 operacje lokalne  $O_1, O_2, O_3$  są jednak wygodne do skonstruowania pierwszej części ogólnego algorytmu syntezy pokrycia minimalnego, umożliwiającą redukowanie danego obrazu do obrazu cyklicznego. W szczególnych przypadkach ta pierwsza część algorytmu wystarcza do określenia pokrycia minimalnego.

### 3.5. Synteza minimalnych pokryć obrazów cyklicznych metodą rozłącznych gwiazd

#### 3.5.1. Uwagi wstępne

Założmy, że  $T(f)$  jest obrazem cyklicznym. Określmy dla każdego pola  $e_j \in F^1$  gwiazdę  $G(e_j)$ , przy czym przypiszmy, jak poprzednio, różnym zespołom maksymalnym  $L_i$  różne indeksy  $i = 1, 2, \dots, p$ .

Każdemu polu  $e_j$  przyporządkujemy <sup>1)</sup> zbiór indeksów  $I(e_j) = \{i : L_i \in G(e_j)\}$ . Pokrycia nieredukowalne obrazu  $T(f)$  można określić na podstawie następującej funkcji:

$$p(L_1, \dots, L_p) = \bigwedge_{j=1}^m \bigvee_{i \in I(e_j)} L_i \quad (3.9)$$

<sup>1)</sup> Przy ręcznej realizacji algorytmu zbiory  $I(e_j)$  wpisuje się w poszczególne pola  $e_j$  (zob. /87/).

gdzie  $e_1, e_2, \dots, e_m \in F^1$ ,  $m = c(F^1)$ .

Zespoły maksymalne  $L_i$  traktuje się tu jako zmienne logiczne. Funkcja  $p(L_1, \dots, L_p)$  jest analogiczna do funkcji skonstruowanej przez Petrica [7] (będącej funkcją prostych implikantów funkcji  $f$ ). Jeśli wymnożymy poszczególne czony wyrażenia (3.9) przez siebie, przy jednoczesnym stosowaniu praw pochłaniania, to otrzymamy nieredukowalne d.n.w. funkcji  $p(L_1, \dots, L_p)$ , które (ze względu na monotoniczność tej funkcji) jest jednocześnie jej wyrażeniem minimalnym.

Każdemu składnikowi otrzymanego wyrażenia odpowiada zbiór zespołów maksymalnych  $L_i$ , który stanowi pewne nieredukowalne pokrycie obrazu  $T(f)$ . Nieredukowalność tego pokrycia wynika z tego, że składniki omawianego d.n.w. funkcji  $p$  są jej prostymi implikantami, a wśród zespołów  $L_i$  w wyrażeniu (3.9) brak jest, ze względu na cykliczność obrazu  $T(f)$ , zespołów, które nie wchodzą przynajmniej do jednego pokrycia nieredukowalnego tego obrazu. Pokrycie minimalne określa się drogą przeglądu tak otrzymanych pokryć nieredukowalnych.

Realizacja powyższego algorytmu, już nawet w stosunkowo prostych przypadkach, wymaga wykonania bardzo dużej ilości operacji. W punktach 3.5.2. - 3.5.4. opisano pewną metodę, w której wymusza się tylko pewną część członów k.n.w. funkcji  $p(L_1, \dots, L_p)$  i na podstawie składników otrzymanego wyrażenia określa pokrycie minimalne.

### 3.5.2. Konstruowanie rodziny $G^r$ rozłącznych gwiazd

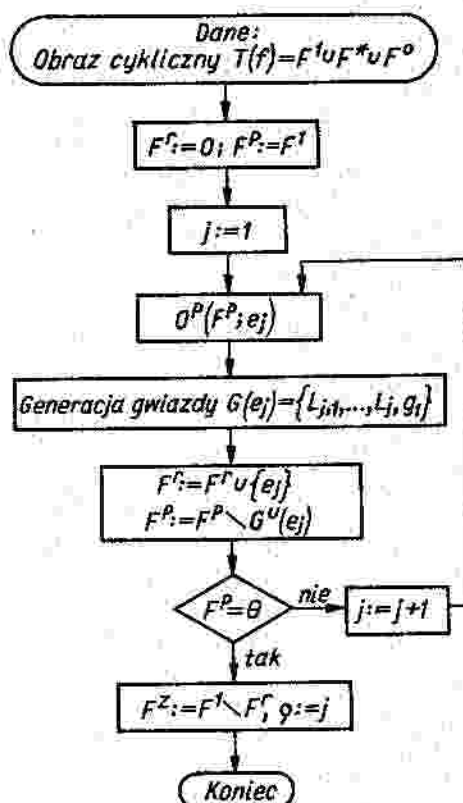
Wykonajmy algorytm  $R$  przedstawiony na rys. 3.10.

Przez  $F^r$ ,  $F^D$ ,  $F^Z$  rozumie się w tym algorytmie pewne zmienne, których wartościami są zbiory. Obecnie przez  $F^r$  i  $F^Z$  będziemy rozumieć zbiory, które są wartościami tych zmiennych uzyskanymi po zakończeniu algorytmu  $R$  (wartość  $F^D$  jest zbiorem pustym). Generując gwiazdę  $G(e_j)$  pola  $e_j$ , zespoły tej gwiazdy oznacza się w algorytmie symbolami  $L_{j_1}, L_{j_2}, \dots, L_{j_g}$ .

Rodziny gwiazd  $G(e_j)$ , które zostały określone w algorytmie  $R$ , oznacza się

$$G^r = \{G(e_j)\}_{j=1}^F$$





Rys. 3.10. Algorytm R dla konstrukcji rodziny rozłącznych gwiazd  $G^r$

Ze sposobu wyboru pól, dla których generuje się gwiazdy  $G(e_j)$ , wynika, że dla dowolnej pary indeksów  $(j_1, j_2)$ ,  $j_1, j_2 \in \{1, \dots, \xi\}$ ,  $j_1 \neq j_2$ , zachodzi relacja:

$$G(e_{j_1}) \cap G(e_{j_2}) = \emptyset$$

która oznacza, że nie istnieje para zespołów  $(L_{j_1, k_1}, L_{j_2, k_2}) \in G(e_{j_1}) \times G(e_{j_2})$  taka, że  $L_{j_1, k_1} = L_{j_2, k_2}$ . Rodzina  $G^r$  jest zatem rodziną gwiazd wzajemnie rozłączonych. Jest to rodzina maksymalna względem inkluzji, ponieważ algorytm R kontynuuje się do chwili, gdy  $F^p = \emptyset$ , co oznacza, że nie istnieje już pole

$e \in F^1 \setminus F^2 = F^2$ , którego gwiazda  $G(e)$  nie przecinałaby się z którąś z gwiazd  $G^x$ . Rodzinę  $G^x$  nazywa się dalej rodziną gwiazd rozłącznych.

Twierdzenie 3.11. Dla każdego pokrycia nieredukowalnego  $N_1(T)$  zachodzi relacja  $c(N_1(T)) \geq \rho$ .

Dowód. Zbiór  $F^x = \{e_j\}_{j=1}^{\rho}$  zawiera się w zbiorze  $F^1$ , a więc jakiegokolwiek pokrycie nieredukowalne  $N_1(T)$  musi - zgodnie z definicją 3.1 - zawierać zespoły maksymalne pokrywające pola  $e_j \in F^x$ . Ze względu na rozłączność gwiazd rodziny  $G^x$ , aby pokryć pola  $e_j \in F^x$ , należy z każdej gwiazdy rodziny  $G^x$  wybrać przynajmniej po jednym zespole. Z tego wynika, że zbiór zespołów maksymalnych pokrywających pola zbioru  $F^x$  musi zawierać co najmniej  $c(G^x) = \rho$  zespołów.

Q.E.D.

W przypadku przyjęcia innego niż w algorytmie R porządku wyliczenia pól, dla których generuje się gwiazdy rozłączne, może okazać się, że moc  $\rho_1$  uzyskanej w ten sposób rodziny gwiazd rozłącznych jest większa niż moc  $\rho$ . Wówczas twierdzenie 3.11 obowiązywałoby dla wartości  $\rho_1$ .

### 3.5.3. Opis teoretyczny metody

Wykonajmy operację  $OP(F^Z; e_{\rho+1}, e_{\rho+2}, \dots, e_m)$ , gdzie  $m = c(F^Z)$ . Dla każdego pola  $e_j \in F^Z$  można znaleźć co najmniej jeden zespół  $L_{j,k}$  należący do którejś z gwiazd rozłącznych  $G(e_j)$ ,  $j \in \{1, \dots, \rho\}$ , który pokrywa to pole. Zbiory  $G(e_j)$  zespołów maksymalnych pokrywających pola  $e_j \in F^Z$ , uzupełnimy obecnie w nowe zespoły, tak aby otrzymać gwiazdy  $G(e_j)$ ,  $j = \rho+1, \rho+2, \dots, m$ . Przy określaniu gwiazdy  $G(e_a)$ ,  $a \in \{\rho+1, \rho+2, \dots, m\}$  zespoły uzupełniające oznaczmy  $L_{a,1}, \dots, L_{a,g_a}$ . Zespoły te będą pokry-

wać jednocześnie pewne inne pola  $e_j \in F^Z$ , toteż przy określaniu gwiazdy następných pól  $e_j$ ,  $j > a$ , uzupełniającymi będą tylko nowe zespoły, które poprzednio nie były jeszcze określone. Rodzinę gwiazd  $G(e_j)$ ,  $j = \rho+1, \dots, m$ , oznacza się symbolem  $G^z$  i

nazywa rodziną gwiazd zależnych. Zbiór  $\bigcup_{j=1}^m G(e_j)$  jest zbiorem wszystkich zespołów maksymalnych w obrazie  $T(f)$ , to znaczy zbiorem  $P(T)$ . Przypiszmy każdemu polu  $e_j$  zbiór  $I(e_j)$  par indek-

sów  $(\underline{j}, \underline{k})$  taki, że  $G(e_j) = \{L_{\underline{j}, \underline{k}}\}, (\underline{j}, \underline{k}) \in I(e_j)$ . Funkcje  $p$  można obecnie zapisać następująco:

$$p = p^r \wedge p^z \quad (3.10)$$

gdzie

$$p^r = \bigwedge_{j=1}^{\rho} \bigvee_{k=1}^{\rho_j} L_{\underline{j}, \underline{k}}, \quad p^z = \bigwedge_{j=\rho+1}^m \bigvee_{(\underline{j}, \underline{k}) \in I(e_j)} L_{\underline{j}, \underline{k}}$$

Zespoły maksymalne  $L_{\underline{j}, \underline{k}}$  traktuje się tu jako zmienne logiczne.

Wymnożmy poszczególne czony wyrażenia  $p^z$  przez siebie, stosując przy tym prawa pochłaniania. Otrzymamy nieredukowalne d.n.w. funkcji  $p^z$  o postaci:

$$p^z = \beta_1 \vee \beta_2 \vee \dots \vee \beta_b \quad (3.11)$$

gdzie  $\beta_i, i = 1, 2, \dots, b$ , są pewnymi koniunkcjami zmiennych  $L_{\underline{j}, \underline{k}}$ .

Zbiory zmiennych  $L_{\underline{j}, \underline{k}}$  wchodzących do poszczególnych składników  $\beta_i$  oznaczymy symbolem  $\tilde{\beta}_i$  i traktujemy jako zbiory zespołów maksymalnych. Ze sposobu utworzenia zbiorów  $\tilde{\beta}_i$  wynika, że zespoły maksymalne każdego takiego zbioru pokrywają wszystkie pola zbioru  $F^z$  oraz pewne pola zbioru  $F^r$ . Określmy dla każdego zbioru  $\tilde{\beta}_i$  parametr  $\Delta_i$ :

$$\Delta_i = c(\tilde{\beta}_i) - d_i$$

gdzie  $d_i$  - maksymalna liczba zespołów w zbiorze  $\tilde{\beta}_i$ , które posiadają różną wartość indeksu  $j$  i przy tym nie większą niż  $\rho$ .

Parametr  $\Delta_i = 0$ , jeśli każdy z zespołów maksymalnych zbioru  $\tilde{\beta}_i$  posiada różną wartość indeksu  $j$  i nie większą niż  $\rho$ .

**Twierdzenie 3.12.** Dla każdego zbioru  $\tilde{\beta}_i$  można znaleźć pokrycie nieredukowalne  $N(T) \supseteq \tilde{\beta}_i$ , posiadające moc równą  $\rho + \Delta_i$ .

**Dowód.** Twierdzenia dowodzi się przez konstrukcję pokrycia  $N(T)$ . Wybierzmy dowolny zbiór  $\tilde{\beta}_w, w \in \{1, 2, \dots, b\}$ . Określmy w zbiorze  $\tilde{\beta}_w$  dowolny, maksymalny względem inkluzji, podzbiór  $\tilde{\beta}_w^r$  zespołów maksymalnych posiadających różne i nie większe niż  $\rho$  war-

tości indeksu  $j$ . Każdy zespół tego podzbioru jest elementem pewnej gwiazdy  $G(e_j)$ ,  $e_j \in F^x$ . Oznaczmy symbolem  $E$  zbiór pól  $e_j \in F^x$ , które są pokryte przez zespoły zbioru  $\tilde{\beta}_w^r$ . Zespoły podzbioru  $\tilde{\beta}_w^r$  pokrywają więc pola zbioru  $F^x \cup E$ . Wybierzmy z każdej gwiazdy  $G(e_j)$ ,  $e_j \in F^x \setminus E$ , dowolny zespół maksymalny i oznaczmy otrzymany zbiór symbolem  $B$ . Zbiór  $\tilde{\beta}_w \cup B$  jest pokryciem obrazu  $T(f)$  o mocy  $(\Delta_1 + d_1) + (\varphi - d_1) = \varphi + \Delta_1$ . Ze względu na to, że  $\tilde{\beta}_w$  jest składnikiem nieredukowalnego d.n.w. funkcji  $p^z$ , a gwiazdy  $G(e_j)$ ,  $e_j \in F^x \setminus E$ , są rozłączne, otrzymane pokrycie jest pokryciem nieredukowalnym, to znaczy poszukiwanym pokryciem  $N(T)$ . Q.E.D.

Twierdzenie 3.13. Moc pokrycia minimalnego  $M(T)$  jest równa  $\varphi + \Delta_{\min}$ , gdzie  $\Delta_{\min} = \min_{i \in \{1, \dots, b\}} \Delta_i$ .

Dowód wynika bezpośrednio z lematu 3.12 Q.E.D.

#### 3.5.4. Algorytm C określenia pokrycia minimalnego obrazu cyklicznego

Na podstawie rozważań w pktach 3.5.2. i 3.5.3. skonstruujemy algorytm C określenia pokrycia minimalnego  $M(T)$  obrazu cyklicznego:

1. Wyznacza się rodzinę gwiazd rozłącznych  $G^x$  oraz rodzinę gwiazd zależnych  $G^z$ . Rodzinę  $G^x$  konstruuje się według algorytmu R. Każdemu polu  $e_j \in F^1$  przypisuje się zbiór

$$I(e_j) = \left\{ (\underline{j}, \underline{k}) : I_{\underline{j}, \underline{k}} \in G(e_j) \right\}.$$

Przy ręcznym wykonywaniu algorytmu zbiory  $I(e_j)$  wpisuje się w poszczególne pola  $e_j$ .

2. Określa się składniki  $\beta_1$  nieredukowalnego d.n.w. funkcji  $p^z$  oraz odpowiadające im parametry  $\Delta_1$ .

3. Wśród składników  $\beta_1$  określa się zbiór  $\{\beta_i\}_{i \in J}$  składników, którym odpowiadają parametry  $\Delta_i = \Delta_{\min}$ .

4. W zbiorze  $\{\beta_i\}_{i \in J}$  określa się składnik  $\beta_m$ , któremu odpowiada zbiór  $\tilde{\beta}_m$  o minimalnej złożoności  $z(\tilde{\beta}_m)$ .

5. Dla zbioru  $\tilde{\beta}_m$  określa się pokrycie nieredukowalne  $N(\tilde{\beta}_m) = N(T) \supseteq \tilde{\beta}_m$  tak jak podano w dowodzie lematu 3.12, z tym jednak, że z gwiazd  $G(e_j)$ ,  $e_j \in F^x \setminus E$ , wybiera się zespoły posiadające

najmniejszą złożoność. Otrzymane pokrycie będzie pokryciem najkrótszym o minimalnej złożoności, to znaczy poszukiwanym pokryciem minimalnym  $M(T)$ .

W szczególnych przypadkach algorytm C może być uproszczony. Załóżmy, że określając kolejno zbiory  $\tilde{\beta}_i$  oraz parametry  $\Delta_i$  dla pewnego  $i = m$  parametr  $\Delta_m = 0$ . Jeśli można stwierdzić, że złożoność  $z(\tilde{\beta}_i)$  każdego innego zbioru  $\tilde{\beta}_i$ , dla którego  $\Delta_i = 0$ , nie może być mniejsza niż  $z(\tilde{\beta}_m)$  (taki przypadek zachodzi na przykład, gdy gwiazdy rodziny  $G^2$  składają się z zespołów o jednakowej złożoności), to pokrycie  $N(T) \supseteq \tilde{\beta}_m$ , określone jak w kroku 5 algorytmu C, będzie pokryciem minimalnym. W ten sposób określanie dalszych zbiorów  $\tilde{\beta}_i$ ,  $i > m$ , staje się zbędne.

### 3.6. Algorytm $A^m$ syntezy pokrycia minimalnego dowolnego obrazu $T(f)$

#### 3.6.1. Opis algorytmu

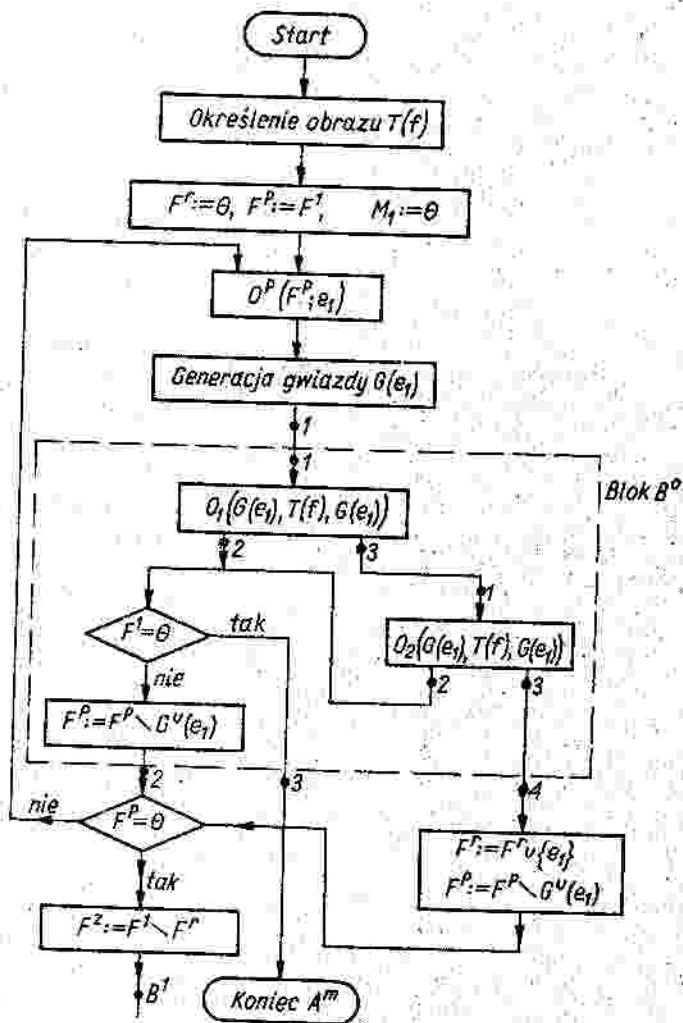
Poniżej formułuje się algorytm  $A^m$  określania pokryć minimalnych dowolnego obrazu  $T(f)$ . W algorytmie tym wykorzystuje się wyniki uzyskane w pktach 3.4. i 3.5. Idea konstrukcji algorytmu jest następująca:

Generuje się rodzinę gwiazd rozłącznych  $G^2$ , a następnie rodzinę gwiazd zależnych  $G^2$ ; jednocześnie wykonuje się operacje upraszczające  $O_1, O_2, O_3$ . Operacje te następnie powtarza się dopóty, dopóki nie uzyska się albo pokrycia minimalnego  $M(T)$ , albo obraz  $T(f)$  nie stanie się obrazem cyklicznym. W przypadku gdy obraz  $T(f)$  stanie się obrazem cyklicznym, wówczas stosuje się algorytm C.

Algorytm  $A^m$  składa się z kolejno wykonywanych algorytmów pomocniczych  $B^1, B^2, \dots, B^5$  oraz z algorytmu C. Obecnie przejdziemy do opisu algorytmów  $B^i$ ,  $i = 1, \dots, 5$ , przy czym algorytmy  $B^1, B^2, B^4$  jako bardziej złożone opisuje się ze pomocą wykresów operacyjnych.

Przez  $F^1, F^*, F^0, F^2, F^3, F^5, F^D$  rozumie się w każdym z opisywanych algorytmów pewne zmienne, których wartościami są zbiory. Wartościami zmiennych  $F^1, F^*, F^0$  na początku algorytmu  $A^m$  są

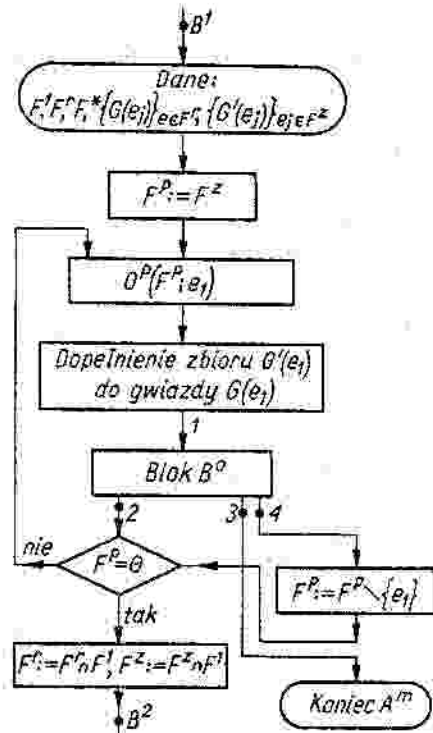
zbiory pól, które w danym obrazie  $T(f)$  posiadają wartości odpowiednio 1, \*, 0.



Rys. 3.12. Wykres operacyjny algorytmu  $B^1$

Algorytm  $B^1$ . W algorytmie tym określa się rodzinę gwiazd rozłącznych  $G^x = \{G(e_j)\}_{e_j \in P^x}$ , przy czym po określeniu każdej

gwiazdy  $G(e_j)$  wykonuje się operacje lokalne  $O_1$  i  $O_2$ .



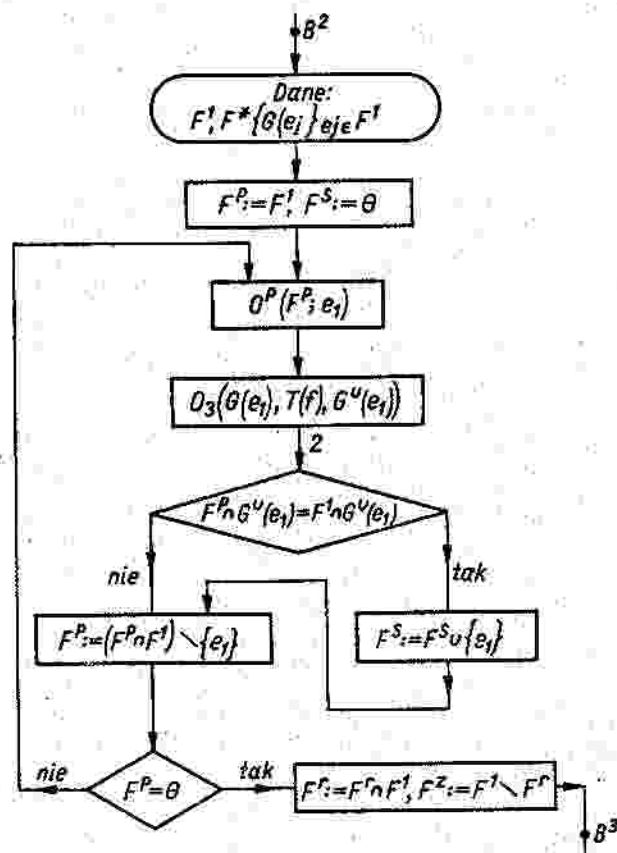
Rys. 3.13. Wykres operacyjny algorytmu  $B^2$ .

Algorytm  $B^2$ . W algorytmie tym uzupełnia się zbiory  $G'(e_j)$ ,  $e_j \in F^2$ , w nowe zespoły maksymalne tak, aby otrzymać gwiazdy  $G(e_j)$ . Po określeniu każdej kolejnej gwiazdy  $G(e_j)$  wykonuje się nad nią operacje  $O_1$  i  $O_2$ . Zmienna  $F^p$  gra rolę pomocniczą. Jeśli wykonanie algorytmu  $B^2$  spowodowało uproszczenie obrazu  $T(f)$ , to wykonuje się algorytm  $B^3$ ; w przeciwnym przypadku wykonuje się algorytm  $B^4$ .

Algorytm  $B^3$ . Polega on na wykonywaniu operacji  $O_1$  i  $O_2$  nad gwiazdami  $G(e_j)$ ,  $e_j \in F^1$ , dopóty, dopóki obraz  $T(f)$  nie stanie się nieupraszczalny za pomocą tych operacji.

Algorytm  $B^4$ . W algorytmie tym wykonuje się operację  $O_3$  kolejno pod gwiazdami  $G(e_j)$  pól  $e_j \in F^1$ . Łatwo zauważyć, że wykonanie algorytmu  $B^4$  prowadzi do redukcji wszystkich pól redukowalnych w aktualnym obrazie  $T(f)$ . Wartość zmiennej  $F^p$  określa

zbiór wszystkich pól  $e_j$  o takiej własności, że operacja  $O_3$  nad gwiazdą  $G(e_j)$  nie prowadzi do uproszczenia obrazu  $T(f)$ .



Rys. 3.14. Algorytm  $B^4$ . Redukcja pól redukowalnych.

Jeśli wykonanie algorytmu  $B^4$  spowodowało uproszczenie obrazu  $T(f)$ , tzn. jeśli  $F^S \neq F^1$ , to wykonuje się algorytm  $B^5$ ; w przeciwnym przypadku wykonuje się algorytm  $C$ .

Algorytm  $B^5$ . Polega on na wykonywaniu na przemian algorytmów  $B^3$  i  $B^4$  dopóty, dopóki obraz nie stanie się nieupraszczalny za pomocą operacji  $O_1, O_2, O_3$ . Jeśli po wykonaniu algorytmu  $B^4$  przechodzi się do wykonania algorytmu  $B^3$ , to w algorytmie  $B^3$  operacje  $O_1$  i  $O_2$  wykonuje się tylko nad gwiazdami pól  $e_j \in F^1 \setminus F^S$ .



Jeśli w rezultacie algorytmu  $B^5$  zbiór  $F^1 \neq 0$ , to oznacza to, że aktualny obraz  $T(f)$  jest obrazem cyklicznym. Wówczas wykonuje się algorytm C opisany w pktcie 3.5.3. Rodzinę gwiazd rozłącznych  $G^x$  tworzy zbiór gwiazd zredukowanych  $\{G(e_j)\}_{e_j \in F^x}$ , a rodzinę gwiazd zależnych  $G^z$  - zbiór gwiazd (zredukowanych)  $\{G(e_j)\}_{e_j \in F^z}$ , gdzie  $F^x, F^z$  - ostatnie wartości zmiennych  $F^x$  i  $F^z$ . Ponieważ indeksy  $j$  pól  $e_j \in F^1$  mogą obecnie nie tworzyć ciągu kolejnych liczb naturalnych, więc należy wykonać operację  $OP\{F^x; e_1, \dots, e_f\}$  i następnie zmienić odpowiednio wartość indeksu  $j$  każdego zespołu  $L_{j,k} \in G(e_j)$ ,  $e_j \in F^1$ .

### 3.6.2. Przykład realizacji algorytmu

Obecnie na prostym przykładzie (zaczepniętym z pracy /44/) pokażemy proces syntezy pokrycia minimalnego  $M(T)$  za pomocą algorytmu  $A^m$ .

Przykład 3.3. Określić pokrycie minimalne  $M(T)$  obrazu  $T(f)$  przedstawionego na rys. 3.15.

	0	1	2	3	4	5	6	7
0								
$x_1$ 8								
16								
$x_2$ 24								
								$x_5$
				$x_3$			$x_4$	

Rys. 3.15. Obraz  $T(f)$  do przykładu 3.3

1. Realizuje się algorytm  $B^1$ . Obraz funkcji  $T(f)$  po zrealizowaniu algorytmu  $B^1$  przedstawiono na rys. 3.16 (pola, dla których generowano gwiazdy rozłączne, oznaczono +; pola, które zmieniły wartość z 1 na \* przekreślono). Na rys. 3.17 przedsta-

wiono gwiazdy rodziny  $G^F$  określone w algorytmie  $B^1$  (zespoły otoczone kółkiem są ekstremalami jądrowymi).

	0	1	2	3	4	5	6	7
0	$\begin{matrix} 1,1 \\ 1,2 \end{matrix}$	$\begin{matrix} 1,1 \\ 2,1 \end{matrix}$		$\begin{matrix} 2,2 \\ 2,1 \end{matrix}$	$\begin{matrix} 1,2 \\ 3,2 \end{matrix}$		$\begin{matrix} 3,1 \\ 3,2 \end{matrix}$	$\begin{matrix} 2,2 \\ 3,1 \end{matrix}$
$x_1$			$\begin{matrix} 4,1 \\ 4,1 \end{matrix}$	$\begin{matrix} 2,2 \\ 3,1 \end{matrix}$	$\begin{matrix} 3,2 \\ 5,1 \end{matrix}$	$\begin{matrix} 5,1 \\ 5,1 \end{matrix}$	$\begin{matrix} 3,2 \\ 3,1 \end{matrix}$	$\begin{matrix} 3,1 \\ 4,1 \end{matrix}$
$x_2$	$\begin{matrix} 1,1 \\ 7,1 \end{matrix}$	$\begin{matrix} 1,1 \\ 6,1 \end{matrix}$				$\begin{matrix} 6,1 \\ 6,2 \end{matrix}$	$\begin{matrix} 3,1 \\ 6,2 \end{matrix}$	$\begin{matrix} 6,2 \\ 3,1 \end{matrix}$
	16							
	24	$\begin{matrix} 7,1 \\ 7,1 \end{matrix}$	$\begin{matrix} 4,1 \\ 7,2 \end{matrix}$	$\begin{matrix} 4,1 \\ 7,2 \end{matrix}$	$\begin{matrix} 5,1 \\ 7,2 \end{matrix}$	$\begin{matrix} 5,1 \\ 6,2 \end{matrix}$	$\begin{matrix} 4,1 \\ 7,2 \end{matrix}$	$\begin{matrix} 4,1 \\ 5,1 \end{matrix}$
							$x_4$	$x_5$
					$x_3$			

Rys. 3.16. Obraz  $T(f)$  po zrealizowaniu algorytmu  $B^1$

$\gamma(e_j)$	$G(e_j)$	Rozmiary $L_{j,k} \in G(e_j)$
0	$L_{1,1}, L_{1,2}$	2, 1
3	$L_{2,1}, L_{2,2}$	1, 2
6	$L_{3,1}, L_{3,2}$	3, 1
10	$L_{4,1}$	3
13	$L_{5,1}$	3
21	$L_{6,1}, L_{6,2}$	1, 2
24	$L_{7,1}, L_{7,2}$	1, 2

Rys. 3.17. Gwiazdy rodziny  $G^F$  w obrazie  $T(f)$  z rys. 3.15.

2. Realizujemy algorytm  $B^2$ . W wyniku operacji  $O_1$  nad gwiazdą  $G(e^{22})$  otrzymuje się ekstremalę jądrową  $L_{3,1}$ . Pola  $e^6, e^7$  i  $e^{23}$  zmieniają wartość z 1 na  $*$ .

3. Realizuje się algorytm  $B^3$ . Obraz  $T(f)$  pozostaje bez zmiany.

4. Realizuje się algorytm  $B^4$ . Obraz  $T(f)$  pozostaje bez zmiany. Otrzymany obraz cykliczny oznacza się  $T'(f)$ .

5. Realizuje się algorytm C. Zbiór  $F^x = \{e^0, e^3, e^{21}, e^{24}\} = \{e_i\}_{i=1}^4$ . Obraz  $T(f)$  po dokonaniu odpowiednich zmian wartości indeksów  $j$  w gwiazdkach  $G(e_j)$ ,  $e_j \in F^1$ , przedstawiono na rys. 3.18.

	$\begin{matrix} 1,1 \\ 1,2 \end{matrix} +$	$\begin{matrix} 1,1 \\ 1,2 \end{matrix}$		$\begin{matrix} 1,1 \\ 2,1 \end{matrix} +$	$\begin{matrix} 1,2 \\ 1,2 \end{matrix}$		*	*
			*	*	*	*	*	*
$x_1$	$\begin{matrix} 1,1 \\ 4,1 \end{matrix}$	$\begin{matrix} 1,1 \\ 3,1 \end{matrix}$				$\begin{matrix} 3,2 \\ 3,1 \end{matrix} +$	*	*
$x_2$	$\begin{matrix} 4,2 \\ 4,1 \end{matrix} +$		*	*	*	*	*	*
							$x_4$	$x_5$
							$x_3$	

Rys. 3.18. Obraz  $T(f)$

$$p^z = (L_{1,1} \vee L_{2,1})(L_{1,2} \vee L_{5,1})(L_{1,1} \vee L_{4,1})(L_{1,1} \vee L_{3,1})$$

$$p^z = \bigvee_{i=1}^6 \beta_i$$

gdzie

$$\beta_1 = L_{1,1} L_{1,2}, \quad \beta_2 = L_{1,1} L_{5,1}$$

$$\beta_3 = L_{2,1} L_{1,2} L_{1,1}, \quad \beta_4 = L_{2,1} L_{1,2} L_{4,1} L_{3,1}$$

$$\beta_5 = L_{2,1} L_{5,1} L_{1,1}, \quad \beta_6 = L_{2,1} L_{5,1} L_{4,1} L_{3,1}$$

Określa się parametry  $\Delta_i$ ,  $i = 1, 2, \dots, 6$ , przyjmując  $\varphi = 4$ :  $\Delta_1 = 1, \Delta_2 = 1, \Delta_3 = 1, \Delta_4 = 0, \Delta_5 = 1, \Delta_6 = 1$ .

Określa się pokrycie  $N(\tilde{\beta}_4) = \{L_{1,2}, L_{2,1}, L_{3,1}, L_{4,1}\}$ . Jest to pokrycie minimalne (najkrótsze)  $M^k(T)$ . Moc tego pokrycia  $c_1 = 4$ , a złożoność  $z_1 = 16$ . Łatwo sprawdzić, że pokrycie  $N(\tilde{\beta}_2) = \{L_{1,1}, L_{2,2}, L_{3,2}, L_{4,2}, L_{5,1}\}$ , mimo że posiada moc  $c_2$  większą,  $c_2 = 5 > c_1$ , posiada złożoność  $z_2$  mniejszą,  $z_2 = 15 < z_1$ ,

niż pokrycie  $N(\tilde{\beta}_4)$ . Jest ono więc pokryciem minimalnym  $M^1(T')$ . Uzupełniając pokrycie  $N(\tilde{\beta}_4)$  w poprzednio określone ekstremale jądrowe i powracając do poprzednich wartości indeksów  $j$  otrzymuje się:

$$M(T) = \{L_{1,2}, L_{2,1}, L_{3,1}, L_{4,1}, L_{5,1}, L_{6,1}, L_{7,1}\}$$

Pokryciu  $M(T)$  odpowiada wyrażenie minimalne funkcji  $f$ :

$$M(f) = \alpha_{1,2} \vee \alpha_{2,1} \vee \alpha_{3,1} \vee \alpha_{4,1} \vee \alpha_{5,1} \vee \alpha_{6,1} \vee \alpha_{7,1}$$

gdzie

$$\alpha_{j,k} = \alpha(L_{j,k}) \quad \alpha_{4,1} = x_2 x_4$$

$$\alpha_{1,2} = \bar{x}_1 \bar{x}_2 \bar{x}_4 \bar{x}_5 \quad \alpha_{5,1} = x_2 x_3$$

$$\alpha_{2,1} = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \quad \alpha_{6,1} = x_1 \bar{x}_2 \bar{x}_4 x_5$$

$$\alpha_{3,1} = x_3 x_4 \quad \alpha_{7,1} = x_1 \bar{x}_3 \bar{x}_4 x_5$$

Obraz  $T(f)$  w powyższym przykładzie jest bardzo prosty i jego pokrycie minimalne  $M(T)$  można by znaleźć nawet metodą prób i błędów (pkt 3.2). W przykładzie chodzi tylko o zilustrowanie procesu syntezy pokrycia  $M(T)$  za pomocą ścisłego algorytmu  $A^m$ , nie zaś o określenie tego pokrycia w sposób najprostsz y w danym szczególnym przypadku.

### 3.6.3. Charakterystyka algorytmu $A^m$

Algorytm  $A^m$  syntezy pokrycia minimalnego stanowi połączenie metody lokalnego wydobywania ekstremali z metodą rozłącznych gwiazd będącą pewną modyfikacją rozwiązania algebrológicznego Petrica. Zastosowanie operacji lokalnych umożliwia zredukowanie w sposób stosunkowo prosty (bez na przykład konieczności uprzedniego określenia wszystkich zespołów maksymalnych) danego obrazu  $T(f)$  do obrazu cyklicznego. Metoda rozłącznych gwiazd (algorytm C) umożliwia zmniejszenie liczby operacji w stosunku do rozwiązania algebrológicznego przy określaniu pokrycia minimalnego obrazu cyklicz-

nego. Zysk ten jest szczególnie widoczny wówczas, gdy na początku procesu określania składników  $\beta_i$  znajduje się składnik  $\beta_m$ , dla którego  $\Delta_m = 0$ . Jeśli bowiem pokrycie  $N(T) \geq \beta_m$ , określone jak w kroku 5 algorytmu C, przyjmiemy za rozwiązanie (jest ono bądź pokryciem  $M(T)$ , bądź pokryciem najkrótszym, które może tylko nieznacznie różnić się od pokrycia  $M(T)$ ), to odpada konieczność określania pozostałych składników.

W przypadku jednak obrazów funkcji logicznych dużej liczby zmiennych (szczególnie w przypadkach złożonych obrazów cyklicznych, które posiadają na przykład funkcje skonstruowane przez Wasiliewa /25/) algorytm  $A^m$  może prowadzić do konieczności wykonania zbyt wielkiej liczby operacji. Wyjściem w takiej sytuacji jest zastosowanie odpowiednio prostego algorytmu dostarczającego rozwiązanie przybliżone. Algorytm taki konstruuje się w pktcie 3.7.

### 3.7. Algorytm $A^q$ syntezy pokrycia quasi-minimalnego

#### 3.7.1. Opis algorytmu

Na podstawie wyników uzyskanych w pktcie 3.5 skonstruujemy obecnie pewien algorytm  $A^q$  syntezy pokrycia w przybliżeniu minimalnego dowolnego obrazu  $T(f)$ . W wyniku realizacji tego algorytmu otrzymuje się pewne pokrycie  $M^q(T)$ , nazywane pokryciem quasi-minimalnym i jednocześnie uzyskuje się oszacowanie maksymalnej możliwej różnicy między tym pokryciem a pokryciem minimalnym  $M(T)$ . Różnicę tę wyraża się w liczbie elementów  $\Delta$  i w złożoności  $\delta$ :

$$c(M^q(T)) - c(M(T)) \leq \Delta \quad (3.12)$$

$$z(M^q(T)) - z(M(T)) \leq \delta \quad (3.13)$$

Idea konstrukcji algorytmu jest następująca. Nad każdą gwiazdą  $G(e)$  rodziny gwiazd rozłącznych wykonuje się operację lokalną, polegającą na wyborze z danej gwiazdy pewnego zespołu maksymalnego nazywanego quasi-ekstremalą (definicja 3.14). Jeśli uzyskany w ten sposób zbiór nie stanowi pokrycia obrazu  $T(f)$ , to wówczas dopeknia się go dodatkowymi zespołami tak, aby uzyskać pokrycie. Liczba i suma złożoności dodatkowych zespołów określa parametry  $\Delta$  i  $\delta$ .

(w określenie ostatecznej wartości  $\delta$  wchodzi jeszcze pewna dodatkowa wielkość).

Niech dany jest pewien obraz  $T(f) = F^1 \cup F^* \cup F^0$ , gdzie  $F^1, F^*, F^0$  są zbiorami - wartościami zmiennych  $F^1, F^*, F^0$  po zrealizowaniu pewnego kroku algorytmu  $A^q$ . Na początku algorytmu  $A^q$ , tak jak w algorytmie  $A^m$ , wartościami zmiennych  $F^1, F^*, F^0$  są zbiory pól, które w wyjściowym obrazie  $T(f)$  posiadają wartości odpowiednio 1, \*, 0.

Definicja 3.14. Quasi-ekstremalna gwiazda  $G(e_j)$ ,  $e_j \in F^1$ , w obrazie  $T(f)$  nazywa się zespołem maksymalny  $L_k \in G(e_j)$ , który pokrywa maksymalną liczbę pól zbioru  $F^1$  i wśród zespołów pokrywających taką samą liczbę pól zbioru  $F^1$  posiada minimalną złożoność.

Quasi-ekstremalną gwiazdę  $G(e_j)$  w obrazie  $T(f)$  oznacza się  $L^q(e_j, F^1)$  lub - krótko -  $L^q$ , jeśli wiadomo, o jaką gwiazdę i jaki obraz, a konkretnie, o jaką wartość  $F^1$  chodzi. Zespół maksymalny w danej gwiazdzie  $G(e_j)$ , posiadający minimalną złożoność oznacza się  $L^m(e_j)$  lub - krótko -  $L^m$ .

W ogólnym przypadku gwiazda  $G(e)$  w danym obrazie  $T(f)$  może zawierać więcej niż jedną quasi-ekstremalną, a także więcej niż jeden zespół o minimalnej złożoności. W tym przypadku pisząc  $L^q$  lub  $L^m$  będziemy mieli na myśli dowolny taki zespół (na przykład pierwszy w kolejności ich określania przy generowaniu gwiazdy  $G(e_j)$ ).

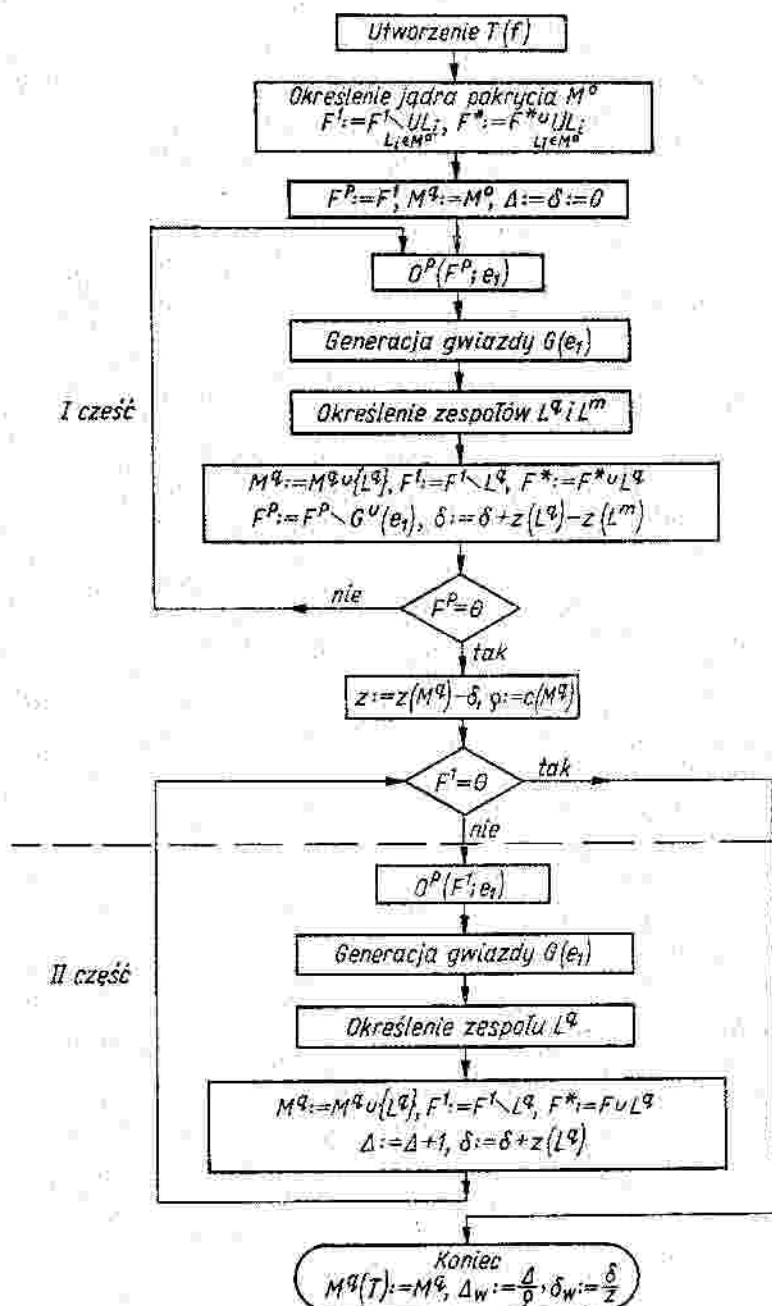
Wykres operacyjny algorytmu  $A^q$  przedstawiono na rys. 3.19. Algorytm ten składa się z dwóch części.

Część I. W pierwszym kroku tej części określa się jądro pokrycia  $M^0$  i zmienia wartości pól jądra obrazu z 1 na \*. Operację tę wykonuje się w ten sposób, że kolejno dla każdego pola  $e \in F^1$  generuje się gwiazdę  $G(e)$  i sprawdza się, czy jest spełniony warunek

$$c(G(e)) = 1 \tag{a}$$

Jeśli warunek (a) jest spełniony, to zespół maksymalny  $L \in G(e)$  włącza się do zbioru  $M^0$  i wykonuje działania:

$$F^1 := F^1 \setminus L, \quad F^* := F^* \cup L \tag{3.14}$$



Rys. 3.19. Algorytm  $\Delta^q$

Jeśli warunek (a) nie jest spełniony, to nie wykonuje się żadnych działań. Ponieważ w powyższej operacji chodzi wyłącznie o sprawdzenie warunku (a), więc proces generacji gwiazdy  $G(e)$  przerywa się za każdym razem z chwilą określenia dwóch zespołów maksymalnych. Można tu posłużyć się jeszcze prostszym algorytmem. Określa się zespoły rozmiaru 1 pokrywające dane pole. Jeśli stwierdzamy, że dowolna para takich zespołów nie może wejść w skład jednego zespołu o większych rozmiarach, to oznacza to, że nie może być spełniony warunek (a).

Następnie kolejno generuje się gwiazdy rodziny gwiazd rozłącznych (tak jak w algorytmie R) i z każdej z nich wybiera się quasi-ekstremalę  $L^q$ . Zespoły  $L^q$  włącza się do zbioru  $M^q$  i wykonuje za każdym razem działania (3.14). Określa się również sumę różnic złożoności zespołów  $L^q$  i  $L^m$  w kolejno określonych gwiazdach rozłącznych (parametr  $\delta$ ). Po wykonaniu tych operacji dla ostatniej gwiazdy rozłącznej (gdzie  $F^p = \emptyset$ ), oblicza się parametry  $\varphi = c(M^q)$  i  $z = z(M^q) - \delta$ . Parametr  $\varphi$ , zgodnie z twierdzeniem 3.11, określa minimalną liczbę elementów, jakie musi posiadać pokrycie minimalne  $M(T)$ . Parametr  $z$  określa złożoność podzbioru utworzonego z zespołów o minimalnej złożoności, zawartych w poszczególnych gwiazdach rozłącznych, a więc wyraża minimalną złożoność, jaką musi posiadać pokrycie  $M(T)$ . Jeśli po wykonaniu części I zbiór  $F^1 = \emptyset$ , to wówczas uzyskany zbiór  $M^q$  stanowi pokrycie obrazu  $T(f)$ . Pokrycie to posiada moc równą mocy pokrycia minimalnego  $M(T)$ , a złożoność jego może się różnić od złożoności pokrycia minimalnego co najwyżej o  $\delta$ . Jeśli  $F^1 = \emptyset$ , to przechodzi się do wykonania części II.

Część II. Wykonuje się następującą operację: określa się gwiazdę pola o najniższym numerze w zbiorze  $F^1$ , wybiera się quasi-ekstremalę  $L^q$  i wykonuje działania (3.14). Dalejsze kroki algorytmu polegają na powtórzeniu powyższej operacji dopóty, dopóki  $F^1$  nie równa się  $\emptyset$ . Uzyskany w ten sposób zbiór  $M^q$  stanowi pewne nieredukowalne pokrycie obrazu  $T(f)$ .

Parametr  $\Delta$ , którego wartość jest równa liczbie kroków wykonanych w części II, określa maksymalną możliwą różnicę między mocą otrzymanego pokrycia a mocą pokrycia  $M(T)$ . Parametr  $\delta$  zwiększa się po wykonaniu części II o sumę złożoności określonych w tej części quasi-ekstremal. Parametr ten wyraża więc maksymalną możli-



wą różnicę między złożonością pokrycia  $M^q$  a złożonością pokrycia  $M(T)$ . Parametry  $\Delta_w = \Delta/\varphi$  i  $\delta_w = \delta/z$  oszacowują maksymalną względną różnicę w liczbie elementów i w złożoności między pokryciem  $M^q$  a pokryciem  $M(T)$ .

Pokrycie  $M^q$ , które uzyskuje się w wyniku realizacji algorytmu  $A^q$ , nazywa się pokryciem quasi-minimalnym obrazu  $T(f)$  i oznacza się symbolem  $M^q(T)$ . Zgodnie z powyższymi rozważaniami pokrycie  $M^q(T)$  spełnia nierówności (3.12) i (3.13).

Jeśli powtórzyć algorytm  $A^q$  wybierając z poszczególnych gwiazd inne quasi-ekstremale lub inne zespoły (na przykład zespoły o najmniejszej złożoności) albo też przyjmując inny porządek określania gwiazd rozłącznych (na przykład przez rozpoczęcie algorytmu z innego pola), parametry  $\Delta$  i  $\delta$  mogą wypaść mniejsze. W ten sposób przez powtarzanie algorytmu  $A^q$  można uzyskać poprawienie wyniku. Wykonywanie tego rodzaju prób jest celowe wtedy, kiedy "koszt" ich wykonania (określony na przykład czasem pracy maszyny cyfrowej) jest wyraźnie mniejszy niż "strata" (określona parametrami  $\Delta$  i  $\delta$ ), jaką możemy ponieść przez przyjęcie rozwiązania przybliżonego zamiast rozwiązania minimalnego.

W przypadku gdy  $\Delta = 0$  i  $\delta = 0$ , pokrycie quasi-minimalne  $M^q(T)$  jest równe pokryciu minimalnemu  $M(T)$ . Warto podkreślić, że gdy  $\Delta \neq 0$ ,  $\delta \neq 0$  pokrycie  $M^q(T)$  również może w rzeczywistości być równe pokryciu  $M(T)$ , lecz informacja, jaką posiadamy pozwala nam stwierdzić jedynie słuszność nierówności (3.12) i (3.13).

Przy realizacji algorytmu  $A^q$  można, zasadniczo biorąc, nie wykonywać operacji określania jądra pokrycia  $M^q$ . W tym jednak przypadku może zdarzyć się, że niektóre gwiazdy rodziny gwiazd rozłącznych będą składały się z ekstremal jądrowych. Wybierając z tych gwiazd quasi-ekstremale, wybiera się tylko jedną ekstremalę jądrową z każdej gwiazdy, pozostałe więc będziemy musieli wybrać realizując część II algorytmu. To spowoduje odpowiednie zwiększenie parametrów  $\Delta$  i  $\delta$ , mimo że wszystkie te zespoły jako jądrowe będą zawierać się zarówno w pokryciu  $M^q(T)$  jak i w pokryciu  $M(T)$ .

### 3.7.2. Przykład realizacji algorytmu

Przykład 3.4. Określić pokrycie  $M^q(T)$  obrazu  $T(f)$  funkcji

$$f = S_{0,2,3,4,6}^6$$

Funkcja symetryczna  $S_{0,2,3,4,6}^6$  posiada maksymalną liczbę prostych implikantów, jaką może mieć funkcja 6 zmiennych, a mianowicie 92 proste implikanty (zob. pkt 5.1.3).

	0	1	2	3	4	5	6	7
0	1,1			2,1 2,4 2,2 + 2,5 2,3 2,6		3,1 3,4 3,2 + 3,5 3,3 3,6	4,1 4,4 4,2   4,5 4,3 4,6	5,1 5,4 5,2 + 5,5 5,3 5,6
8		5,1 5,4 5,2 + 5,5 5,3 5,6	6,1 6,4 6,2 + 6,5 6,3 6,6	7,1 7,4 7,2 + 7,5 7,3 7,6	8,1 8,4 8,2 + 8,5 8,3 8,6	9,1 9,4 9,2 + 9,5 9,3 9,6	10,1 10,4 10,2 + 10,5 10,3 10,6	11,1 11,4 11,2 + 11,5 11,3 11,6
16		11,1 11,4 11,2 + 11,5 11,3 11,6	12,1 12,4 12,2 + 12,5 12,3 12,6	13,1 13,4 13,2 + 13,5 13,3 13,6	14,1 14,4 14,2 + 14,5 14,3 14,6	15,1 15,4 15,2 + 15,5 15,3 15,6	16,1 16,4 16,2 + 16,5 16,3 16,6	17,1 17,4 17,2 + 17,5 17,3 17,6
24		17,1 17,4 17,2 + 17,5 17,3 17,6	18,1 18,4 18,2 + 18,5 18,3 18,6	19,1 19,4 19,2 + 19,5 19,3 19,6	20,1 20,4 20,2 + 20,5 20,3 20,6	21,1 21,4 21,2 + 21,5 21,3 21,6	22,1 22,4 22,2 + 22,5 22,3 22,6	23,1 23,4 23,2 + 23,5 23,3 23,6
$x_1$		23,1 23,4 23,2 + 23,5 23,3 23,6	24,1 24,4 24,2 + 24,5 24,3 24,6	25,1 25,4 25,2 + 25,5 25,3 25,6	26,1 26,4 26,2 + 26,5 26,3 26,6	27,1 27,4 27,2 + 27,5 27,3 27,6	28,1 28,4 28,2 + 28,5 28,3 28,6	29,1 29,4 29,2 + 29,5 29,3 29,6
32		29,1 29,4 29,2 + 29,5 29,3 29,6	30,1 30,4 30,2 + 30,5 30,3 30,6	31,1 31,4 31,2 + 31,5 31,3 31,6	32,1 32,4 32,2 + 32,5 32,3 32,6	33,1 33,4 33,2 + 33,5 33,3 33,6	34,1 34,4 34,2 + 34,5 34,3 34,6	35,1 35,4 35,2 + 35,5 35,3 35,6
40		35,1 35,4 35,2 + 35,5 35,3 35,6	36,1 36,4 36,2 + 36,5 36,3 36,6	37,1 37,4 37,2 + 37,5 37,3 37,6	38,1 38,4 38,2 + 38,5 38,3 38,6	39,1 39,4 39,2 + 39,5 39,3 39,6	40,1 40,4 40,2 + 40,5 40,3 40,6	41,1 41,4 41,2 + 41,5 41,3 41,6
$x_2$		41,1 41,4 41,2 + 41,5 41,3 41,6	42,1 42,4 42,2 + 42,5 42,3 42,6	43,1 43,4 43,2 + 43,5 43,3 43,6	44,1 44,4 44,2 + 44,5 44,3 44,6	45,1 45,4 45,2 + 45,5 45,3 45,6	46,1 46,4 46,2 + 46,5 46,3 46,6	47,1 47,4 47,2 + 47,5 47,3 47,6
48		47,1 47,4 47,2 + 47,5 47,3 47,6	48,1 48,4 48,2 + 48,5 48,3 48,6	49,1 49,4 49,2 + 49,5 49,3 49,6	50,1 50,4 50,2 + 50,5 50,3 50,6	51,1 51,4 51,2 + 51,5 51,3 51,6	52,1 52,4 52,2 + 52,5 52,3 52,6	53,1 53,4 53,2 + 53,5 53,3 53,6
$x_3$		53,1 53,4 53,2 + 53,5 53,3 53,6	54,1 54,4 54,2 + 54,5 54,3 54,6	55,1 55,4 55,2 + 55,5 55,3 55,6	56,1 56,4 56,2 + 56,5 56,3 56,6	57,1 57,4 57,2 + 57,5 57,3 57,6	58,1 58,4 58,2 + 58,5 58,3 58,6	59,1 59,4 59,2 + 59,5 59,3 59,6
56		59,1 59,4 59,2 + 59,5 59,3 59,6	60,1 60,4 60,2 + 60,5 60,3 60,6	61,1 61,4 61,2 + 61,5 61,3 61,6	62,1 62,4 62,2 + 62,5 62,3 62,6	63,1 63,4 63,2 + 63,5 63,3 63,6	64,1 64,4 64,2 + 64,5 64,3 64,6	65,1 65,4 65,2 + 65,5 65,3 65,6
$x_4$		65,1 65,4 65,2 + 65,5 65,3 65,6	66,1 66,4 66,2 + 66,5 66,3 66,6	67,1 67,4 67,2 + 67,5 67,3 67,6	68,1 68,4 68,2 + 68,5 68,3 68,6	69,1 69,4 69,2 + 69,5 69,3 69,6	70,1 70,4 70,2 + 70,5 70,3 70,6	71,1 71,4 71,2 + 71,5 71,3 71,6
$x_5$		71,1 71,4 71,2 + 71,5 71,3 71,6	72,1 72,4 72,2 + 72,5 72,3 72,6	73,1 73,4 73,2 + 73,5 73,3 73,6	74,1 74,4 74,2 + 74,5 74,3 74,6	75,1 75,4 75,2 + 75,5 75,3 75,6	76,1 76,4 76,2 + 76,5 76,3 76,6	77,1 77,4 77,2 + 77,5 77,3 77,6
$x_6$		77,1 77,4 77,2 + 77,5 77,3 77,6	78,1 78,4 78,2 + 78,5 78,3 78,6	79,1 79,4 79,2 + 79,5 79,3 79,6	80,1 80,4 80,2 + 80,5 80,3 80,6	81,1 81,4 81,2 + 81,5 81,3 81,6	82,1 82,4 82,2 + 82,5 82,3 82,6	83,1 83,4 83,2 + 83,5 83,3 83,6

Rys. 3.20. Obraz funkcji  $S_{0,2,3,4,6}^6$  po zrealizowaniu algorytmu  $B^1$

Zanim przejdziemy do realizacji algorytmu  $A^2$ , rozważmy możliwości określenia pokrycia minimalnego  $M(T)$  obrazu tej funkcji na

podstawie algorytmu  $A^m$  lub algorytmu algebrologicznego Petrica (tj. za pomocą funkcji  $P$  - równ. (3.9)).

Obraz  $T(f)$  po zrealizowaniu algorytmu pomocniczego  $B^1$  przedstawiono na rys. 3.20.

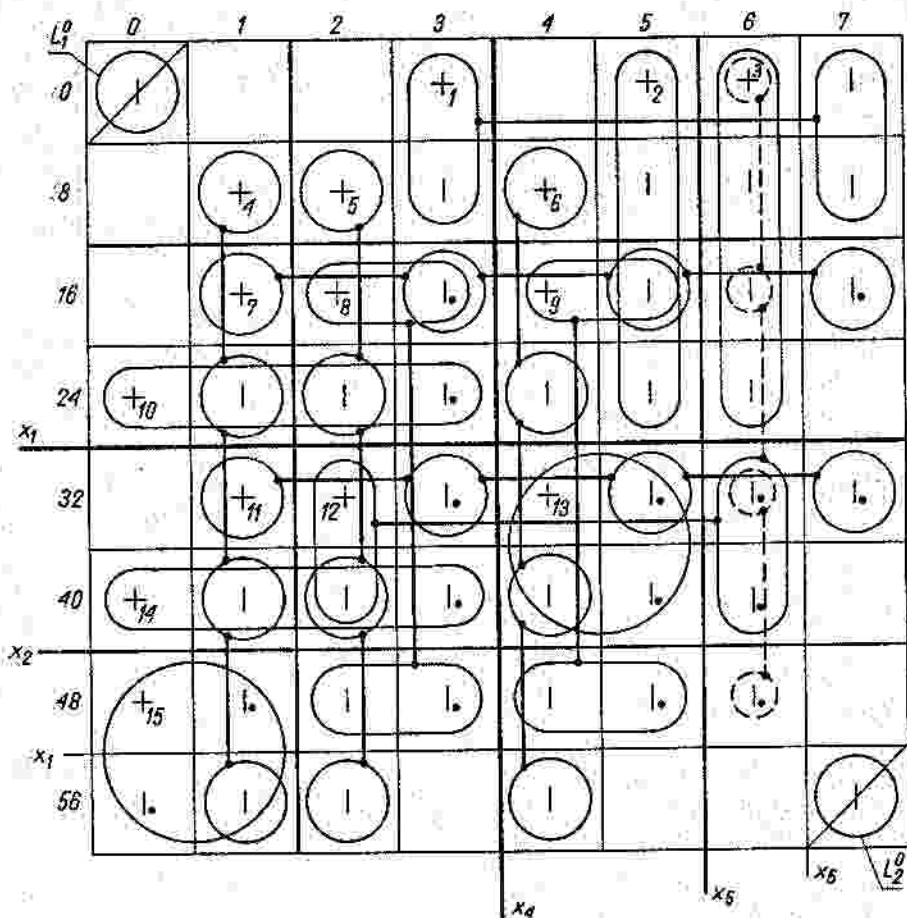
Pola, dla których generowano gwiazdy rozłączne, oznaczono znakiem +; pola, które zmieniły wartość z 1 na \* przekreślono. Zespoły  $L_{1,1} = \{e^0\}$  i  $L_{17,1} = \{e^{63}\}$  tworzą jądro pokrycia. Otrzymany obraz jest obrazem cyklicznym (algorytmy  $B^2 - B^5$ , jak łatwo zauważyć, nie prowadzą do jego uproszczenia). Określenie pokrycia minimalnego według algorytmu Petrica wymagałoby wymnożenia przez siebie 30 członów 6-elementowych i 20 członów 9-elementowych przy jednoczesnym stosowaniu praw pochłaniania (niewykonywanie uproszczeń wynikających z praw pochłaniania prowadziłyby do otrzymania  $6^{30} \cdot 9^{20} \approx 10^{23} \cdot 10^{19} \approx 10^{42}$  składników w d.n.w. funkcji  $p$ ). Następnie drogą przeglądu należałoby wybrać składnik otrzymanego d.n.w. odpowiadający pokryciu minimalnemu. Wykonanie tych operacji wydaje się nierealizowalne nawet na bardzo szybkiej maszynie cyfrowej. Również zastosowanie metody Lawlera /14/ (najnowszej znanej metody, którą można by w tym przypadku zastosować) nie prowadzi, jak wykazują orientacyjne rozważania, do wyraźnego zmniejszenia liczby operacji.

Przy zastosowaniu algorytmu C należałoby określić składniki  $\beta_i$  d.n.w. funkcji  $p^z$ , której k.n.w. zawiera 15 członów 6-elementowych i 20 członów 9-elementowych. Następnie należałoby obliczyć dla nich parametry  $\Delta_i$  i znaleźć wśród nich minimalny, który umożliwiłby określenie pokrycia  $M(T)$ . Jeśli składniki  $\beta_i$  określać kolejno i dla pewnego  $\beta_m$  parametr  $\Delta_m = 0$ , to określanie dalszych składników byłoby zbędne, ponieważ pokrycie  $N(T) \supset \tilde{\beta}_m$ , określone jak w kroku 5 algorytmu C, będzie pokryciem minimalnym. Ze względu bowiem na jednakowe rozmiary zespołów makeymalnych każdy inny zbiór  $\tilde{\beta}_i$ , dla którego  $\Delta_i = 0$ , będzie miał taką samą złożoność jak zbiór  $\tilde{\beta}_m$ .

Mimo że sposób ten daje możliwość uzyskania dość znacznej redukcji bezwzględnej liczby operacji, to jednak jest również zbyt pracochłonny (przy założeniu, że nie zajdzie korzystny przypadek znalezienia składnika  $\beta_m$  na początku procesu określania składników  $\beta_i$ ).

Wykonajmy algorytm  $A^q$ . Na rys. 3.21. przedstawiono obraz  $T(f)$

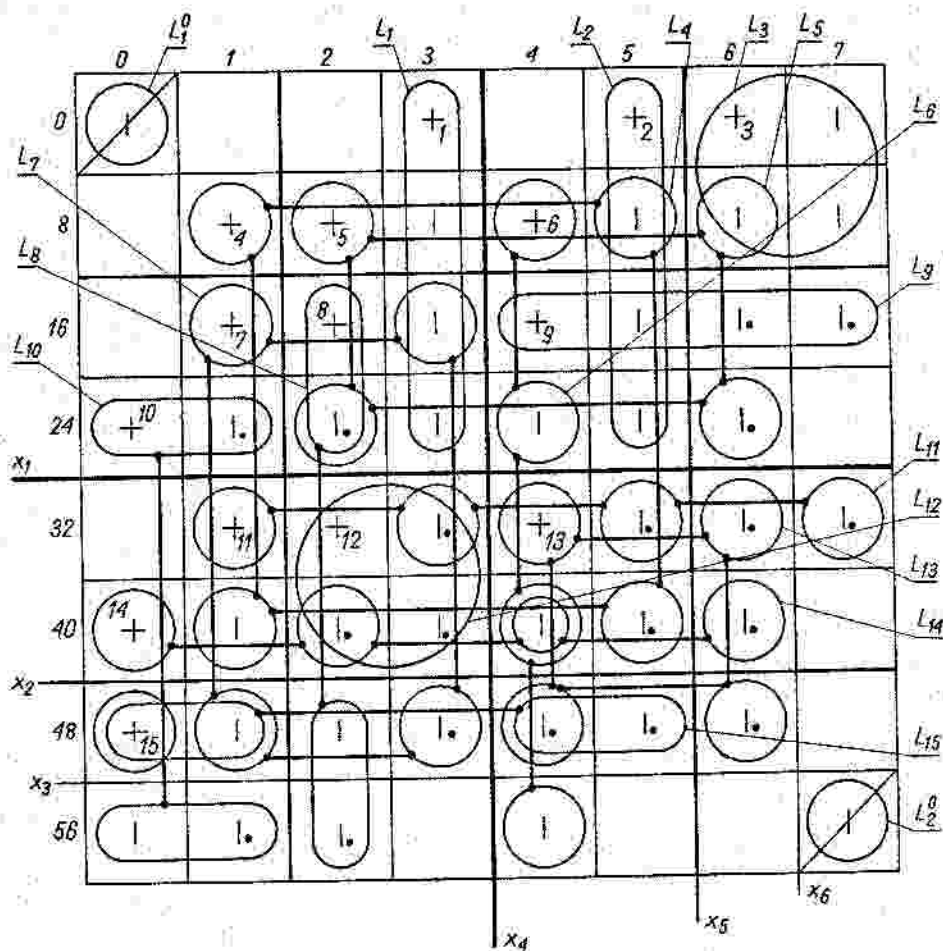
wraz z zaznaczonym pewnym pokryciem quasi-minimalnym  $M_1^q(T)$  (z równoważnych quasi-ekstremal wybierano takie, aby suma numerów pól będących ich elementami była minimalna).



Rys. 3.21. Pokrycie  $M_1^q(T)$

Pokrycie  $M_1^q(T)$  składa się z 18 zespołów maksymalnych, przy czym  $\Delta = 1$ ,  $\delta = 4$ . Zespoły  $L_1^0$ ,  $L_2^0$  tworzą jądro pokrycia; quasi-ekstremalę określoną w części II algorytmu zaznaczono linią

przerywaną. Liczby wpisane obok znaków  $+$  wskazują kolejność generowania gwiazd rozłącznych. Pola  $e \in F^1$  pokrywane przez inne zespoły danej gwiazdy rozłącznej (tj. pola  $G^U(e_i) \setminus L_i$ , gdzie  $L_i$  - wybrana z danej gwiazdy quasi-ekstremala) oznaczono kropką.



Rys. 3.22. Pokrycie  $M_2^0(T)$

Na rys. 3.22. przedstawiono obraz  $T(f)$  z zaznaczonym pokry-

ciem  $M_2^q(T)$ , które otrzymano przez wybór pewnych innych niż poprzednio quasi-ekstremal z gwiazd rozłącznych (w dwu przypadkach wybrano zespoły nie będące quasi-ekstremalami). Pokrycie to składa się z 17 zespołów maksymalnych, przy czym  $\Delta = 0$ ,  $\delta = 0$ . Pokrycie  $M_2^q(T)$  jest więc pokryciem minimalnym.

Z powyższego przykładu widać, że realizacja algorytmu  $A^q$  jest bardzo prosta; wymaga stosunkowo niedużej liczby operacji oraz niewielkiej pojemności pamięci przy realizacji maszynowej. Konieczne jest tylko pamiętanie zespołów jądra pokrycia, kolejno określanych quasi-ekstremal oraz informacji pozwalającej odróżnić pola zbioru  $F^1$  pokryte przez zespoły określonych do danego momentu gwiazd rozłącznych od pozostałych pól tego zbioru. Informację taką zapamiętuje się przez znacznikowanie odpowiednich pól zbioru  $F^1$  (znacznikowanie to wykonywano przez wpisywanie kropek).

Przy realizacji maszynowej znacznikowanie pól zbioru  $F^1$  może być wykonane w sposób bardzo prosty. Niech  $(i)$  oznacza wartość pola  $e \in F^1$ , które jest oznaczone znacznikiem. Wartościom 0, 1, (1), \* pól można przypisać następujące dwupozycyjne liczby dwójkowe:

0 - 00  
 1 - 01  
 (1) - 10  
 \* - 11

Ponieważ zapamiętanie trzech wartości pól wymaga dwóch bitów pamięci, więc znacznikowanie pól w sposób podany wyżej może się odbyć bez konieczności zwiększenia objętości zajmowanej pamięci w maszynie cyfrowej.

Inny przykład realizacji algorytmu  $A^q$  podano w ramach przykładu 4.2.

### 3.7.3. Zastosowanie algorytmów $A^m$ i $A^q$ w przypadku ogólnie postawionego problemu pokrycia

Załóżmy, że w tablicy logicznej nie wyróżnia się osi zmiennych  $x_1, x_2, \dots, x_n$ , lecz traktuje się ją jako zbiór  $2^n$  pól  $e_i$ , którym przyporządkowano numery  $\gamma(e_i)$ . Liczba  $n$  jest tu pewną określoną liczbą naturalną.

Niech dane jest odwzorowanie  $f$  zbioru  $N = \{0, 1, 2, \dots, 2^n - 1\}$  w zbiór  $\{0, 1, *\}$ :

$$f : N \rightarrow \{0, 1, *\}$$

Przypiszmy każdemu polu  $e_i$  tablicy wartość  $f(x(e_i))$ . Zbiór wszystkich pól tablicy wraz z przypisanymi w powyższy sposób wartościami ze zbioru  $\{0, 1, *\}$  nazywa się obrazem  $T(f)$  odwzorowania  $f$ . Symbolami  $F^1$ ,  $F^*$ ,  $F^0$  oznacza się jak poprzednio zbiory pól o wartościach odpowiednio 1, \*, 0.

Ogólne sformułowanie problemu pokrycia polega na odniesieniu pojęcia pokrycia (definicja 3.1) do obrazu odwzorowania  $T(f)$ . Pojęcie zespołu ulega tutaj zmianie.

Dana jest funkcja  $\varphi(E_1)$  odwzorowująca rodzinę podzbiorów pól  $E_1$  spełniających warunki:  $E_1 \subseteq F^1 \cup F^*$ ,  $E_1 \cap F^1 = \emptyset$ , w zbiór  $\{1, 0\}$ . Zbiory  $E_1$  takie, że  $\varphi(E_1) = 1$ , nazywa się zespołami w obrazie  $T(f)$ . Funkcja  $\varphi$  jest określona i niezmienna dla każdego konkretnego sformułowania problemu pokrycia. Zespoły interpretuje się jako zbiory pól, które w zależności od sformułowania odpowiadają na przykład funkcji logicznej realizowalnej na jednym funktorze (a więc koniunkcji - przy syntezie d.n.w., funkcji progowej - przy syntezie wyrażeń dysjunkcyjno-progowych itd.), zbiorom funktorów mieszczących się na jednej płytce - przy sformułowaniu odnoszącym się do optymalnego rozdziału funktorów na płytki itp.

Każdemu zespołowi  $E_1$  można przypisać pewną liczbę naturalną, nazywaną złożonością  $z(E_1)$ , która wszędzie jako składnik przy określaniu złożoności pokrycia zawierającego ten zespół (definicja 3.2). Złożoność może na przykład wyrażać liczbę wejść na funktor realizujący funkcję logiczną odpowiadającą temu zespołowi. Tak więc pojęcie pokrycia minimalnego  $M(T)$  może pozostać bez zmiany (definicja 3.3).

W algorytmach  $A^m$  i  $A^q$  jedyną operacją, która zależy od konkretnego sformułowania problemu pokrycia, jest operacja generacji gwiazdy  $G(e)$  pól  $e \in F^1$ . Algorytmy te mogą więc być zastosowane do rozwiązania dowolnego problemu pokrycia, jeśli tylko jest znany algorytm generacji gwiazd  $G(e)$ . Adaptacja algorytmów  $A^m$  i  $A^q$  do konkretnego sformułowania problemu polega zatem na znalezieniu odpowiedniego algorytmu generacji gwiazd. Znając interpretację po-

jęcia zespołu dla danego sformułowania taki algorytm można zawsze skonstruować.

### 3.7.4. Charakterystyka algorytmu $A^Q$

Obecnie omówimy krótko najistotniejsze własności algorytmu  $A^Q$ .

1. Algorytm  $A^Q$  dostarcza przybliżone rozwiązanie problemu pokrycia wraz z oszacowaniem maksymalnej różnicy między otrzymywanym pokryciem  $M^Q(T)$  a pokryciem  $M(T)$ , wyrażanej liczbą elementów  $\Delta$  i złożonością  $\delta$ .

2. Ponieważ pokrycie  $M^Q(T)$  składa się z quasi-ekstremal, które są lokalnie optymalnymi zespołami wybranymi z poszczególnych gwiazd, więc można się spodziewać, że pokrycie to będzie z reguły zbliżone do pokrycia minimalnego  $M(T)$ .

3. Posiadanie oszacowania w postaci parametrów  $\Delta$  i  $\delta$  umożliwia stwierdzenie, kiedy pokrycie quasi-minimalne  $M^Q(T)$  pokrywa się z pokryciem minimalnym  $M(T)$  (gdy  $\Delta = 0$ ,  $\delta = 0$ ), co umożliwia poszukiwanie pokrycia minimalnego przez powtarzanie algorytmu  $A^Q$ .

4. Realizacja algorytmu  $A^Q$  nie wymaga generowania wszystkich zespołów maksymalnych w obrazie  $T(f)$ , generuje się tylko zespoły należące do gwiazd rozłącznych i tylko do niektórych gwiazd zależnych (ten ostatni przypadek zachodzi wówczas, gdy realizuje się część II algorytmu). Po każdym kroku generacji gwiazdy rozłącznej pamięta się tylko wybraną quasi-ekstremalę oraz znacznikuje pola zbioru  $F^1$  pokrywane przez pozostałe zespoły tej gwiazdy. Znacznikowanie takie, jak pokazano przy omawianiu przykładu 3.4, nie wymaga zwiększenia pojemności pamięci maszyny cyfrowej realizującej algorytm.

5. Stosowalność algorytmu  $A^Q$  jest ograniczona przede wszystkim możliwościami zapamiętania obrazu  $T(f)$  oraz generowania gwiazd  $G(e)$ . Przy realizacji maszynowej algorytm  $A^Q$  może być stosowany przy dużej liczbie zmiennych.

Warto dodać, że algorytm  $A^Q$  można zmodyfikować w ten sposób, by realizować go nie na podstawie obrazu  $T(f)$ , lecz wyłącznie na podstawie zbiorów  $\Omega^1$  i  $\Omega^0$ . Oznacza to, że można go realizować bez konieczności pamiętania zbioru  $\Omega^*$ , który w przypadkach funkcji dużej liczby zmiennych posiada dużą moc (w obrazie  $T(f)$ ).



zbiór ten pamiętany jest jako wartość początkowa zmiennej  $F^*$ ). Modyfikacja powyższa opiera się na zastosowaniu algorytmu  $G^2$  przy generacji gwiazd.

Przy realizacji ręcznej algorytm  $A^q$  może być wykonany bez większych trudności dla dowolnej funkcji do 8 zmiennych. Autor znalazł za pomocą algorytmu  $A^q$  pokrycie minimalne  $M(T)$  obrazu funkcji 8 zmiennych  $f = S_{0,1,3,4,5,7,8}^8$ , posiadającej maksymalną liczbę prostych implikantów, jaką może posiadać funkcja 8 zmiennych, to jest 576 (zob. pkt 5.1.3 oraz Dodatek). Przy rozwiązywaniu tego przykładu metodą Quine'a-McCluskeya otrzymuje się cykliczną tablicę prostych implikantów o rozmiarach  $560 \times 182$ .

Ewentualne ulepszenie algorytmu  $A^q$  wymaga wyjaśnienia niektórych nie podjętych w pracy zagadnień. Do zagadnień tych należy: określenie własności obrazu  $T(f)$ , które mają wpływ na większą lub mniejszą wartość parametrów  $\Delta$  i  $\delta$ , zbadanie rozkładów mocy rodzin gwiazd rozłącznych otrzymywanych w wyniku różnego porządku wyboru pól przy realizacji algorytmu oraz zbadanie warunków istnienia rodziny gwiazd rozłącznych posiadającej moc równą mocy pokrycia minimalnego.

#### 4. ROZPOZNAWANIE SYMETRII FUNKCJI LOGICZNYCH

##### 4.1. Uwagi wstępne

W niniejszym rozdziale opisano metodę rozpoznawania symetrii funkcji logicznych polegającą na zastosowaniu pojęcia obrazu funkcji. Metoda ta umożliwi rozpoznawanie całkowitej lub częściowej symetrii funkcji względem liter zarówno w przypadku, gdy funkcja jest określona całkowicie, jak i w przypadku, gdy jest określona częściowo. Podano również sposób zastosowania tej metody wówczas, gdy funkcja jest wielowyjściowa.

Zanim przejdziemy do opisu metody, wprowadzimy pojęcie relacji

$\sim$ . Jeśli  $X = \left\{ \begin{matrix} \delta_k \\ x_k \end{matrix}, \begin{matrix} \delta_l \\ x_l \end{matrix} \right\}$  jest zbiorem symetrii funkcji  $f$ , to zgodnie z definicją 1.4 - literę  $x_k^{\delta_k}$  można wymienić z literą  $x_l^{\delta_l}$  i odwrotnie, bez zmiany funkcji  $f$ , jeśli funkcja  $f$  jest całkowicie określona lub, jeśli funkcja  $f$  jest częściowo określona,

bez zmiany pewnych funkcji  $f_1^{\delta_1}$  w zbiorze  $\Phi(f)$ . Zbiór tych ostatnich funkcji będziemy oznaczać symbolem  $\Phi_{k,1}$ . Relację mówiącą, że litera  $x_k^{\delta_k}$  jest wymienna z literą  $x_1^{\delta_1}$  bez zmiany funkcji  $f$ , zapisuje się jako  $(x_k^{\delta_k} \sim x_1^{\delta_1})_f$ . Jeśli funkcja  $f$  jest częściowo określona, to przez zapis  $(x_k^{\delta_k} \sim x_1^{\delta_1})_f$  rozumie się, że zbiór  $\Phi_{k,1}$  nie jest pusty. Jeśli zbiór  $\Phi_{k,1}$  zostanie konkretnie określony, to pisze się  $(x_k^{\delta_k} \sim x_1^{\delta_1}) \Phi_{k,1}$ . Można łatwo wykazać, że relacja  $\sim$  jest zwrotna, symetryczna i przechodnia, przy czym w przypadku, gdy  $\Omega \neq \emptyset$ , przechodniość zachodzi w sensie:

$$(x_k \sim x_1) \Phi_{k,1} \wedge (x_1 \sim x_m) \Phi_{1,m} \implies (x_k^{\delta_k} \sim x_m^{\delta_m}) \Phi_{k,1} \cap \Phi_{1,m} \quad (4.1)$$

Jeśli zbiorem symetrii funkcji  $f$  jest zbiór  $X = \{x_1^{\delta_1}, \dots, x_m^{\delta_m}\}$ , to relację wymienności liter w dowolnej parze liter zbioru  $X$  zapisuje się jako  $(x_1^{\delta_1} \sim x_2^{\delta_2} \sim \dots \sim x_m^{\delta_m})_f$ .

## 4.2. Rozpoznawanie symetrii funkcji całkowicie określonych

### 4.2.1. Rozpoznawanie całkowitej symetrii funkcji

Dana funkcja  $f(x_1, \dots, x_n)$  jest całkowicie symetryczna, jeśli jej zbiór symetrii o największej mocy, to znaczy zbiór  $X^m$ , posiada moc równą  $n$ . Sprawdzenie całkowitej symetrii opiera się na następującym twierdzeniu:

Twierdzenie 4.1. Warunkiem koniecznym i dostatecznym, by dana funkcja  $f(x_1, \dots, x_n)$  była funkcją całkowicie symetryczną, jest istnienie ciągu  $\delta = (\delta_2, \delta_3, \dots, \delta_n)$ ,  $\delta_1 \in \{0,1\}$ , takiego że:

$$\forall i \in \{2, \dots, n\}, (x_1 \sim x_1^{\delta_i})_f \quad (4.2)$$

Dowód. D o s t a t e c z n o ś ć. Wynika bezpośrednio z faktu, że relacja jest przechodnia.

K o n i e c z n o ś ć. Zgodnie z twierdzeniem 1.2, jeśli zbiór  $\{x_1^{\delta_1}, x_2^{\delta_2}, \dots, x_n^{\delta_n}\}$  jest zbiorem symetrii, to zbiór  $\{x_1^{1-\delta_2}, \dots, x_n^{1-\delta_n}\}$

jest również zbiorem symetrii funkcji  $f$ . Dlatego, jeśli funkcja jest całkowicie symetryczna, to musi istnieć zbiór symetrii zawierający literę  $x_1$ . Załóżmy, że funkcja  $f$  jest całkowicie symetryczna i że zbiór  $X = \{x_1, x_2^{\delta_2}, \dots, x_n^{\delta_n}\}$  jest jej zbiorem symetrii. Wówczas ciąg  $(\delta_2, \dots, \delta_n)$  jest ciągiem o pożądanych własnościach. Q.E.D.

Łatwo zauważyć, że twierdzenie 4.1 jest słuszne również przy dowolnej permutacji indeksów  $i \in \{1, \dots, n\}$  liter  $x_1^{\delta_i}$ . Z twierdzenia tego wynika, że sprawdzenie całkowitej symetrii można sprowadzić do sprawdzenia symetrii względem określonych par liter. Minimalna ilość par liter do sprawdzenia wynosi  $n - 1$ , maksymalna zaś  $2(n - 1)$ .

Rozwińmy funkcję  $f(x_1, \dots, x_n)$  względem zmiennych  $x_k$  i  $x_l$ ;  $k, l \in \{1, \dots, n\}$ ,  $k \neq l$ :

$$f(x_1, \dots, x_n) = \bar{x}_k \bar{x}_l f_{k,l}^0 \vee \bar{x}_k x_l f_{k,l}^1 \vee x_k \bar{x}_l f_{k,l}^2 \vee x_k x_l f_{k,l}^3 \quad (4.3)$$

gdzie

$$f_{k,l}^i = f(x_1, \dots, x_{k-1}, \omega_k^i, x_{k+1}, \dots, x_{l-1}, \omega_l^i, x_{l+1}, \dots, x_n),$$

$$i = 0, 1, 2, 3, \quad \omega_k^i = 2\omega_k^i + \omega_l^i.$$

#### Twierdzenie 4.2:

A. Relacja  $(x_k^{\delta_k} \sim x_l^{\delta_l})_f$ , gdzie  $\delta_k = \delta_l$ ,  $\delta_k, \delta_l \in \{0, 1\}$ , zachodzi wtedy i tylko wtedy, kiedy

$$f_{k,l}^1 = f_{k,l}^2 \quad (a)$$

B. Relacja  $(x_k^{\delta_k} \sim x_l^{\delta_l})_f$ , gdzie  $\delta_k \neq \delta_l$ ,  $\delta_k, \delta_l \in \{0, 1\}$ , zachodzi wtedy i tylko wtedy, kiedy

$$f_{k,l}^0 = f_{k,l}^3 \quad (b)$$

#### Dowód:

A. Jeśli warunek (a) jest spełniony, to przekształcając wyrażenie (4.3) otrzymuje się:

$$f(x_1, \dots, x_n) = \bar{x}_k \bar{x}_1 f_{k,1}^0 \vee (\bar{x}_k x_1 \vee x_k \bar{x}_1) f_{k,1}^1 \vee x_k x_1 f_{k,1}^3$$

Z otrzymanego wyrażenia wynika, że permutacje  $\begin{pmatrix} x_k & x_1 \\ x_1 & x_k \end{pmatrix}$  i  $\begin{pmatrix} \bar{x}_k & \bar{x}_1 \\ \bar{x}_1 & \bar{x}_k \end{pmatrix}$

nie zmieniają funkcji  $f$ , co jest równoważne relacji  $(x_k^k \sim x_1^1)_f$ , gdy  $\delta_k = \delta_1$ . Jeśli warunek (a) nie jest spełniony, to powyższe permutacje zmieniają funkcję  $f$ , co dowodzi konieczność warunku (a).

B. Część B dowodzi się analogicznie.

Q.E.D.

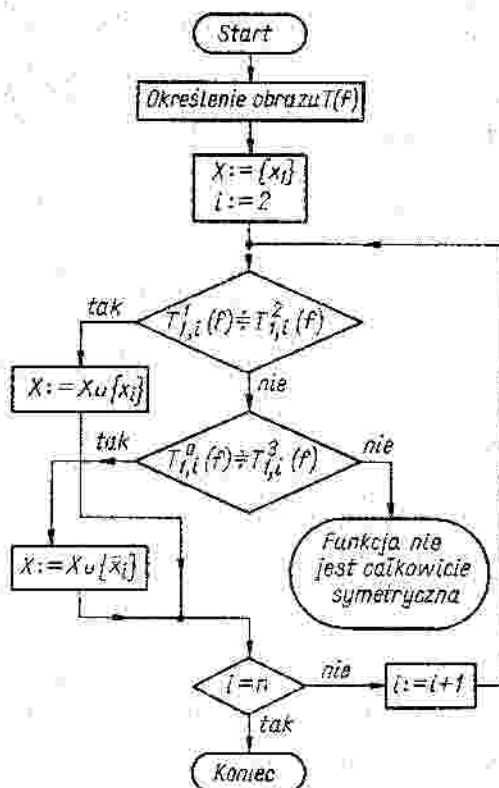
Warunki (a) i (b) twierdzenia 4.2 mogą być łatwo sprawdzone na podstawie obrazu  $T(f)$ . Obrazy  $T(f_{k,1}^i)$ ,  $i = 0, 1, 2, 3$ , są określone przez podzbiory obrazu  $T(f)$  składające się z pól zbiorów  $E^i = E(x_k^k, x_1^1)$ , gdzie  $2\delta_k^i + \delta_1^i = 1$ . Takie podzbiory obrazu  $T(f)$  oznacza się  $T_{k,1}^i(f)$ . Warunki (a) i (b) są równoważne relacjom  $T_{k,1}^1(f) \doteq T_{k,1}^2(f)$  lub  $T_{k,1}^0(f) \doteq T_{k,1}^3(f)$ , gdzie przez relację  $\doteq$  rozumie się, że odpowiadające<sup>1)</sup> sobie pola w obrazach  $T_{k,1}^1(f)$  i  $T_{k,1}^2(f)$  lub  $T_{k,1}^0(f)$  i  $T_{k,1}^3(f)$  posiadają jednakowe wartości. Jeśli warunek (a) jest spełniony, to mówi się, że zachodzi symetria pierwszego rodzaju, a jeśli jest spełniony warunek (b), mówi się, że zachodzi symetria drugiego rodzaju względem pary zmiennych  $\{x_k, x_1\}$ .

Na rys. 4.1 przedstawiono algorytm S sprawdzenia, czy funkcja  $f$  jest całkowicie symetryczna; podstawą jego są twierdzenia 4.1 i 4.2. Symbolem  $X$  oznaczono zmienną, której wartość po zakończeniu algorytmu jest zbiorem symetrii  $X^m$  o mocy  $c(X^m) = n$ , jeśli funkcja jest całkowicie symetryczna.

W ogólnym przypadku funkcja  $f$  może posiadać szereg alternatywnych zbiorów symetrii  $X_j^m$ ,  $j = 1, 2, \dots$ , różniących się tym, że poszczególne zmienne występują w nich albo w postaci  $\bar{x}_j$ , albo w postaci  $x_j$ . Aby określić wszystkie zbiory  $X_j^m$ , należy sprawdzić dla każdej pary zmiennych  $\{x_i, x_1\}$ ,  $i = 2, 3, \dots, n$ , zarów-

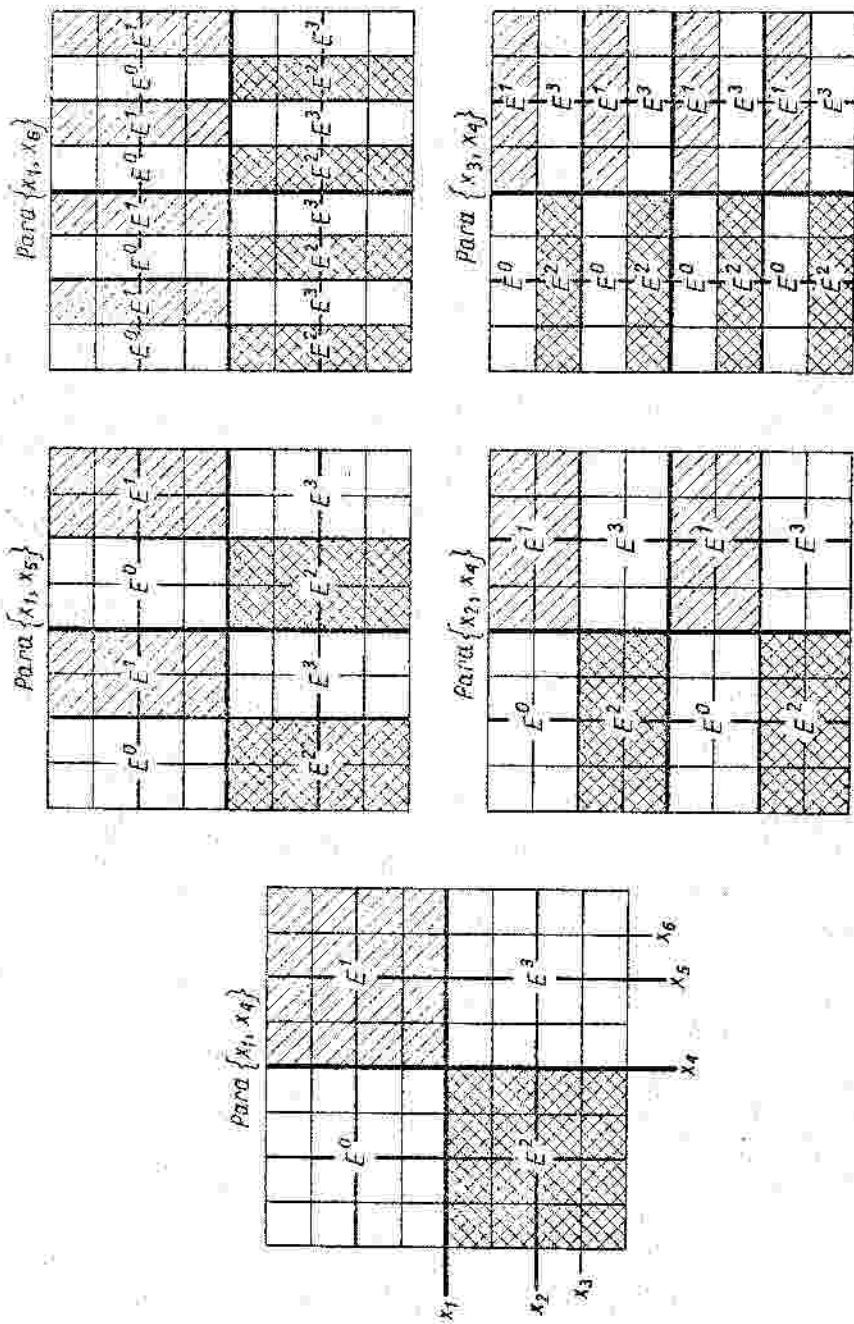
1) Przez odpowiadające sobie pola w  $T_{k,1}^{i_1}(f)$  i  $T_{k,1}^{i_2}(f)$ , gdzie odpowiednio  $i_1 = 1, 0$ ,  $i_2 = 2, 3$ , rozumie się odpowiadające sobie pola w zespołach  $E^{i_1}$  i  $E^{i_2}$  (definicja 2.12).

no warunek (a) jak i warunek (b). Znając wszystkie pary liter symetrii  $\{x_1, x_{i-1}^{\sigma_i}\}$ ,  $i = 2, 3, \dots$ , dzięki przechodniości relacji można łatwo określić zbiory  $X_j^{\sigma_i}$ .



Rys. 4.1. Algorytm S

W przypadku ręcznej realizacji algorytmu wygodniej jest przyjąć inny niż w algorytmie S porządek sprawdzania symetrii względem par zmiennych, a mianowicie: sprawdzać symetrię kolejno dla par:  $\{x_1, x_{\lfloor n/2 \rfloor + 1}\}$ ,  $\{x_1, x_{\lfloor n/2 \rfloor + 2}\}$ , ...,  $\{x_1, x_n\}$ ,  $\{x_2, x_{\lfloor n/2 \rfloor + 1}\}$ ,  $\{x_3, x_{\lfloor n/2 \rfloor + 1}\}$ , ...,  $\{x_{\lfloor n/2 \rfloor}, x_{\lfloor n/2 \rfloor + 1}\}$ . W tym przypadku zbiory pól, których wartości porównuje się dla sprawdzenia symetrii, są usytuowane w tablicy logicznej w pewien charakterystyczny i łatwy do określenia sposób. Na rys. 4.2 zaznaczone te zbiory pól w przypadku sprawdzania całkowitej symetrii funkcji 6 zmiennych.



$$E^0 = E(\bar{x}_1, \bar{x}_4) \vee E^1 = E(\bar{x}_1, x_4), \quad E^2 = E(x_1, \bar{x}_4), \quad E^3 = E(x_1, x_4)$$

rys. 4.2. Zbiory pól, których wartości porównuje się przy sprawdzaniu całkowitej symetrii funkcji logicznej 6 zmiennych

Na podstawie określonych par liter symetrii określa się zbiór  $X^m$ .

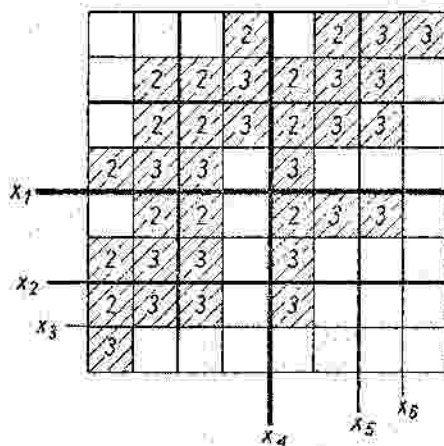
#### 4.2.2. Określanie zbioru $\alpha$ -indeksów funkcji całkowicie symetrycznej.

Jeśli funkcja  $f$  jest całkowicie symetryczna, to może być przedstawiona w postaci  $S_A(X)$ , gdzie  $X$  jest zbiorem symetrii,  $A$  - zbiorem  $\alpha$ -indeksów. Stwierdziwszy za pomocą algorytmu  $S$ , że funkcja  $f$  jest całkowicie symetryczna, mamy również określony zbiór  $X$ . Aby określić funkcję  $S_A(X) = f$ , należy więc znaleźć zbiór  $A$ . Do tego pomocne jest

Twierdzenie 4.3. Jeśli  $f = S_A^n$ , to zbiór  $F^1 \in T(f)$  jest zbiorem wszystkich pól  $e$ , których wagi  $\delta(e) \in A$ .

Dowód. Niech  $A = \{a_k\}_{k=1, \dots, s}$ . Jeśli  $f = S_A^n$ , to  $f = \bigvee_{k=1}^s S_{a_k}^n$ . Funkcja  $S_{a_k}^n$  posiada  $C_n^{a_k}$  składników, z których każdy zawiera  $a_k$  liter typu  $x_i$  oraz  $n - a_k$  liter typu  $\bar{x}_i$ . Każdemu składnikowi odpowiada ciąg  $\Omega_j$  zawierający  $a_k$  jedynek i  $n - a_k$  zer, zatem - zgodnie z definicją 2.5 - wagi pól odpowiadające tym ciągom są równe  $\delta(e) = a_k$ . Ponieważ funkcja  $f$  jest dysjunkcją funkcji  $S_{a_k}$ , więc zbiór  $F^1$  składa się ze wszystkich pól  $e$ , których wagi  $\delta(e) \in A$ . Q.E.D.

Rys. 4.3.  
Obraz funkcji  $S_{2,3}^6$



Dla zilustrowania twierdzenia 4.3 na rys. 4.3 przedstawiono ob-

raz funkcji  $S_{2,3}^6$  (pola zbioru  $F^1$  zakreskowano).

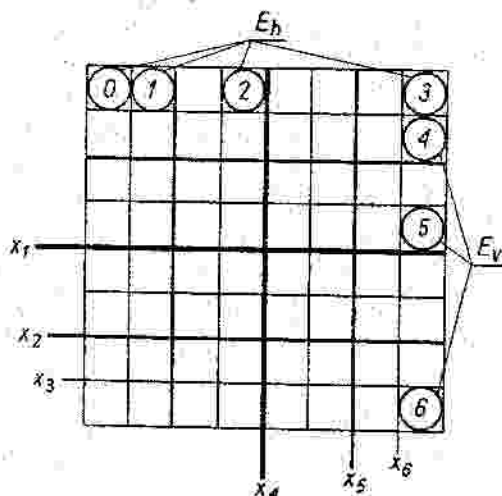
Dzięki powyższemu twierdzeniu określenie zbioru  $A$ , w przypadku gdy zbiór symetrii  $X_m = \{x_1, \dots, x_n\}$ , można sprowadzić do wykonania następujących czynności:

1) określenia pewnego minimalnego zbioru pól  $E = \{e_i\}_{i=1}^n$  takiego, że wagi  $\delta(e_i) = i$ , to jest wyczerpują zbiór  $\{0, 1, \dots, n\}$  wszystkich możliwych wartości indeksów  $a_k$ ;

2) sprawdzenia, które pola  $e_i$  należą do zbioru  $F^1$ ; jeśli  $e_i \in F^1$ , to  $a_k = \delta(e_i) \in A$ .

Można łatwo wykazać, że  $E = E_h \cup E_v$ , gdzie  $E_h$  jest zbiorem pól tablicy logicznej o numerach  $\gamma(e) = 2^j - 1, j = 0, 1, \dots, n - \lfloor n/2 \rfloor$ , a  $E_v$  jest zbiorem pól o numerach  $\gamma(e) = 2^{n - \lfloor n/2 \rfloor + k} - 1, k = 1, 2, \dots, \lfloor n/2 \rfloor$ . Pola zbioru  $E_h$  mają wagi odpowiednio  $\delta(e) = 0, 1, \dots, n - \lfloor n/2 \rfloor$ , a pola zbioru  $E_v$  mają wagi odpowiednio  $\delta(e) = n - \lfloor n/2 \rfloor + 1, n - \lfloor n/2 \rfloor + 2, \dots, n$ . Zbiór  $E_h$  jest zawarty w zbiorze  $E(\bar{x}_1, \dots, \bar{x}_{\lfloor n/2 \rfloor})$ , to jest w zbiorze pól leżących w górnej skrajnej warstwie tablicy logicznej. Zbiór  $E_v$  jest zawarty w zbiorze  $E(x_{\lfloor n/2 \rfloor + 1}, \dots, x_n)$ , to jest w zbiorze pól leżących w prawej skrajnej kolumnie tablicy.

Na rys. 4.4. zaznaczono zbiór  $E = E_h \cup E_v$  w tablicy logicznej 6 zmiennych.



Rys. 4.4. Zbiór pól  $E = E_h \cup E_v$  w tablicy logicznej 6 zmiennych

Załóżmy, że zbiór symetrii  $X = \{x_i\}_{i=1}^n$ , gdzie  $s_i$  - ustalona liczba ze zbioru  $\{0, 1\}$ , zawiera przynajmniej jedną literę  $x_i^0$ ,

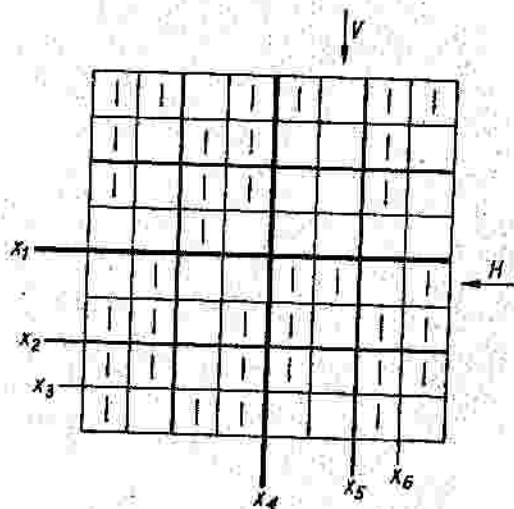


to znaczy literę zanegowaną  $\bar{x}_i$ . Aby obecny przypadek sprowadzić do poprzedniego, można by skonstruować obraz  $T'(f)$  składający się z pół tablicy logicznej o osiach  $x'_i$ ,  $i = 1, \dots, n$ , gdzie  $x'_i = x_i^{6_1}$ . Obraz  $T(f)$  można otrzymać z obrazu  $T'(f)$  przez kolejne przesunięcia dla każdego "i", takiego, że  $6_1 = 0$ , zbioru pół, któremu podporządkowana jest litera  $x'_i$ , na miejsce zbioru pół, któremu przyporządkowana jest litera  $\bar{x}_i$  i odwrotnie.

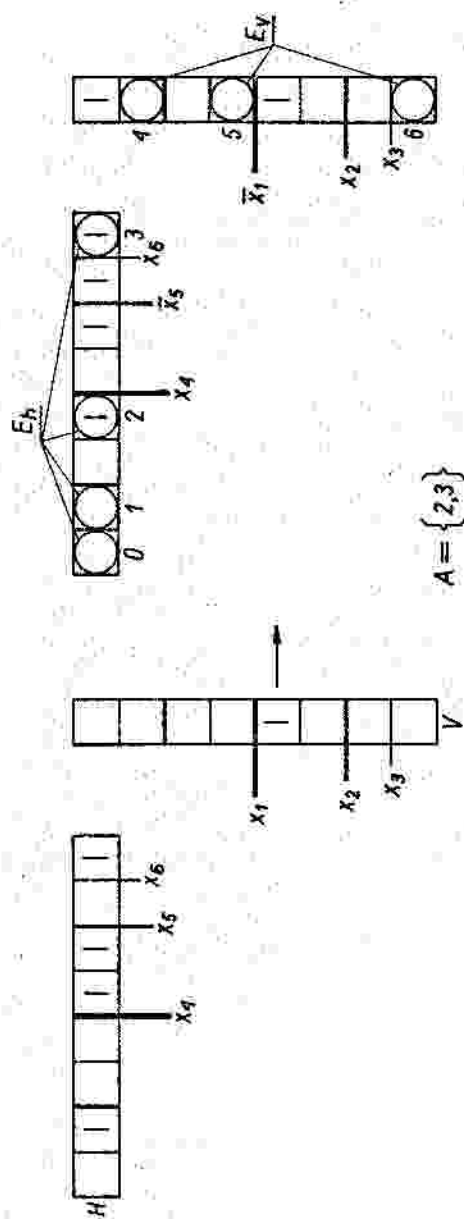
Dla określenia zbioru  $A$  należy zbadać wartości pół zbioru  $E$  leżących w górnej skrajnej warstwie i prawej skrajnej kolumnie obrazu  $T(f)$ . Dlatego wystarczy powyższą transformację wykonać wewnątrz zbioru  $H = E(x_1^{1-6_1}, \dots, x_{\lfloor n/2 \rfloor}^{1-6_{\lfloor n/2 \rfloor}})$  dla zanegowanych liter  $\bar{x}_i \in X$ , gdzie  $\lfloor n/2 \rfloor + 1 \leq i \leq n$  oraz wewnątrz zbioru  $V = E(x_p^{6_p}, \dots, x_n^{6_n})$ ,  $p = \lfloor n/2 \rfloor + 1$ , dla zanegowanych liter  $\bar{x}_i \in X$ , gdzie  $1 \leq i \leq \lfloor n/2 \rfloor$ .

Zbiory tak otrzymane będą leżały w górnej warstwie i w prawej kolumnie obrazu  $T'(f)$ . Ze względów praktycznych z dwóch równoważnych zbiorów liter symetrii:  $X$  oraz  $X' = \{x_1^{1-6_1}, \dots, x_n^{1-6_n}\}$ , wygodniej jest wybrać ten, który zawiera mniej liter typu  $\bar{x}_i$ .

**Przykład 4.1.** Sprawdzić, czy funkcja  $f(x_1, \dots, x_6)$ , której obraz  $T(f)$  przedstawiono na rys. 4.5, jest całkowicie symetryczna i jeśli tak, to określić funkcję  $S_A(X) = f$ .



Rys. 4.5. Obraz  $T(f)$  dla przykładu 4.1.

Rys. 4.6. Transformacja zbiorów  $H$  i  $V$

Wykonujemy algorytm S przy uwzględnieniu modyfikacji wynikłej z przyjęcia porządku sprawdzenia symetrii względem par zmiennych (wygodnego przy jego ręcznej realizacji), to jest porównujemy wartości pól zbiorów przedstawionych na rys. 4.2.

$$1. (a) T_{1,4}^1(f) \doteq T_{1,4}^2(f) - \text{nie}$$

$$(b) T_{1,4}^0(f) \doteq T_{1,4}^3(f) - \text{tak, } X := \{x_1, \bar{x}_4\}$$

⋮  
⋮  
⋮

$$5. (a) T_{3,4}^1(f) \doteq T_{3,4}^2(f) - \text{tak, } X := \{x_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, x_5, \bar{x}_6\}$$

Równoważny zbiorowi symetrii X jest zbiór

$$X' = \{\bar{x}_1, x_2, x_3, x_4, \bar{x}_5, x_6\},$$

który wybieramy ze względu na mniejszą ilość liter  $\bar{x}_1$ . Określamy zbiory  $H = E(x_1, \bar{x}_2, \bar{x}_3)$  oraz  $V = E(x_4, \bar{x}_5, x_6)$  (zob. rys.4.5).

Po dokonaniu transformacji zbiorów H i V, jak pokazano na rys. 4.6, otrzymuje się  $A = \{2, 3\}$ . Tak więc  $F = S_{2,3}(X') = S_{3,4}(X)$ .

#### 4.2.3. Rozpoznawanie częściowej symetrii

Funkcja  $f(x_1, \dots, x_n)$  jest funkcją częściowo symetryczną, jeśli  $1 < c(X^n) < n$ . Określimy maksymalną liczbę  $s(n)$  różnych możliwych zbiorów symetrii przy danej liczbie zmiennych  $n$ .

Każdemu zbiorowi symetrii X można przyporządkować ciąg  $A = (a_1, \dots, a_n)$  gdzie

$$a_i = \begin{cases} 0, & \text{jeśli } \bar{x}_i \in X \\ 1, & \text{jeśli } x_i \in X \\ 2, & \text{jeśli } X \text{ nie zawiera ani } \bar{x}_i, \text{ ani } x_i \end{cases}$$

$i = 1, \dots, n$

Różnych ciągów A może być  $3^n$ . Ponieważ zbiory X są co najmniej 2-elementowe, więc

$$s(n) = 3^n - (2n + 1) \quad (4.4)$$

W zbiorze wszystkich  $s(n)$  zbiorów  $X$  można wyróżnić  $s^{\circ}(n) = 2(2^n - n - 1)$  podzbiorów, w których wszystkie litery są albo proste, albo zanegowane. Na rys. 4.7 podano tabelkę wartości  $s(n)$  i  $s^{\circ}(n)$  dla  $n = 2, \dots, 12$ .

n	$s^{\circ}(n)$	$s(n)$
2	2	4
3	8	20
4	22	72
5	52	232
6	114	716
7	240	1172
8	494	6544
9	1004	19664
10	2026	60050
11	4072	180450
12	8166	541350

Rys. 4.7. Tabelka wartości  $s^{\circ}(n)$  i  $s(n)$  dla  $n = 2, \dots, 12$

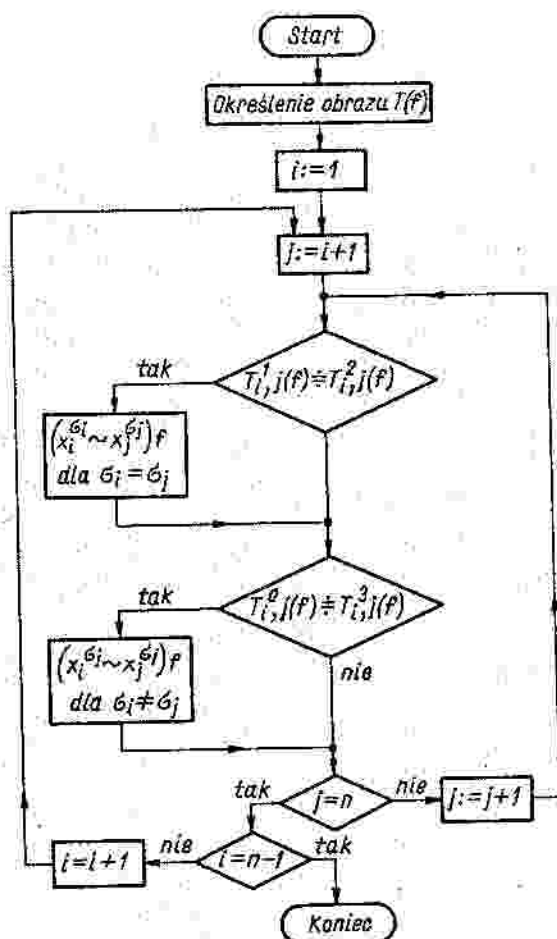
Wartości  $s(n)$  i  $s^{\circ}(n)$  dla  $n > 12$  można łatwo obliczyć według wzorów:

$$s(n + 1) = 3(s(n)) + 4n \quad (4.5)$$

$$s^{\circ}(n + 1) = 2(s^{\circ}(n)) + 2n \quad (4.6)$$

Jak widać z powyższej tabelki, udział zbiorów złożonych wyłącznie z liter prostych lub zanegowanych w zbiorze wszystkich zbiorów symetrii maleje szybko wraz z  $n$  i przy  $n = 12$  wynosi już tylko ok. 1,5 %. Dominują natomiast zbiory liter "mieszanych". Wynika z tego, że rozpoznawanie symetrii względem liter stanowi przypadek znacznie ogólniejszy niż rozpoznawanie symetrii względem zmiennych (to jest liter prostych), którego dotyczy na przykład metoda Dietmeyera i Schneidera /100/.

Liczbę zbiorów symetrii równą  $s(n)$  mają funkcje całkowicie symetryczne, które dla każdej pary zmiennych mają symetrię zarówno



Rys. 4.8. Algorytm  $S^c$

pierwszego jak i drugiego rodzaju. Takimi funkcjami są na przykład funkcje o postaci ogólnej  $S_{0,2,4,\dots,k_1}^n$  lub  $S_{1,3,5,\dots,k_2}^n$  gdzie

$$k_1 = \begin{cases} n & , \text{jeśli } n \text{ parzysta} \\ n - 1 & , \text{jeśli } n \text{ nieparzysta} \end{cases}$$

$$k_2 = \begin{cases} n & , \text{jeśli } n \text{ nieparzysta} \\ n - 1 & , \text{jeśli } n \text{ parzysta} \end{cases}$$

Należy jednak zaznaczyć, że chodzi tu o wszystkie możliwe zbiory symetrii, nie zaś tylko o zbiory maksymalne. Tych ostatnich wyżej podane funkcje mają tylko  $2^n$ .

Do określenia wszystkich maksymalnych  $X_j^m$ ,  $j = 1, 2, \dots$ , zbiorów częściowej symetrii danej funkcji  $f$  wystarcza znajomość wszystkich par liter symetrii. Pomocna tu jest następująca implikacja:

$$(\tilde{X}_a \sim x_k^{6k})_f \wedge (\tilde{X}_b \sim x_k^{6k}) \implies (X_a \sim X_b \sim x_k^{6k})_f \quad (4.7)$$

gdzie:  $\tilde{X}_a$  - litera  $x_a^{6a}$  lub ciąg liter połączonych znakiem  $\sim$ :  $x_{a_1}^{6a_1} \sim x_{a_2}^{6a_2} \sim \dots$ ,  $\tilde{X}_b$  - analogicznie.

Słuszność powyższej implikacji wynika z przechodniości relacji  $\sim$ . Na rys. 4.8 przedstawia się algorytm  $S^c$  dla określenia wszystkich par symetrii danej funkcji  $f$ .

Zo względu na przechodniość relacji  $\sim$ , odpowiednie operacje algorytmu  $S^c$  mogą być pominięte: jeśli  $(x_k^{6k} \sim x_1^{61})_f$  i  $(x_i^{6i} \sim x_m^{6m})_f$ , to również  $(x_k^{6k} \sim x_m^{6m})_f$  i sprawdzanie symetrii względem pary liter  $\{x_k^{6k}, x_m^{6m}\}$  jest zbędne. Gdy funkcja  $f$  jest całkowicie symetryczna, algorytm  $S^c$  równy jest algorytmowi  $S$ , zmodyfikowanemu w ten sposób, że dla każdej pary zmiennych sprowadza się zarówno warunek (a) jak i (b).

#### 4.2.4. Określenie wyrażeń algebraicznych funkcji częściowo symetrycznych.

Niech  $X = \{x_i^{6i}\}_{i \in I}$ ,  $I \subseteq \{1, \dots, n\}$  jest zbiorem symetrii danej funkcji  $f(x_1, \dots, x_n)$ . Załóżmy, bez straty na ogólności, że  $I = \{1, 2, \dots, m\}$ ,  $1 < m < n$ . Uogólniając wzór Shannona /39/ na przypadek symetrii względem liter funkcji  $f(x_1, \dots, x_n)$  można zapisać następująco:

$$f(x_1, \dots, x_n) = \bigvee_{j=0}^m S_j(X) f_j(x_{m+1}, x_{m+2}, \dots, x_n) \quad (4.8)$$

gdzie  $f_j(x_{m+1}, x_{m+2}, \dots, x_n) = f(\omega_1, \dots, \omega_m, x_{m+1}, x_{m+2}, \dots, x_n)$ ,

$$\omega_i = \begin{cases} 1, & \text{jeśli } x_i \in X \\ 0, & \text{jeśli } \bar{x}_i \in X \end{cases} \quad \text{dla } i = 1, 2, \dots, j, \quad j \neq 0$$

$$\omega_i = \begin{cases} 0, & \text{jeśli } x_i \in X \\ 1, & \text{jeśli } \bar{x}_i \in X \end{cases} \quad \text{dla } i = j+1, j+2, \dots, m \\ \text{oraz gdy } j = 0$$

Jeśli maksymalne zbiory symetrii  $X_i^m$ ,  $i = 1, 2, \dots$ , funkcji  $f$  są określone, to dla każdego zbioru można utworzyć wyrażenie (4.8). Funkcje  $f_j$  można określić na podstawie dowolnego wyrażenia algebraicznego funkcji  $f$ . W przypadku, gdy funkcja  $f$  jest funkcją całkowicie symetryczną, wyrażenie (4.8) redukuje się do postaci

$$\bigvee_{j=0}^n S_j(X) f_j = S_A(X).$$

#### 4.3. Rozpoznawanie symetrii funkcji częściowo określonych

Zagadnienie rozpoznawania symetrii funkcji częściowo określonych rozpatrzmy ogólnie, bez rozgraniczania przypadku, gdy funkcja jest całkowicie symetryczna, od przypadku, gdy funkcja jest symetryczna częściowo. Załóżmy, że  $f(x_1, \dots, x_n)$  jest funkcją częściowo określoną, a więc że  $\Omega^* \neq \emptyset$  (co równoważne jest temu, że  $F^* \neq \emptyset$ ).

Funkcja  $f$  jest symetryczna względem pary liter  $\left\{ \begin{matrix} \epsilon_k \\ x_k \end{matrix}, \begin{matrix} \epsilon_l \\ x_l \end{matrix} \right\}$  jeśli - zgodnie z definicją 1.4 - istnieje niepusty zbiór  $\bar{\Phi}_{k,l} \subseteq \bar{\Phi}(f)$ , taki, że dla każdej  $f_1^* \in \bar{\Phi}_{k,l}$  zachodzi relacja:

$$T_{k,l}^1(f_1^*) \doteq T_{k,l}^2(f_1^*), \quad \text{gdy } \epsilon_k = \epsilon_l \quad (a)$$

lub

$$T_{k,l}^0(f_1^*) \doteq T_{k,l}^3(f_1^*), \quad \text{gdy } \epsilon_k \neq \epsilon_l \quad (b)$$

Załóżmy dla ustalenia uwagi, że  $\epsilon_k = \epsilon_l$ . Niech  $\left\{ \begin{matrix} \epsilon_j^k \\ e_j^k \end{matrix}, \begin{matrix} \epsilon_j^l \\ e_j^l \end{matrix} \right\}$ ,  $j = 1, 2, \dots, 2^{n-2}$ , są parami odpowiadających sobie pól w obrazach odpowiednio  $T_{k,l}^1(f_1^*)$  i  $T_{k,l}^2(f_1^*)$ , gdzie  $f_1^* \in \bar{\Phi}_{k,l}$ .

Zbiór  $\bar{\Phi}_{k,l}$  jest równoważny (z pewnymi ograniczeniami - zob.

wiersz 3 tabelki poniżej) funkcji częściowo określonej  $f_{k,1}$ , której obraz  $T(f_{k,1})$  powstaje z obrazu  $T(f)$  przez zrealizowanie nad każdą parą  $\{e_j^k, e_j^1\}$ , w zależności od wartości pól tej pary w  $T(f)$ , następujących operacji:

	$e_j^k$	$e_j^1$	operacja
1	0	0	wartości pól pozostają bez zmiany
2	1	1	
3	*	*	
4	0	*	wartości * zmienia się na wartość 0 lub 1 tak, aby otrzymać równość wartości pól $e_j^k$ i $e_j^1$
5	*	0	
6	1	*	
7	*	1	
8	0	1	sprzeczność. $\bar{\Phi}_{k,1} = \emptyset$
9	1	0	

Z podanej tabelki wynika, że warunkiem koniecznym istnienia symetrii względem pary  $\{x_k^{\sigma_k}, x_1^{\sigma_1}\}$  jest warunek, aby pola żadnej pary  $\{e_j^k, e_j^1\}$ ,  $j = 1, 2, \dots, 2^{n-2}$  nie posiadały wartości odpowiednio 0 i 1 lub 1 i 0.

Dla przypadku  $\sigma_k \neq \sigma_1$  warunek symetrii jest analogiczny, ale odnośnie par odpowiadających sobie pól w  $T_{k,1}^0(f)$  i  $T_{k,1}^1(f)$ .

Z powyższych rozważań wynika, że dla danego obrazu  $T(f)$  oraz danych wartości  $\sigma_k$  i  $\sigma_1$  zbiór  $\bar{\Phi}_{k,1}$  może być jednoznacznie wyznaczony na podstawie uporządkowanej trójki  $(C_{k,1}^0, C_{k,1}^1, R_{k,1})$ , gdzie  $C_{k,1}^0$  - jest zbiorem pól zmieniających wartość \* (w  $T(f)$ ) na wartość 0 (w  $T(f_{k,1})$ ),  $C_{k,1}^1$  - zbiorem pól zmieniających wartość \* na wartość 1, a  $R_{k,1}$  - jest rodziną zbiorów  $\{e_j^k, e_j^1\}$  takich, że pola  $e_j^k$  i  $e_j^1$  posiadają w obrazie  $T(f)$  wartość \*. Pola każdego takiego zbioru muszą w obrazie  $T(f_1^*)$ ,  $f_1^* \in \bar{\Phi}_{k,1}$ , posiadać jednakowe wartości 0 lub 1.

Założmy, że  $(x_k^{\sigma_k} \sim x_1^{\sigma_1})_f$ ,  $(x_1^{\sigma_1} \sim x_m^{\sigma_m})_f$ , oraz że odpowiednie trójki  $(C_{k,1}^0, C_{k,1}^1, R_{k,1})$  i  $(C_{1,m}^0, C_{1,m}^1, R_{1,m})$  są określone.



Twierdzenie 4.4.

$$(x_k^{\sigma_k} \sim x_1^{\sigma_1})_f \wedge (x_1^{\sigma_1} \sim x_m^{\sigma_m})_f \implies (x_k^{\sigma_k} \sim x_1^{\sigma_1} \sim x_m^{\sigma_m})_f$$

wtedy i tylko wtedy, kiedy

$$C_{k,1}^0 \cap C_{1,m}^1 = \emptyset \quad (a)$$

$$C_{k,1}^1 \cap C_{1,m}^0 = \emptyset \quad (b)$$

$$\nexists (e^0, e^1) \in (C_{k,1}^0 \times C_{k,1}^1 \cup C_{1,m}^0 \times C_{1,m}^1) \exists (c_i^* \in R_{k,1} \cup R_{1,m}),$$

$$\{e^0, e^1\} \subseteq C_1^* \quad (c)$$

Dowód. Konieczność: Załóżmy, że  $(x_k^{\sigma_k} \sim x_1^{\sigma_1} \sim x_m^{\sigma_m})_f$ . W takim razie istnieje niepusty zbiór  $\bar{\Phi}_{k,1,m} \subseteq \bar{\Phi}(f)$  taki, że dla każdej funkcji  $f_i^* \in \bar{\Phi}_{k,1,m}$  zbiór  $\{x_k^{\sigma_k}, x_1^{\sigma_1}, x_m^{\sigma_m}\}$  jest zbiorem symetrii. Zgodnie ze wzorem (4.1)  $\bar{\Phi}_{k,1,m} = \bar{\Phi}_{k,1} \cup \bar{\Phi}_{1,m}$ . Jeśli (a) lub (b) nie zachodzi, to istnieje pole  $e$ , które w obrazie każdej funkcji  $f_i^* \in \bar{\Phi}_{k,1}$  ma przeciwną wartość niż w obrazie każdej funkcji  $f_i^* \in \bar{\Phi}_{1,m}$ . Z tego wynika, że  $\bar{\Phi}_{k,1} \cap \bar{\Phi}_{1,m} = \emptyset$ , a więc  $\bar{\Phi}_{k,1,m} = \emptyset$ , co zaprzecza poprzedniemu założeniu. Jeśli (c) nie zachodzi, to istnieje para pól, które dla każdej funkcji  $f_1^* \in \bar{\Phi}_{k,1}$  mają identyczne wartości, a dla każdej  $f_1 \in \bar{\Phi}_{1,m}$  wartości przeciwne lub odwrotnie. Z powyższego również wynika, że  $\bar{\Phi}_{k,1,m} = \emptyset$ .

Dostateczność. Dostateczności warunków (a), (b) i (c) dowodzi się przez konstrukcję zbioru  $\bar{\Phi}_{k,1,m}$ .

Niech  $C_{k,1,m}^0 = C_{k,1}^0 \cup C_{1,m}^0 \cup C_r^0$  i  $C_{k,1,m}^1 = C_{k,1}^1 \cup C_{1,m}^1 \cup C_r^1$  gdzie  $C_r^0$  jest zbiorem wszystkich pól  $e_1^0$  takich, że

$$\forall e_1^0 \exists e_2^0 \in C_{k,1}^0 \cup C_{1,m}^0, \{e_1^0, e_2^0\} \in R_{k,1} \cup R_{1,m}$$

$C_r^1$  jest zbiorem wszystkich pól  $e_1^1$  takich, że

$$\forall e_1^1 \exists e_2^1 \in C_{k,1}^1 \cup C_{1,m}^1, \{e_1^1, e_2^1\} \in R_{k,1} \cup R_{1,m}$$

Niech  $R_{k,1}^0, R_{k,1}^1 \subseteq R_{k,1}$  oznaczają zbiory tych elementów rodziny  $R_{k,1}$ , które zawierają pola odpowiednio  $e_1^0$  lub  $e_1^1$ . Niech  $R_{1,m}^0$  i  $R_{1,m}^1$  oznaczają analogiczne podzbiory rodziny  $R_{1,m}$ . Niech  $R_1 = R_{k,1} \setminus (R_{k,1}^0 \cup R_{k,1}^1)$  i  $R_2 = R_{1,m} \setminus (R_{1,m}^0 \cup R_{1,m}^1)$ . Określmy zbiór  $R_{k,1,m} = R_1 \sqcup R_2$ , gdzie  $\sqcup$  oznacza następującą operację:

- 1) określa się zbiór  $R_1 \cup R_2$ ;
- 2) z każdej pary elementów zbioru  $R_1 \cup R_2$ , których przecięcie jest niepuste, tworzy się jeden element będący ich sumą; następnie powyższą operację na otrzymanych w ten sposób zbiorach, powtarza się dopóty, dopóki nie powstanie rodzina, której dowolne dwa elementy mają puste przecięcie (w przypadku rozważanym obecnie ten proces kończy się po pierwszym kroku).

Jeśli zachodzi (a) i (b), to

$$(c_{k,1}^0 \cup c_{1,m}^0) \cap (c_{k,1}^1 \cup c_{1,m}^1) = \emptyset \quad (a')$$

Jeśli zachodzi (c), to

$$c_r^0 \cap (c_{k,1}^1 \cup c_{1,m}^1) = \emptyset \quad (b')$$

oraz

$$c_r^1 \cap (c_{k,1}^0 \cup c_{1,m}^0) = \emptyset \quad (c')$$

Ponieważ elementy rodziny  $R_{k,1} \cup R_{1,m}$  zawierające jakieś pola zbioru  $c_{k,1}^0 \cup c_{k,1}^1 \cup c_{1,m}^0 \cup c_{1,m}^1$  są rozłączne, więc

$$c_r^0 \cap c_r^1 = \emptyset \quad (d')$$

Z (a') - (d') wynika, że

$$c_{k,1,m}^0 \cap c_{k,1,m}^1 = \emptyset \quad (e')$$

Niech  $T(f_{k,1,m})$  oznacza obraz, który otrzymuje się przez zmianę w  $T(f)$  wartości pól zbioru  $c_{k,1,m}^0$  z \* na 0 oraz pól  $c_{k,1,m}^1$  z \* na 1. Niech  $\{f_i^*\}_{i \in I_{k,1,m}}$ ,  $I_{k,1,m} \subseteq \{0, 1, \dots, 2^{c(\Omega^*)} - 1\}$ .

oznacza zbiór całkowicie określonych funkcji, których obrazy  $T(f_i^*)$  otrzymuje się przez zmianę wartości  $*$  pól obrazu  $T(f_{k,l,m})$ , posiadających tę wartość, w każdy możliwy sposób na wartość 0 lub 1, ale przy założeniu, że pola każdego zbioru rodziny  $R_{k,l,m}$  otrzymują jednakową wartość 0 lub 1. Dla każdej funkcji  $f_i^*$ ,  $i \in I_{k,l,m}$  zachodzą relacje  $(x_k^{\delta_k} \sim x_1^{\delta_1})_{f_i^*}$  oraz  $(x_1^{\delta_1} \sim x_m^{\delta_m})_{f_i^*}$ . Wynika stąd, że  $(x_k^{\delta_k} \sim x_1^{\delta_1} \sim x_m^{\delta_m})_{f_i^*}$ . W takim razie  $\Phi_{k,l,m} = \{f_i^*\}, i \in I_{k,l,m}$ . Jeśli w obrazie  $T(f_{k,l,m})$  nie ma pól o wartości  $*$  wówczas zbiór  $\Phi_{k,l,m}$  zawiera tylko jedną funkcję, a mianowicie funkcję  $f_{k,l,m}$ . Q.E.D.

Z dowodu dostateczności warunków (a), (b), (c) twierdzenia 4.4 wynika, że zbiór  $\Phi_{k,l,m}$  jest jednoznacznie określony za pomocą trójki  $(C_{k,l,m}^0, C_{k,l,m}^1, R_{k,l,m})$ . Twierdzenie 4.4 określa warunki kiedy dwa dwuelementowe zbiory symetrii mające niepuste przecięcie tworzą trzejelementowy zbiór symetrii. Obecnie uogólnimy powyższe twierdzenie na przypadek wieloelementowych zbiorów symetrii.

Niech  $A = \{a_1, \dots, a_p\}$ ,  $B = \{b_1, \dots, b_r\}$ ,  $A, B \subseteq \{1, \dots, n\}$ ,  $p, r \geq 2$  oraz  $A \cap B \subseteq \{k\}$ . Niech  $\tilde{X}_A$  i  $\tilde{X}_B$  oznaczają ciągi liter połączonych znakiem  $\sim$ :  $x_{a_1}^{\delta_{a_1}} \sim \dots \sim x_{a_p}^{\delta_{a_p}}$  oraz  $x_{b_1}^{\delta_{b_1}} \sim \dots \sim x_{b_r}^{\delta_{b_r}}$ . Załóżmy bez straty na ogólności, że  $a_1 = b_1 = k$ . Załóżmy dalej, że spełnione są relacje  $(\tilde{X}_A)_f$  i  $(\tilde{X}_B)_f$  oraz że zostały określone trójki odpowiednio  $\{C_A^0, C_A^1, R_A\}$  i  $\{C_B^0, C_B^1, R_B\}$ .

Twierdzenie 4.5.

$$(\tilde{X}_A)_f \wedge (\tilde{X}_B)_f \implies (\tilde{X}_{A \cup B})_f$$

wtedy i tylko wtedy, kiedy:

(a)  $C_A^0 \cap C_B^1 = \emptyset$

(b)  $C_A^1 \cap C_B^0 = \emptyset$

(c)  $\nexists (e^0, e^1) \in (C_A^0 \times C_A^1 \cup C_B^0 \times C_B^1) \exists (c_i^* \in R_A \cup R_B)$ ,  
 $\{e^0, e^1\} \subseteq c_i^*$ .

Twierdzenia 4.5 dowodzi się tak samo jak twierdzenia 4.4. Również tak jak w dowodzie dostateczności tego twierdzenia wyznacza się trójkę  $(C_{A \cup B}^0, C_{A \cup B}^1, R_{A \cup B})$ , która określa jednoznacznie zbiór  $\Phi_{A \cup B} \subseteq \mathbb{F}(f)$  taki, że  $(\bar{x}_{A \cup B}) \Phi_{A \cup B}$ .

Jak wynika z podanych poprzednio rozważań, pary symetrii  $\{x_1^{\delta_1}, x_j^{\delta_j}\}$ ,  $i, j \in \{1, \dots, n\}$ ,  $i \neq j$ , oraz odpowiadające im trójki  $(C_{i,j}^0, C_{i,j}^1, R_{i,j})$  łatwo można wyznaczyć na podstawie obrazu  $T(f)$ . Znając wszystkie takie pary i trójki można na podstawie twierdzenia 4.5 określić wszystkie maksymalne zbiory symetrii i odpowiadające im trójki.

Proces określenia maksymalnych zbiorów symetrii można też realizować bezpośrednio na podstawie obrazu  $T(f)$ . Sprawdza się symetrię względem poszczególnych par zmiennych w kolejności jak w algorytmie  $S^c$ . Przez zmianę wartości na 0 lub 1 dąży się do spełnienia warunku (a), a - jeśli to niemożliwe - warunku (b) twierdzenia 4.2 dla kolejnych par zmiennych. Przy rozpatrywaniu danej pary zmiennych zakłada się istnienie ustalonych w poprzednich krokach nowych wartości pól wyjściowego zbioru  $F^*$ . W ten sposób w procesie tym zbiór  $F^*$  stopniowo redukuje się. Jeśli począwszy od pewnej pary zmiennych nie można spełnić ani warunku (a), ani (b), oznacza to, że uzyskane pary symetrii tworzą jeden maksymalny zbiór symetrii. Poszukiwanie następnego takiego zbioru polega na powtórzeniu powyższego procesu przy próbie spełnienia warunku (a) lub (b) dla innych par zmiennych, takich, dla których istnienie symetrii nie jest wykluczone wynikami realizacji poprzedniego procesu. Wynika z tego, że określenie w powyższy sposób każdego maksymalnego zbioru symetrii wymaga oddzielnej "kopii" obrazu  $T(f)$ .

Założmy teraz, że funkcja  $f$  jest  $m$ -wyjściowa. Wówczas traktuje się ją jako zbiór  $m$  funkcji jednowyjściowych i dla każdej z nich określa się maksymalne zbiory symetrii. Zbiory symetrii danej funkcji  $f$  uzyskuje się jako przecięcia zbiorów symetrii poszczególnych funkcji jednowyjściowych /100/.

**Przykład 4.2.** Określić maksymalny zbiór symetrii funkcji  $f$  o obrazie  $T(f)$  przedstawionym na rys. 4.9 i przedstawić funkcję w postaci (4.8).

Sprawdzamy możliwości spełnienia warunku (a) lub (b) dla par



	(1)	(1)	(1)	(0)			1	0
	(1)	(0)	(1)	(1)			1	0
	1	0	(1)	(0)	(1)	1	1	(0)
$x_1$	(1)	1	0	1	1	(0)	1	(1)
	1	0	1	0	1	(1)	(1)	(0)
$x_2$	(1)	1	0	(1)	1	0	(1)	1
	(1)	0			(1)	0	(1)	(0)
$x_3$	(0)	(1)	1		1	1	0	(1)
					$x_4$	$x_5$	$x_6$	

Rys. 4.10. Obraz  $T(f)$  po ustanowieniu symetrii względem zbioru  $x_1, x_2, x_4, x_5$

Enakiem  $\odot$  oznaczone pola, których gwiazdy składają się z pojedynczych zespołów, to znaczy zaznaczonych zespołów jądrowych. Na rys. 4.12 przedstawiono pewne pokrycie  $M^q(T)$  obrazu  $T(f)$ , po włączeniu jądra obrazu do zbioru  $F^*$ . Dla pokrycia tego  $\Delta = 0$ ,  $\delta = 4$  (mimo, że  $\delta > 0$  jest - jak można wykazać - pokryciem ściśle minimalnym).

Powyższe pokrycie otrzymano po kilkukrotnym powtórzeniu algorytmu  $A^q$  (zespoły  $L_8, L_9, L_{10}, L_{11}$  nie są quasi-ekstremalami). Wyrażenie funkcji odpowiadające pokryciu  $M^q(T)$  uzupełnionemu ekstremalami jądrowymi jest równe:

$$f = \bigvee_{i=1}^{13} \alpha_i$$

gdzie

$$\alpha_1 = \alpha(L_1)$$

$$\alpha_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$$

$$\alpha_2 = \bar{x}_3 \bar{x}_6$$

$$\alpha_3 = \bar{x}_1 \bar{x}_3 \bar{x}_4 \bar{x}_5$$

$$\alpha_4 = \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$$

$$\alpha_5 = \bar{x}_1 \bar{x}_2 \bar{x}_4 \bar{x}_5$$

$$\alpha_6 = \bar{x}_1 \bar{x}_2 \bar{x}_6$$

$$\alpha_7 = x_3 \bar{x}_4 \bar{x}_5 \bar{x}_6$$

$$\alpha_8 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$$

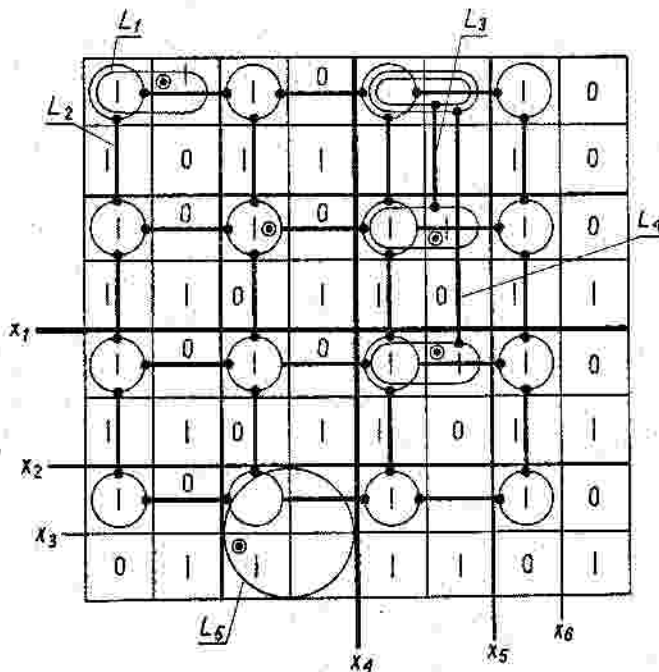
$$\alpha_9 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$$

$$\alpha_{10} = x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$$

$$\alpha_{11} = x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$$

$$\alpha_{12} = x_4 \bar{x}_5 \bar{x}_6$$

$$\alpha_{13} = x_1 \bar{x}_2 \bar{x}_3 \bar{x}_6$$

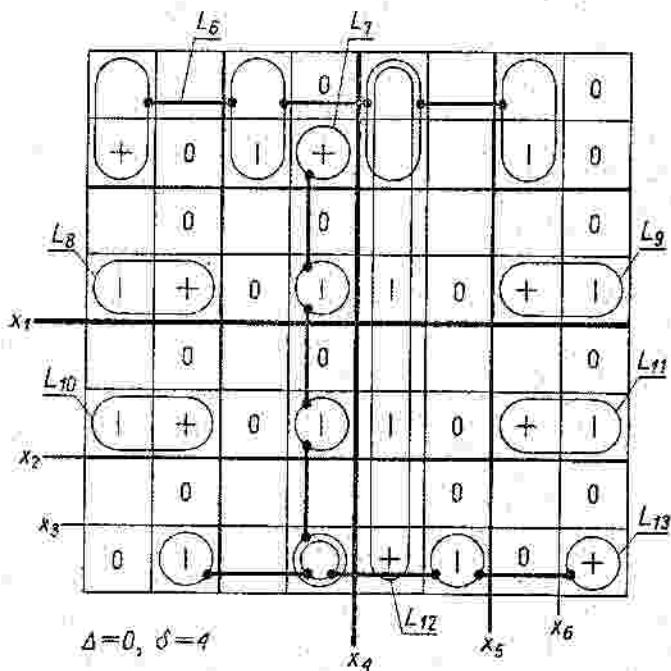


Rys. 4.11. Obraz  $T(f)$  wraz z zaznaczonymi zespołami jądrowymi

Określamy funkcje  $f_j = f_j(x_3, x_6)$ ,  $j = 0, 1, \dots, 4$ :

$$f_0 = \bar{x}_3 \bar{x}_6 \vee \bar{x}_3 \vee \bar{x}_6 \quad f_2 = \bar{x}_3 \bar{x}_6 \vee x_6 \vee x_3 \bar{x}_6 \quad f_4 = 1$$

$$f_1 = \bar{x}_3 \bar{x}_6 \vee \bar{x}_3 \vee \bar{x}_6 \quad f_3 = \bar{x}_3 \bar{x}_6 \vee x_3 \bar{x}_6$$



Rys. 4.12. Obraz  $T(f)$  wraz z zaznaczonym pokryciem  $M^Q(T)$

Po wstawieniu do wzoru (4.8) i dokonaniu odpowiednich przekształceń otrzymujemy:

$$f = \bar{x}_3 \bar{x}_6 \vee S_{0,1}(X) \bar{x}_3 \vee S_{0,1,2}(X) \bar{x}_6 \vee S_{2,3}(X) x_3 x_6 \vee S_4(X)$$

gdzie  $X = \{x_1, x_2, \bar{x}_4, x_5\}$ .

#### 4.4. Uwagi o możliwościach praktycznej realizacji metody

W opisaney metodzie rozpoznawania symetrii funkcji logicznych wykonuje się operacje sprawdzenia identyczności pewnych podobrazów obrazu funkcji, gdy funkcja jest całkowicie określona ( $\Omega^* = \emptyset$ ), lub uzyskania identyczności tych podobrazów przez zmianę wartości  $*$  na 1 lub 0, gdy funkcja jest częściowo określona ( $\Omega^* \neq \emptyset$ ).



Operacje tego rodzaju są proste do wykonania zarówno ręcznego jak i maszynowego.

Przy ręcznej realizacji metodą tą można łatwo rozpoznawać symetrię całkowitą lub częściową dowolnej funkcji logicznej do około 8 zmiennych w przypadku funkcji całkowicie określonych. W przypadku funkcji częściowo określonych trudności zależą w dużym stopniu również od mocy zbioru  $\Omega^*$  i dlatego przy większych wartościach  $c(\Omega^*)$  mogą powstać trudności w zrealizowaniu metody już przy mniejszej liczbie zmiennych.

Przy realizacji maszynowej obraz funkcji zapisuje się w postaci macierzy, toteż rozpoznawanie symetrii będzie polegało w tym przypadku na porównywaniu odpowiednich podmacierzy tej macierzy. Operacje takie wykonuje się łatwo na maszynie cyfrowej, szczególnie na maszynie typu równoległego, w której możliwe jest jednoczesne porównanie wartości większej liczby elementów porównywanych podmacierzy. Ograniczoność stosowalności metody wynika przede wszystkim z możliwości zapamiętania macierzy  $T(f)$ .

## 5. WYKORZYSTANIE INFORMACJI O SYMETRII FUNKCJI LOGICZNEJ DO SYNTETY JEJ WYRAŻENIA MINIMALNEGO

### 5.1. Wykorzystanie informacji o symetrii do syntezy wyrażeń minimalnych funkcji całkowicie symetrycznych

#### 5.1.1. Uwagi wstępne

Funkcję logiczną całkowicie symetryczną  $f = S_A(X)$  określa jednoznacznie zbiór symetrii  $X$  oraz zbiór indeksów  $A$ . Zbiory  $X$  i  $A$  mogą być zatem wykorzystane do bezpośredniego określenia różnych pochodnych informacji o funkcji  $f$ , które są wymagane przy syntezy jej wyrażenia minimalnego. W pktcie 5.1.2 opisano sposób bezpośredniego określenia na podstawie zbioru  $X$  i  $A$  zbioru  $P(f)$  wszystkich prostych implikantów funkcji całkowicie symetrycznej  $f$ , a w pktcie 5.1.4 - sposób określenia gwiazd  $G(e)$  oraz zbioru pól w obrazie  $T(f)$ , których gwiazdy są rozłączne, to znaczy zbioru  $F^r$ . Dzięki prostocie powyższych sposobów określenia gwiazd i zbioru  $F^r$  realizacja opisanych w pracy algorytmów syntezy pokryć, przede wszystkim algorytmu  $A^q$ , może być znacznie uproszczona.

W pracy /101/ opisano metodę, której celem jest bezpośrednio określenie na podstawie zbioru indeksów  $A$  wyrażenia minimalnego funkcji całkowicie symetrycznej względem zmiennych. Po bliższym zbadaniu metoda ta okazała się jednak metodą niepoprawną. Wyrażenie algebraiczne uzyskiwane tą metodą spełnia bowiem jedynie warunek konieczny, a nie dostateczny, aby być ekwiwalentne danej funkcji. Z tego powodu tylko w niektórych prostszych przypadkach (rozpatrywanych w przykładach podanych przez autora omawianej pracy) metodą tą można uzyskać wyrażenie minimalne danej funkcji. Metodę tę można zmodyfikować na przykład w ten sposób, aby każde równoważne, ze względu na złożoność, otrzymywane wyrażenie sprawdzać, czy jest jednocześnie ekwiwalentne zadanej funkcji. Tak zmodyfikowana metoda jest jednak bardziej pracochłonna (jak wynika z przykładów, które rozważał autor) niż metoda opisywana w pktcie 5.1.5.

Według podanego w pktcie 5.1.2 wzoru określającego liczbę prostych implikantów, jakie posiada funkcja całkowicie symetryczna, w pktcie 5.1.3 określono funkcje logiczne posiadające maksymalną liczbę prostych implikantów wśród funkcji danej liczby zmiennych. Przy pisaniu pktów 5.1.2 i 5.1.3 brano pod uwagę wyniki pracy /52/.

#### 5.1.2 Określenie zbioru $P(f)$ wszystkich prostych implikantów funkcji całkowicie symetrycznej

Dana jest funkcja całkowicie symetryczna  $f = S_A^n(X)$ . Załóżmy najpierw, że  $X = \{x_1, x_2, \dots, x_n\}$ . Funkcję  $f$  można przedstawić jako sumę elementarnych funkcji symetrycznych:

$$f = \bigvee_{a \in A} S_a^n \quad (5.1)$$

Rozbijmy zbiór  $A$  na maksymalne względem inkluzji podzbiory, które nazywa się blokami, mające tę własność, że każdy z nich stanowi ciąg kolejnych liczb całkowitych. W przypadku gdy  $a \in A$ ,  $a - 1$ ,  $a + 1 \notin A$ , powstaje blok zawierający tylko ten jeden element  $a$ .

Oznaczmy symbolem  $\tilde{S}_a^1$ ,  $a \in \{0, 1, \dots, n\}$ ,  $a \leq l \leq n$ , zbiór wszystkich koniunkcji  $\alpha_j$  o długości  $l$ , złożonych z liter zmiennych  $x_1, x_2, \dots, x_n$  i zawierających  $a$  liter prostych  $x_1$  oraz

l - a liter zanegowanych  $\bar{x}_1$ . Moc zbioru  $\tilde{S}_a^1$  wynosi  $C_n^1 C_1^a$ .

Zbiór  $\tilde{S}_a^n$  pokrywa się ze zbiorem składników funkcji  $S_a^n$ . Oznaczmy przez  $S_a^1$  dysjunkcję koniunkcji zbioru  $\tilde{S}_a^1$ .

Założmy, że zbiór A zawiera blok  $B = \{a, a+1\}$ ,  $a \leq n-1$ . Funkcja  $S_a \vee S_{a+1}$  implikuje więc daną funkcję f. Zbiór jej składników pokrywa się ze zbiorem  $\tilde{S}_a \cup \tilde{S}_{a+1}$ . Dla każdej koniunkcji ze zbioru  $\tilde{S}_a^n$  można w zbiorze  $\tilde{S}_{a+1}^n$  znaleźć koniunkcję sąsiednią i odwrotnie. Jeśli w obu zbiorach  $\tilde{S}_a^n$  i  $\tilde{S}_{a+1}^n$  powtórzyć odpowiednie koniunkcje dostateczną liczbę razy, to wówczas zbiór będący sumą tych zbiorów można rozbić na pary koniunkcji sąsiednich. Każdej takiej parze, zgodnie ze wzorem (1.5), odpowiada jedna koniunkcja o długości  $n-1$ , zawierająca a liter prostych i  $n-1-a$  liter zanegowanych. Zbiór koniunkcji odpowiadających wszystkim parom będzie równy zbiorowi  $\tilde{S}_a^{n-1}$ . Z tego wynika, że zachodzi równość:

$$S_a^n \vee S_{a+1}^n = S_a^{n-1} \quad (5.2)$$

Założmy teraz, że w zbiorze A istnieje blok  $b_k = \{a, a+1, a+2\}$   $a \leq n-2$ . Zgodnie z (5.2) zachodzą równości  $S_a^n \vee S_{a+1}^n = S_a^{n-1}$  i  $S_{a+1}^n \vee S_{a+2}^n = S_{a+1}^{n-1}$ . Dla każdej koniunkcji ze zbioru  $\tilde{S}_a^{n-1}$  można w zbiorze  $\tilde{S}_{a+1}^{n-1}$  znaleźć koniunkcję sąsiednią i odwrotnie. Analogicznie do poprzedniego można wykazać, że zachodzi równość:

$$S_a^n \vee S_{a+1}^n \vee S_{a+2}^n = S_a^{n-2} \quad (5.3)$$

Kontynuując powyższe rozważania można dowieść, że jeśli zbiór A zawiera blok  $B = \{a, a+1, \dots, a+k\}$ ,  $a \leq n-k$ , to zachodzi równość:

$$S_a^n \vee S_{a+1}^n \vee \dots \vee S_{a+k}^n = S_a^{n-k} \quad (5.4)$$

Zbiór  $\tilde{S}_a^{n-k}$  zawiera  $C_n^{n-k} C_{n-k}^a = \frac{n!}{a!b!k!}$ , gdzie  $b = n-a-k$ , koniunkcji o długości  $n-k = a+b$ , posiadających a liter prostych oraz b liter zanegowanych.

Zgodnie z określeniem bloku zachodzą relacje:  $a-1 \notin A$  i  $a+k+1$

¶ A. Z tego wynika, że dla żadnej koniunkcji ze zbioru  $\bar{S}_a^n \cup \bar{S}_{a+1}^n \cup \dots \cup \bar{S}_{a+k}^n$  nie może istnieć w wyrażeniu kanonicznym funkcji  $f$  koniunkcja sąsiednia, która do tego zbioru nie należy. Koniunkcje zbioru  $\bar{S}_a^{n-k}$  nie mogą więc zawierać w sobie koniunkcji krótszych, które również implikują funkcję  $f$ , są zatem prostymi implikantami tej funkcji.

Założmy, że zbiór indeksów  $A$  funkcji  $f$  składa się z bloków  $B_i = \{a^i, a^i + 1, a^i + 2, \dots, a^i + k^i\}$ ,  $i = 1, 2, \dots, d$ .

Twierdzenie 5.1. Zbiór  $P(f)$  wszystkich prostych implikantów funkcji  $f = S_A^n$  wyraża wzór

$$P(f) = \bigcup_{i=1}^d \bar{S}_{a^i}^{n-k^i} \quad (5.5)$$

a jego moc równa jest:

$$p(f) = \sum_{i=1}^d \frac{n!}{a^i! b^i! k^i!} \quad (5.6)$$

gdzie  $b^i = n - a^i - k^i$ .

Dowód. Twierdzenie wynika bezpośrednio z poprzednich rozważań.

Q.E.D.  
Założmy obecnie, że zbiór symetrii  $X = \{x_1^{61}\}_{i=1}^n$  zawiera przynajmniej jedną literę typu  $\bar{x}_1$ . Jeśli  $X'$  oznacza zbiór  $\{x'_i\}_{i=1}^n$ , gdzie  $x'_i = x_1^{61}$ ,  $i = 1, 2, \dots, n$ , to zachodzi równość

$$f(x'_1, x'_2, \dots, x'_n) = S_A(X').$$

Przez zmianę zmiennych  $x_i$  na  $x'_i$  obecny przypadek sprowadza się więc do poprzedniego.

Na podstawie twierdzenia 5.1 i powyższej uwagi można dla każdej funkcji całkowicie symetrycznej  $f = S_A(X)$  wyznaczyć zbiór jej prostych implikantów bezpośrednio na podstawie zbiorów  $X$  oraz  $A$ .

### 5.1.3. Określenie funkcji logicznych posiadających maksymalną liczbę prostych implikantów przy danej liczbie zmiennych

Określmy funkcje  $f = S_A(X)$ ,  $X = \{x_i\}_{i=1}^n$ ,  $A \subseteq \{0, 1, 2, \dots, n\}$ , posiadające maksymalną liczbę prostych implikantów wśród funkcji całkowicie symetrycznych  $n$  zmiennych. Funkcje te, jak wynika z rozważań podanych w pracy /52/, będą miały zarazem maksymalną liczbę prostych implikantów wśród wszystkich funkcji logicznych  $n$  zmiennych, to znaczy równą  $p(n)$  (zob. wzór (1.13)).

Wartość  $p(f)$  we wzorze (5.6) nie zależy od tego, jaki jest zbiór całkowitej symetrii  $X$ , to jest, jakie są wartości  $s_i$ ,  $i = \{1, 2, \dots, n\}$ , lecz jedynie od zbioru  $a$ -indeksów  $A$ . Maksymalna liczba różnych zbiorów  $a$ -indeksów przy danej liczbie zmiennych  $n$  jest równa maksymalnej liczbie różnych podzbiorów zbioru  $\{0, 1, 2, \dots, n\}$ , to znaczy  $2^{n+1}$ . Oznaczmy te różne zbiory  $a$ -indeksów symbolem  $A_j$ ,  $j = 1, 2, \dots, 2^{n+1}$ . Wyznaczenie funkcji posiadających maksymalną liczbę prostych implikantów polega na znalezieniu wśród zbiorów  $A_j$  takiego zbioru  $A(n)$ , dla którego wyrażenie (5.6) osiąga maksimum:

$$p(n) = \max_j p(f_j) \quad (5.7)$$

gdzie  $f_j$  - funkcja całkowicie symetryczna posiadająca zbiór  $a$ -indeksów  $A_j$ .

Określenie ogólnego wzoru wyznaczającego zbiór  $A(n)$  dla dowolnego  $n$  jest zadaniem trudnym. Okazuje się jednak, że przez rozwiązanie prostych nierówności można wyznaczyć zbiór  $A(n)$ , jeśli znany jest zbiór  $A(n-1)$ . Określenie zaś zbioru  $A(n)$  dla dostatecznie małej wartości  $n$ , na przykład  $n = 3$ , jest zadaniem trywialnym. Zbiory  $A(n)$  dla określonego zakresu wartości  $n$  można więc łatwo wyznaczyć w sposób iteracyjny. W tablicy przedstawionej na rys. 5.1 podano zbiory  $A(n)$  i odpowiadające im wartości  $p(n)$  dla zakresu wartości  $n = 1, 2, 3, \dots, 24$ .

Przy wartości  $n = 2$  występują trzy równoważne, a przy wartościach  $n = 1, 4, 7, 10, 13, 16, 19, 22$  - dwa równoważne zbiory  $A(n)$  (zbiory te połączono klamrą). Niektóre drobne różnice w wartościach  $p(n)$  w stosunku do wartości podanych w pracy /52/ wynikają

n	zbiór A(n)	p(n)
1	{0}	1
2	{0, 1}	2
3	{0, 1, 2}	6
4	{0, 1, 2, 3}	13
5	{0, 1, 2, 3, 4}	32
6	{0, 1, 2, 3, 4, 5}	92
7	{0, 1, 2, 3, 4, 5, 6}	218
8	{0, 1, 2, 3, 4, 5, 6, 7}	576
9	{0, 1, 2, 3, 4, 5, 6, 7, 8}	1 698
10	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}	4 300
11	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}	11 770
12	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}	34 914
13	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}	91 105
14	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}	254 438
15	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}	759 488
16	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}	2 030 618
17	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17}	5 746 274
18	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}	17 189 858
19	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19}	46 698 068
20	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}	133 334 440
21	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21}	399 479 982
22	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22}	1 099 206 284
23	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23}	3 159 208 516
24	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24}	9 470 895 656

Rys. 5.1. Zbiory A(n) oraz wartości p(n) dla n = 1, 2, ..., 24

z pewnych niedokładności, jakie widocznie wkładły się do pracy /52/.

Na podstawie znalezionych wartości  $\Lambda(n)$  funkcje o maksymalnej liczbie prostych implikantów przy danej liczbie zmiennych  $n$  określamy jako:

$$f_j = S_{\Lambda(n)}(X_j), \quad j = 0, 1, \dots, 2^n - 1$$

gdzie  $X_j = \{x_1^{\delta_1}, x_2^{\delta_2}, \dots, x_n^{\delta_n}\}$  - zbiory całkowitej symetrii powstające przez przypisanie indeksom  $\delta_i$  różnych wartości ze zbioru  $\{0, 1\}$ .

5.1.4. Określanie w obrazie funkcji całkowicie symetrycznej gwiazd pól  $G(e)$  oraz zbioru pól  $F^1$ , których gwiazdy są rozłączne

Założmy, że koniunkcja  $\alpha = x_1^{\delta_1}, x_2^{\delta_2}, \dots, x_n^{\delta_n}$ , gdzie  $\sum_{i=1}^n \delta_i = \hat{a}$  jest składnikiem kanonicznego d.n.w. funkcji  $f = S_{\hat{a}}(X)$ , przy czym  $X = \{x_1, x_2, \dots, x_n\}$ . W takim razie  $\hat{a} \in A$ . Oznaczmy symbolem  $B = \{a, a+1, \dots, a+k\}$  blok w zbiorze  $A$ , którego elementem jest indeks  $\hat{a}$ . Niech  $e$  oznacza pole w obrazie  $T(f)$  odpowiadające koniunkcji  $\alpha$ . Skoro  $\alpha$  jest składnikiem wyrażenia kanonicznego, więc  $e \in F^1$ . Gwiazdzie  $G(e)$  odpowiada na gruncie algebraicznym zbiór prostych implikantów, których litery zawierają się w koniunkcji  $\alpha$ . Oznaczmy taki zbiór prostych implikantów symbolem  $G(\alpha)$ . Zgodnie z twierdzeniem 5.1 w zbiorze  $P(f)$  wszystkich prostych implikantów funkcji  $f$  zawiera się zbiór  $\tilde{S}_a^{n-k}$ . Ponieważ  $\hat{a} \in B$ , więc  $G(\alpha) \subseteq \tilde{S}_a^{n-k}$ . Z powyższych rozważań wynika, że  $G(\alpha)$  jest zbiorem koniunkcji o długości  $n - k$ , które zawierają  $a$  liter prostych i  $n - a - k$  liter zanegowanych, przy czym wybranych ze zbioru liter koniunkcji  $\alpha$ . Zatem gwiazda pola  $e$ :

$$G(e) = \{E(\alpha_i)\}$$

gdzie  $\alpha_i \in G(\alpha)$ . Moc gwiazdy  $c(G(e)) = C_a^a C_a^{n-k-a}$ .

Zespoły tej gwiazdy są jednakowego rozmiaru, ponieważ wszystkie koniunkcje zbioru  $\tilde{S}_a^{n-k}$  są takiej samej długości.

Powyższe rozumowanie dotyczyło dowolnej koniunkcji wyrażenia

kanonicznej funkcji  $f$ , a więc tym samym dowolnego pola  $e \in F^1$ . Oznacza to, że każda gwiazda w obrazie funkcji całkowicie symetrycznej  $f$  składa się z zespołów takiego samego rozmiaru.

W przypadku gdy zbiór symetrii  $X$  funkcji  $f$  jest równy  $\{x_i^{61}\}_{i=1}^n$  i zawiera litery zanegowane, wówczas dokonuje się zamiany zmiennych  $x_i$  na zmienne  $x_i'$ , gdzie  $x_i' = x_i^{61}$ ,  $i=1, 2, \dots, n$ , przez co przypadek ten sprowadza się do poprzedniego. Dlatego, nie tracąc na ogólności, w dalszych rozważaniach dla prostoty będziemy zakładali, że zbiór symetrii  $X = \{x_i\}_{i=1}^n$ .

Obecnie pokażemy, jak na podstawie zbioru  $A$  oraz  $X$  znaleźć w obrazie  $T(f)$  zbiór pól, których gwiazdy tworzą pewną maksymalną względem inkluzji rodzinę gwiazd rozłącznych.

Dowolny zespół rozmiaru 1 w tablicy logicznej  $n$  zmiennych składa się z pól, których wagi różnią się o 1. Wynika to z tego, że polom tego zespołu odpowiadają koniunkcje sąsiednie. Łatwo wykazać, że zespół rozmiaru 2 zawiera jedno pole o wadze  $a$ ,  $a \in \{0, 1, \dots, n-1\}$ , dwa pola o wadze  $a+1$  oraz jedno pole o wadze  $a+2$ . Analogicznie można wykazać, że zespół rozmiaru  $r$  zawiera:

$$\begin{aligned} C_r^0 & \text{ pól o wadze } a \\ C_r^1 & \text{ pól o wadze } a+1 \\ & \vdots \\ C_r^r & \text{ pól o wadze } a+r \end{aligned}$$

gdzie  $a \in \{0, 1, \dots, n-r\}$ .

Zespół  $E(\alpha_i)$  odpowiadający dowolnej koniunkcji  $\alpha_i$  ze zbioru  $S_a^{n-k}$  jest zespołem rozmiaru  $k$ . Pokrywa on  $C_k^0$  pól o wadze  $a$ ,  $C_k^1$  pól o wadze  $a+1$ , ...,  $C_k^k$  pól o wadze  $a+k$ . Zespół ten pokrywa więc tylko jedno pole o wadze  $a$  i tylko jedno pole o wadze  $a+k$ . Z tego wynika, że każde pole o wadze  $a$  lub  $a+k$  jest pokryte oddzielnymi zespołami. Oznacza to, że pola o wadze  $a$  tworzą zbiór pól, których gwiazdy są rozłączne. Inny analogiczny zbiór tworzą pola o wadze  $a+k$ . Oznaczmy te zbiory odpowiednio:  $E^1$  i  $E^2$ . Moc zbioru  $E^1$  wynosi  $C_n^a$ , a moc zbioru  $E^2 = C_n^{a+k}$ . Zbiory  $E^1$  i  $E^2$  są równoliczne, gdy  $a = n-a-k$ , co równoważne jest, że



$$a = \frac{n - k}{2} \quad (5.8)$$

Łatwo zauważyć, że gdy  $a > (n - k)/2$ , to liczniejszy jest zbiór  $E^1$ , w przeciwnym zaś przypadku zbiór  $E^2$ .

Jeśli zbiór  $A$  składa się tylko z bloku  $B$ , to zbiory  $E^1$  lub  $E^2$  są pewnymi maksymalnymi względem inkluzji zbiorami pól, których gwiazdy są rozłączne. Jeśli zbiór  $A$  składa się z kilku bloków, na przykład bloków  $B_i = \{a^i, a^i + 1, \dots, a^i + k^i\}$ ,  $i = 1, 2, \dots, d$ , to analogiczne zbiory pól określone są jako sumy

$$\bigcup_{i=1}^d E_i^p \quad (5.9)$$

gdzie  $p \in \{1, 2\}$ ;  $E_i^p$  - zbiór pól o wagach równych  $a^i$ , jeśli  $p = 1$  lub  $a^i + k^i$ , jeśli  $p = 2$ .

Ponieważ dla każdego  $i$ ,  $i = 1, \dots, d$ , można przyjąć  $p$  równe 1 lub 2, przeto wynika, że zbiorów takich jest  $2^d$ .

Każdy powyższy zbiór jest równoważny pewnemu zbiorowi  $F^r$  w algorytmach  $A^m$  lub  $A^q$ , to znaczy zbiorowi pól, których gwiazdy tworzą rodzinę gwiazd rozłącznych. Oznaczmy omawiane zbiory symbolem  $F_j^r$ ,  $j = 1, 2, \dots, 2^d$ . Realizując algorytm  $A^m$  lub  $A^q$  w przypadku funkcji całkowicie symetrycznej  $f$ , zbiór  $F^r$  można określać bezpośrednio ze zbioru  $A$ , bez posługiwania się sposobem podanym w pktcie 3.5.2 (algorytm R). Ponieważ z punktu widzenia algorytmu  $A^m$  lub  $A^q$  korzystne jest, aby zbiór  $F^r$  miał możliwie największą moc, więc przy realizacji tych algorytmów z wykorzystaniem powyższej możliwości jako zbiór  $F^r$  przyjmuje się najliczniejszy spośród zbiorów  $F_j^r$ . Zbiorem takim jest zbiór powstający przez wybranie dla każdego  $i = 1, 2, \dots, d$  liczniejszego spośród zbiorów  $E_1^1$  i  $E_1^2$ .

#### 5.1.5. Określanie na podstawie zbioru $a$ -indeksów jądra pokrycia

Niech  $B = \{a, a + 1, \dots, a + k\}$  oznacza pewien blok w zbiorze  $a$ -indeksów funkcji  $f = S_A(X)$ , gdzie  $X = \{x_1, \dots, x_n\}$ . Załóżmy, że pierwszy element tego bloku  $a = 0$ . Każdy zespół odpowiadający koniunkcjom zbioru  $\tilde{S}_0^{n-k} \in P(f)$  pokrywa jednocześnie pole  $e^0$  (po-

le o numerze  $0$ ) i tylko jedno pole spośród  $C_n^{a+k}$  pól o wadze  $a+k$ . Oznacza to, że gwiazdy pól o wadze  $a+k$  składają się tylko z jednego zespołu maksymalnego. Zespoły  $L(\alpha_1)$ ,  $\alpha_1 \in \tilde{S}_0^{n-k}$ , są więc zespołami jądrowymi w obrazie  $T(f)$ .

Analogiczna sytuacja zachodzi wówczas, gdy ostatni element bloku  $B$ , tj. element  $a+k=n$ . Wówczas zespołami jądrowymi są zespoły odpowiadające koniunkcjom zbioru  $\tilde{S}_{n-k}^{n-k} \subseteq P(f)$ .

Jeśli w zbiorze  $A$  są bloki jednoelementowe, to pola w obrazie  $T(f)$  o wagach równych tym elementom, a więc należące do zbioru  $F^1$ , wobec braku pól o wagach różniących się od ich wag o  $1$  nie będą mogły utworzyć zespołu z żadnym innym polem zbioru  $F^1$ . Wynika z tego, że zespoły składające się z tych pojedynczych pól są zespołami jądrowymi.

Jeśli zbiór  $A$  nie zawiera elementu ani  $0$  lub  $n$ , ani też nie ma bloków jednoelementowych, to - jak łatwo zauważyć - gwiazdy  $G(a)$ ,  $a \in F^1$ , będą zawierać co najmniej dwa zespoły maksymalne pokrywające różne pola zbioru  $F^1$ . Obraz  $T(f)$  jest w takim przypadku obrazem cyklicznym. Wynika z tego, że w obrazie  $T(f)$  istnieją zespoły jądrowe tylko w przypadkach omówionych powyżej.

Zespoły maksymalne odpowiadające koniunkcjom zbioru  $\tilde{S}_a^{n-k}$  pokrywają tylko pola o wagach  $a, a+1, \dots, a+k$ . Dlatego, jeśli w obrazie  $T(f)$  znajdują się zespoły jądrowe, to po włączeniu pól, które one pokrywają, do zbioru  $F^*$  obraz  $T(f)$  stanie się obrazem cyklicznym (zakłada się, że w zbiorze  $A$  są również inne bloki niż te, którym odpowiadają zespoły jądrowe).

#### 5.1.6. Realizacja algorytmów $A^m$ i $A^q$ przy wykorzystaniu informacji o całkowitej symetrii funkcji

Założmy, że dany jest obraz  $T(f)$  o którym, metodą rozpoznawania symetrii opisaną w rozdz. 4 stwierdzono, że jest obrazem funkcji całkowicie symetrycznej  $f = S_A(X)$ . Na podstawie wyników podanych w pktach 5.1.4 i 5.1.5 pokażemy obecnie, jak można zmodyfikować algorytmy  $A^m$  i  $A^q$  przy określaniu pokrycia obrazu  $T(f)$ , gdy funkcja  $f$  jest całkowicie symetryczna. Założmy, jak poprzednio, że zbiór  $X$  nie zawiera liter zanegowanych. Jeśli zbiór ten zawiera takie litery, to obraz  $T(f)$  o osiach  $x_1, x_2, x_3, \dots, x_n$  przekształca się w obraz  $T(f)$  o osiach  $x'_1, x'_2, \dots, x'_n$ , gdzie

$x'_i = \overset{6_i}{x_i}, \overset{6_i}{x_i} \in X, i = 1, 2, \dots, n$ . Otrzymany obraz jest wówczas obrazem funkcji całkowicie symetrycznej posiadającej zbiór symetrii  $X = \{x'_i\}_{i=1}^n$ .

Na podstawie zbioru  $A$  określa się zespoły jądrowe (jeśli zachodzą przypadki omówione w pktcie 5.1.5), a jądro obrazu włącza się do zbioru  $F^*$ . W rezultacie otrzymuje się obraz cykliczny.

Realizacja algorytmu  $A^m$  po wykonaniu powyższej operacji będzie polegać na bezpośrednim wykonywaniu algorytmu  $C$ . Zbiór pól  $F^Z$  określa się jako różnicę  $F^1 \setminus F^R$ , gdzie jako  $F^R$  przyjmuje się zbiór o maksymalnej mocy spośród omówionych poprzednio zbiorów  $F_j^R$ . Gwiazdy  $G(e)$  pól  $e \in F^Z$  tworzy się tak jak opisano w pktcie 5.1.4. Daleszy ciąg algorytmu pozostaje bez zmian z tym jednak, że zachodzi tu przypadek szczególny, w którym w pktcie 3.5.4 powiedziano, że umożliwiła uproszczenie algorytmu. Pokrycie minimalne można mianowicie w tym przypadku otrzymać bezpośrednio po określeniu pierwszego zbioru  $\tilde{\beta}_1$ , dla którego  $\Delta_1 = 0$ . Możliwość ta wynika z tego, że gwiazdy pól  $e \in F^Z$  tak podobnie jak gwiazdy pozostałych pól w obrazie  $T(f)$  mają jednakowe rozmiary.

Należy jednak stwierdzić, że przy syntezie pokrycia minimalnego obrazów funkcji całkowicie symetrycznych posiadających liczbę prostych implikantów zbliżoną lub równą  $p(n)$  algorytm  $A^m$  (już począwszy od  $n > 5$ ) może wymagać zbyt wielkiej liczby operacji mimo opisanych uproszczeń (jak to na przykład pokazano w przykładzie 3.4).

Obecnie opiszemy krótko możliwość uproszczenia algorytmu  $A^q$  przy syntezie pokrycia obrazu funkcji całkowicie symetrycznej. Po otrzymaniu obrazu cyklicznego, w wyniku włączenia jądra obrazu do zbioru  $F^*$ , określa się zbiór  $F_j^R$  o największej mocy. Zbiór ten traktuje się jako zbiór  $F^R$ . Następnie można kolejno określać sposobem opisanym w pktcie 5.1.4 gwiazdy pól  $e \in F^R$  i wybierać z nich quasi-ekstremale. Znacznikowanie pól zbioru  $F^1$  wobec znajomości zbioru  $F^R$  jest zbędne. Zamiast generować gwiazdy poszczególnych pól  $e \in F^R$ , znacznie prościej jest określać tylko jeden zespół maksymalny w każdej gwiazdce i traktować go jako quasi-ekstremalę. Jeśli po ukończeniu w ten sposób zrealizowanego algorytmu  $A^q$  parametry  $\Delta_1$  i  $\delta_1$  wypadną zbyt duże, to można go powtórzyć, określając inne zespoły w poszczególnych gwiazdach. Tak uproszczona wersja algorytmu może być realizowana nawet przy dużej

liczbie zmiennych i przy dużej liczbie prostych implikantów (np. zbliżonej do  $p(n)$  lub równej jej). W podobny właśnie sposób autor określił pokrycie minimalne obrazu funkcji  $S_{0,1,3,4,5,7,8}^8$  mających liczbę prostych implikantów równą  $p(8) = 576$  (o czym wspomniano w pktcie 3.7.4).

## 5.2. Niektóre możliwości wykorzystania symetrii przy syntezie wyrażeń minimalnych funkcji częściowo symetrycznych

### 5.2.1. Wykorzystanie informacji symetrii przy generowaniu gwiazdy $G(e)$

Dana jest funkcja częściowo symetryczna  $f$  posiadająca zbiór symetrii  $X$ . Niech  $e_k$  oznacza pewne pole zbioru  $F^1$  w obrazie  $T(f)$ , a  $\alpha_k$  - koniunkcję odpowiadającą temu polu. Załóżmy, że chcemy określić gwiazdę  $G(e)$  oraz że jeden zespół maksymalny  $L_k^1$  tej gwiazdy został już określony. Niech  $\alpha_k^1$  oznacza prosty implikant odpowiadający zespołowi  $L_k^1$ . Niech  $\tilde{\alpha}_k$  - oznacza zbiór liter koniunkcji  $\alpha_k$ , a  $\tilde{\alpha}_k^1$  - zbiór liter koniunkcji  $\alpha_k^1$ . Określmy zbiory:

$$X_1 = X \cap \tilde{\alpha}_k^1, \quad (5.10)$$

$$X_2 = (X \cap \tilde{\alpha}_k) \cap \tilde{\alpha}_k^1 \quad (5.11)$$

Jeśli zbiory te nie są puste, to wykonuje się następującą operację: wymienia się w koniunkcji  $\alpha_k^1$  dowolną literę należącą do zbioru  $X_1$  na literę ze zbioru  $X_2$ . Powstała w ten sposób koniunkcja, która ze względu na symetrię funkcji względem liter, które były wymienione, będzie również prostym implikantem funkcji  $f$ . Koniunkcja ta zawiera się w koniunkcji  $\alpha_k$ , a więc zespół jej odpowiadający będzie innym zespołem maksymalnym gwiazdy  $G(e)$ . Oznaczmy symbolem  $G^\alpha$  zbiór wszystkich koniunkcji powstających z koniunkcji  $\alpha_k^1$  przez analogiczną wymianę liter ze zbioru  $X_1$  na litery ze zbioru  $X_2$ . Zespoły odpowiadające tym koniunkcjom będą elementami gwiazdy  $G(e_k)$ . W ten sposób na podstawie jednego zespołu maksymalnego można określić pewne inne zespoły maksymalne generowanej gwiazdy  $G(e_k)$ .

5.2.2. Twierdzenie o możliwości określenia ekstremali w gwieżdździe na podstawie informacji o symetrii

Niech  $G^L$  oznacza zbiór zespołów odpowiadających koniunkcjom zbioru  $G^\alpha$ .

Twierdzenie 5.2. Jeśli

$$G^L = G(e_k)$$

to każdy zespół maksymalny gwiazdy  $G(e_k)$  jest ekstremalą.

Dowód. Zbiorowi  $G^L$  zespołów maksymalnych odpowiada zbiór koniunkcji  $G^\alpha$ . Do wyrażenia minimalnego funkcji  $f$  musi wejść przynajmniej jedna koniunkcja zbioru  $G^\alpha$ . Ponieważ każdą koniunkcję zbioru  $G^\alpha$  można otrzymać z dowolnej innej koniunkcji tego zbioru przez wymianę liter symetrii, przeto dokonując tej zamiany w wyrażeniu minimalnym zawierającym określoną koniunkcję zbioru  $G^\alpha$  otrzymujemy wyrażenia minimalne zawierające pozostałe koniunkcje tego zbioru. Q.E.D.

Na podstawie twierdzenia 5.2 można skonstruować pewną lokalną operację upraszczającą. Jeśli dla określonej gwiazdy  $G(e)$  warunek tego twierdzenia jest spełniony, to wówczas wybiera się dowolny zespół tej gwiazdy, ponieważ wiadomo, że istnieje pokrycie minimalne, którego jest on ekstremalą.

5.2.3. O pewnej możliwości redukcji liczby zmiennych przy syntezie wyrażenia minimalnego funkcji częściowo symetrycznej

Niech  $f_{k,1}^i$ ,  $i = 0, 1, 2, 3$ , będą funkcjami rezidualnymi rozwinięcia danej funkcji częściowo symetrycznej  $f(x_1, \dots, x_n)$  względem zmiennych  $x_k$  i  $x_1$  (zob. s. 104).

Twierdzenie 5.3. Jeśli

$$(x_k \sim x_1)_f \wedge (x_k \sim \bar{x}_1)_f \quad (a)$$

oraz

$$f_{k,1}^0 = 1 \quad (b) \quad \text{lub} \quad f_{k,1}^1 = 1 \quad (c)$$

to

$$M(f) = \bar{x}_1 \bar{x}_2 \vee x_1 x_2 \vee M(f_{k,1}^1) \quad - \text{gdy} \quad f_{k,1}^0 = 1$$

$$M(f) = \bar{x}_1 x_2 \vee x_1 \bar{x}_2 \vee M(f_{k,1}^0) \quad - \text{gdy} \quad f_{k,1}^1 = 1$$

Dowód. Zgodnie z twierdzeniem 4.2, relacja (a) zachodzi wówczas, gdy  $f_{k,1}^0 = f_{k,1}^3$  oraz  $f_{k,1}^1 = f_{k,1}^2$ . Uwzględniając powyższe równości we wzorze (4.3) (s. 83) otrzymujemy:

$$f = (x_k \bar{x}_1 \vee \bar{x}_1 \bar{x}_1) f_{k,1}^0 \vee (\bar{x}_k x_1 \vee x_k \bar{x}_1) f_{k,1}^1$$

Podstawiając  $\varphi = \bar{x}_k \bar{x}_1 \vee x_k x_1$ , otrzymuje się

$$f = \varphi f_{k,1}^0 \vee \bar{\varphi} f_{k,1}^1$$

Jeśli spełniony jest warunek (b), to:

$$f = \varphi \vee \bar{\varphi} f_{k,1}^1 = \varphi \vee f_{k,1}^1$$

Ponieważ  $f_{k,1}^1$  nie zależy od zmiennych  $x_k$  i  $x_1$ , więc wyrażenie minimalne:

$$M(f) = \varphi \vee M(f_{k,1}^1) = \bar{x}_k \bar{x}_1 \vee x_k x_1 \vee M(f_{k,1}^1)$$

Część twierdzenia odnoszącą się do warunku (c) dowodzi się analogicznie. Q.E.D.

Twierdzenie 5.3 umożliwia redukcję liczby zmiennych przy syntezie wyrażeń minimalnych pewnych szczególnych funkcji częściowo symetrycznych. Liczba  $t_5(n)$  funkcji logicznych  $n$  zmiennych spełniających warunki tego twierdzenia jest równa:

$$t_5(n) = 2 \cdot C_n^2 \cdot 2^{2^{n-2}} = n(n-1) 2^{2^{n-2}} = n(n-1) \sqrt[4]{t(n)} \quad (5.12)$$

gdzie  $t(n)$  - liczba wszystkich funkcji logicznych o liczbie zmiennych nie większej od  $n$ .

## ZAKOŃCZENIE

Podsumujemy krótko główne zagadnienia omówione w niniejszej pracy oraz uzyskane ważniejsze wyniki.

Rozdział 1 ma charakter wprowadzający i poświęcony jest podaniu pewnych wiadomości ogólnych oraz definicji pojęć podstawowych.

W rozdziale 2 sformułowano pojęcie tablicy logicznej n zmiennych oraz obrazu funkcji logicznej - pojęcia podstawowe dla przyjętego podejścia do rozwiązania podjętych w pracy problemów. Wprowadzono pojęcie zespołu pól oraz sformułowano twierdzenie o rozpoznawaniu zespołów na podstawie pewnych wyróżnionych pojęć geometrycznych.

Rozdział 3 jest poświęcony w całości zagadnieniom syntezy minimalnych i quasi-minimalnych pokryć obrazu funkcji, którym na gruncie algebraicznym odpowiadają minimalne i w przybliżeniu minimalne wyrażenia funkcji logicznej. Opracowano pewne algorytmy generowania gwiazd pól, wymagane dla skonstruowanych w pracy algorytmów syntezy pokryć. Do zasadniczych wyników tego rozdziału należy opracowanie dwóch algorytmów syntezy pokryć obrazu funkcji. Algorytm pierwszy, nazwany algorytmem  $A^m$ , powstał w wyniku dążenia do skonstruowania metody, która będzie wymagać możliwie najmniejszej, tylko teoretycznie niezbędnej, liczby operacji do określenia pokrycia ściśle minimalnego. Skonstruowanie tego algorytmu poprzedzono teoretyczną analizą możliwości określania elementów (ekstremal) konstruowanego minimalnego pokrycia obrazu za pomocą tzw. operacji lokalnych. Do rozwiązania problemu w przypadku, gdy sformułowane operacje lokalne nie prowadzą do uproszczenia obrazu (gdy obraz jest cykliczny), opracowano tzw. metodę rozłącznych gwiazd (algorytm C).

Drugi algorytm, nazwany algorytmem  $A^q$ , służy do określenia pokryć minimalnych lub w przybliżeniu minimalnych, bez konieczności przeglądu wyrażań nieredukowalnych, co umożliwia znaczną redukcję liczby operacji. Dzięki temu wydaje się on szczególnie korzystny przy rozwiązywaniu problemów pokrycia o dużej złożoności (w przypadku dużej liczby zmiennych i dużej liczby prostych implikan-tów).

Do istotnych cech tego algorytmu należy to, że w przypadku, gdy o otrzymanym pokryciu nie możemy stwierdzić, że jest minimalne

(choć w rzeczywistości może nim być) posiadamy oszacowanie, o ile może ono różnić się od minimalnego. Drugą istotną cechą algorytmu, tak samo zresztą jak algorytmu  $A^m$ , jest jego uniwersalność, umożliwiającą stosowanie go przy różnych sformułowaniach problemu pokrycia.

Algorytm  $A^q$  w sformułowaniu odnoszącym się do syntezy minimalnych wyrażeń normalnych funkcji logicznych (przy czym rozszerzony na przypadek funkcji wielowartościowych - zob. publikacja nr 6 poniżej) zaprogramowano w języku Ljapas na maszynie cyfrowej Odra 1204. Działająca obecnie wersja programu umożliwia syntezę quasi-minimalnych wyrażeń funkcji logicznych wielowartościowych przy ograniczeniu, że łączna liczba zmiennych funkcji  $n$  oraz liczba wyjść  $m$  nie przekracza 31, to jest  $n + m \leq 31$ .

Badania przeprowadzone na maszynie cyfrowej w pełni potwierdziły przydatność praktyczną algorytmu. Wyniki badań oraz opisy różnych modyfikacji algorytmu będą przedmiotem oddzielnej publikacji.

W rozdziale 4 opisano metodę rozpoznawania symetrii funkcji logicznych polegającą na zastosowaniu obrazu funkcji. W metodzie tej sprawdzenie symetrii określonych liter sprowadzono do operacji sprawdzenia identyczności (lub ustanowienia jej przez zmianę wartości  $*$  na 0 lub 1 w przypadku funkcji częściowo określonych) pewnych podobrazów obrazu funkcji. Dzięki temu metoda taka wydaje się korzystna zarówno dla ręcznej jak i dla maszynowej realizacji.

Rozdział 5 poświęcono wyjaśnieniu możliwości ułaskawienia syntezy wyrażeń minimalnych przez wykorzystanie informacji o symetrii funkcji. Podano szereg wyników umożliwiających istotne uproszczenie syntezy pokrycia minimalnego, przede wszystkim w przypadku funkcji całkowicie symetrycznych.

Mimo, że niniejsza praca dotyczy zasadniczo wyłącznie problemów syntezy wyrażeń minimalnych funkcji logicznych i rozpoznawania ich symetrii, to niektóre jej wyniki mogą być stosowane również w szerszym zakresie. Przede wszystkim odnosi się to do algorytmów syntezy pokryć  $A^m$  i  $A^q$ , o których w pktcie 3.7.3 wykazano, że mogą być zastosowane przy dowolnym sformułowaniu problemu pokrycia.

Na zakończenie warto dodać, że wprowadzone w pracy pojęcie obrazu funkcji może być łatwo rozszerzone na przypadek funkcji wielowartościowych. W związku z tym opracowany aparat pojęciowy oraz metody podejścia do rozwiązywanych problemów mogą okazać się przy-



datne przy rozwiązywaniu problemów teorii wielowartościowych układów logicznych.

Z tematem niniejszej pracy wiążą się następujące publikacje autora:

1. Graficzna minimalizacja w klasie alternatywnych normalnych wyrażań funkcji logicznych w oparciu o tablice typu tablic Veitcha-Karnaugh. Warszawa 1967. Pr. Inst. Automat. PAN z. 52.
2. Metoda określania wyrażenia minimalnego dwuwartościowej funkcji logicznej bezpośrednio na podstawie jej graficznego obrazu w tablicy typu Veitcha-Karnaugh. Kraków 1967. Pr. IV Kraj. Konf. Automat. t. 2 s. 29-42.
3. Recognition of total or partial symmetry in a completely or incompletely specified switching function. Proc. IV Congr. IFAC Warszawa June 16-21 1969. Vol. 27. pp.109-129.
4. O quasi-minimalnym rozwiązaniu problemu pokrycia. Arch. Automat. i Telemekhan. 1969 z. 4.
5. On the quasi-minimal solution of the general covering problem. Proc. Vth Yugoslav Intern. Symp. on Information Processing, Bled October 8-11 1969 pp. 125-128.
6. Minimalizacja funkcji logicznych za pomocą metody pokrycia. Zesz. nauk. Politech. Śląsk. 1969 Automatyka z. 14.

#### DODATEK

Dla ilustracji rozważań podanych w pktach 3.7 i 5.1.6 pokażemy przykład zastosowania algorytmu  $A^q$  do syntezy pokrycia obrazu funkcji  $f = S_{0,1,3,4,5,7,8}^8$ .

Zadana funkcja  $f$  jest funkcją całkowicie symetryczną, mając maksymalną liczbę prostych implikantów wśród funkcji 8 zmiennych, tj.  $p(8) = 576$ . Liczba elementów w zbiorze  $F^1$  wynosi:

$$\sum_{i=0,1,3,4,5,7,8} c_8^i = 200$$

a w zbiorze  $F^0$

$$\sum_{i=2,6} c_8^i = 56$$

Tablica prostych implikantów dla tej funkcji ma zatem rozmiar 576 x 200.

Obraz  $T(f)$  funkcji przedstawiono na rys. D1.

Zgodnie z wynikami pktu 5.1.5, gwiazdy pół mających w obrazie  $T(f)$  wagę 1 ( $C_8^1 = 8$  pół) oraz wagę 7 ( $C_8^7 = 8$  pół) są jednoelementowe, zatem zespoły będące ich elementami są zespołami jądrowymi. Jądro pokrycia  $M^0$  składa się więc z zespołów  $L(\alpha_i)$ ,  $i = 1, 2, \dots, 16$ , gdzie

$$\alpha_i \in \tilde{S}_0^7 \cup \tilde{S}_7^7$$

$$\tilde{S}_0^7 = \{ \bar{x}_1 \bar{x}_2 \dots \bar{x}_7, \bar{x}_1 \bar{x}_2 \dots \bar{x}_6 \bar{x}_8, \bar{x}_1 \bar{x}_2 \dots \bar{x}_5 \bar{x}_7 \bar{x}_8, \dots, \bar{x}_2 \bar{x}_3 \dots \bar{x}_8 \}$$

$$\tilde{S}_7^7 = \{ x_1 x_2 \dots x_7, x_1 x_2 \dots x_6 x_8, x_1 x_2 \dots x_5 x_7 x_8, \dots, x_2 x_3 \dots \dots x_8 \}$$

Zespoły  $L(\alpha_i)$  są zespołami o rozmiarze 1, powstałymi przez połączenie pola  $e^0$  z kolejnymi polami o wadze 1 ( $i = 1, 2, \dots, 8$ ) oraz przez połączenie pola  $e^{255}$  z kolejnymi polami o wadze 7 ( $i = 9, 10, \dots, 16$ ). Pola o wadze 1 oraz 7 zaznaczono na rys. D1 znakami  $\oplus$  i ponumerowano kolejno, według wzrastających numerów tych pól, liczbami 1, ..., 16. Jeżeli przyjmiemy, że zespół pokrywający pole oznaczone liczbą 1 oznacza zespół jądrowy  $L(\alpha_1)$ , to zespoły  $L(\alpha_i)$ ,  $i = 1, 2, \dots, 16$ , będą odpowiadały kolejnym koniunkcjom najpierw zbioru  $\tilde{S}_0^7$  a następnie zbioru  $\tilde{S}_7^7$ .

Po włączeniu jądra obrazu do zbioru  $F^*$  otrzymujemy obraz  $T^0(f)$ , w którym zbiór  $F^1$  składa się z pól o wagach 3, 4 i 5 (łącznie

$\sum_{i=3,4,5} C_8^i = 182$  pola). Gwiazdy tych pól mają odpowiednio  $C_5^3 = 10$ ,

$C_4^3 C_4^3 = 16$  i  $C_5^3 = 10$  elementów (zob. pkt 5.1.4).

Obraz  $T^0(f)$  jest więc cykliczny, a odpowiadająca mu cykliczna tablica prostych implikantów ma rozmiary 560 x 182, przy czym w  $C_8^3 + C_8^5 = 112$  kolumnach ma 10 elementów, a w  $C_8^4 = 70$  ma 16 elementów.

Ponieważ żadna ze znanych technik redukcyjnych nie umożliwia uproszczenia takiej tablicy, przeto, aby określić pokrycie minimalne obrazu  $T^0(f)$ , należy zgodnie z metodą Petrica określić funkcję  $p(3.9)$  i następnie wymnożyć logicznie przez siebie 112 czło-

nów dysjunkcyjnych 10-literowych oraz 70 członów 16-literowych wykonując w toku tego wymnażania operacje pochłaniania. Następnie należy wśród otrzymanych składników określić drogą przeglądu ten, który odpowiada pokryciu minimalnemu (gdyby nie dokonywać operacji pochłaniania w toku wymnażania, wówczas otrzymałoby się  $10^{112} \cdot 16^{70}$  składników (1). Mimo że trudno ocenić ściśle liczbę koniecznych do wykonania operacji, wydaje się jednak, że wykonanie tych obliczeń znacznie przekracza możliwości nawet najszybszych maszyn matematycznych.

Zastosujemy do określenia pokrycia obrazu  $T^0(f)$  algorytm  $A^q$  zmodyfikowany zgodnie z pktm 5.1.6.

Wykorzystując wyniki pktu 5.1.4 określamy zbiory  $F_j^x$  pól, których gwiazdy są rozłączne:  $F_1^x$  - zbiór pól o wadze 3 i  $F_2^x$  - zbiór pól o wadze 5. Są one równoliczne:  $c(F_1^x) = c(F_2^x) = 56$ . Wybieramy zbiór  $F_1^x$ . Z gwiazd pól tego zbioru wybieramy kolejno po jednym zespole maksymalnym dążąc do tego, by pokryć nimi cały zbiór  $F_1^x$ . Otrzymane w ten sposób pokrycie (po wykonaniu różnych prób) zaznaczono na rys. D1. Pola zbioru  $F_1^x$  oznaczono na rysunku znakiem  $\oplus$  i ponumerowano kolejno 17, 18, ..., 72. Otrzymane pokrycie.

$$M^q(T) = M^0 \cup M^c,$$

gdzie

$$M^0 = \bigcup_{i=1}^{16} L(\alpha_i) - \text{jądro pokrycia, } M^c = \bigcup_{i=17}^{72} L_i - \text{pokrycie obra-}$$

zu cyklicznego  $T^0(f)$ ,  $L_i$  - zespół maksymalny wybrany z gwiazdy pola oznaczonego liczbą  $i$  zespół pokrywający to pole zaznaczony na rysunku .

Ponieważ z każdej gwiazdy pola zbioru  $F_1^x$  wybrano tylko jeden zespół maksymalny, a złożoności wszystkich zespołów maksymalnych dowolnej gwiazdy w obrazie  $T(f)$  są jednakowe, przeto  $\Delta = 0$  oraz  $\delta = 0$ . Otrzymane pokrycie  $M^q(T)$  jest więc pokryciem ściśle minimalnym.

#### LITERATURA

1. Gordon E.B., House R.W., Lechler A.P., Nelson L.D., Rado T.:

- Simplification of the covering problem for multiple output logical networks, IEEE trans. on Electronic Computers, vol. EC-15, pp. 891-897, December 1966.
2. Gimpel J.F.: The minimization of TANT networks, IEEE Trans. on Electronic Computers, vol. EC-16, pp. 18-38, February 1967.
  3. Dertouzos M.L., Threshold logic: A synthesis approach, the M.I.T. Press, Cambridge, Massachusetts, 1965.
  4. Roth J.P., Gagner E.G.: Algebraic topological methods for the synthesis of switching systems, Part III: Minimization of non-singular Boolean trees, IBM Journal of Research and Development, vol. 3, No 4, pp. 326-344, October 1959.
  5. Lawler E.L.: Electrical Assemblies with a Minimum Number of Interconnections, IRE Trans. on Electronic Computers, vol. EC-11, February 1962.
  6. Quine W.V.: The problem of simplifying truths functions, The American Mathematical Monthly, vol. 59, pp. 521-531, October 1952.
  7. Petrick S.R.: A direct determination of the irredundant forms of a Boolean function from the set of prime implicants, AFGRCTR-56-110, Air Force Cambridge, Research Center, Cambridge, Mass. April, 1956.
  8. Żurawlew J.I.: Ocena dla czysza tupikowych DNF funkcij algebry logiki, Sibirskij matematicheskiy žurnal. No 5, 1962.
  9. McCluskey E.J.: Minimization of Boolean functions, Bell Syst. Technical Journal, vol. 35, pp. 1417-1444, November, 1956,
  10. Pyne B., McCluckey E.J., Jr.: An essay on prime implicant tables, J. of the Society for industrial and applied mathematics, vol. 9, pp. 604-632, December, 1961.
  11. Pyne I.B., McCluckey E.J., Jr.: The reduction of redundancy in solving prime implicant tables, IEEE Trans. on Electronic Computers, vol. EC-11, pp. 473-482, August, 1962.
  12. Gimpel J.F.: A reduction technique for prime implicant tables, Princeton University, Department of Electr. Sys. Lab. Techn. Report No 40, August, 1964.
  13. Luccio F.: A method for the selection of prime implicants, IEEE Trans. on Electronic Computers, vol. EC-15, pp. 206-212, April 1966.
  14. Lawler E.L., Covering problems: duality relations and a new

- method of solution J.SIAM Appl. Math., vol. 14, No. 5, pp.1115-1132, September, 1966.
15. Friedman A.D.: Comment on "A method for the selection of prime implicants", IEEE Trans. on Electronic Computers, vol. EC-16, pp. 221-222, April, 1967.
  16. Robinson III S.U. and House R.W.: Gimpel's Reduction Technique Extended to the Covering Problem with Costs, IEEE Trans. on Electronic Computers, vol. EC-16, pp. 509-514, August, 1967.
  17. Gomory R.E.: All-- Integer programming algorithm, IBM Research Rept. RC-189, January, 1960.
  18. Cobham A., Fridshal R. and North J.H.: An application of linear programming to the minimization of Boolean functions, Proceedings of the Second Annual Symposium on Switching Theory and Logical Design, Detroit 1961.
  19. Cobham A., Fridshal R., and North J.H.: A statistical study of the minimization of Boolean functions using integer programming, I.B.M. Research Report RC-756, I.B.M. Corp. Yorktown Heights, New York, June 21, 1962.
  20. Cobham A. and North J.H.: Extensions of the integer programming approach to the minimization of Boolean functions, I.B.M. Research Report RC-915, I.B.M. Corp., Yorktown Heights, New York, April 2, 1963.
  21. Majstrowa T.E.: Liniejnoje programmirowanie i zadacza minimizacji normalnych form bulewych funkcij, Problemy pieredaczi informacii, wyp. 12, Izd. AN SSSR, Moskwa 1963.
  22. Gawriłow M.A.: Sowriemiennoje sostojanie teorij rielejnych ustrojstw i koniecznych awtomatow, Teorija koniecznych i wierojatnostnych awtomatow, Trudy IFAK, Izd. "Nauka", Moskwa 1965.
  23. Meo A.R.: On the synthesis of many variable switching functions, Instituto Electronico Nazionale G. Ferraris, Torino, Italy, 1966 (preprint).
  24. House R.W., Nelson L.D., Rado T.: Computer studies of a certain class of linear integer problems, Recent Advances in Optimization Techniques, (Lavi A. Vogl T. Eds.), Wiley, pp. 241-280, New York 1966.
  25. Wasiliew J.Z.: Trudnostiminimizacii bulewych funkcij na osnowie uniwersalnych podchodow, Doklady AN SSSR, Tom 171, No 1, 1966.

26. Maych B.H.: On the simplification of logical expressions, SIAM Journal on Appl. Math., vol. 15, No 4, pp. 898-905, July 1967.
27. Kerntopf P.: Aktualne problemy i osiagnięcia automatycznego projektowania sieci logicznych, Materiały Sympozjum "Automatyczne projektowanie maszyn cyfrowych", Warszawa, 27-29 czerwca 1968.
28. Żurawlew J.I.: O niemożności postrojenia minimalnych дизjunktywnych normalnych form funkcji algebry logiki w adnom klasie algorytmow, Dokłady AN SSSR, t. 132, No 3, ss.504-506, 1960.
29. Roth I.P., Algebraic topological methods for the Synthesis of switching systems, Trans. of the Americ. Mathematic. Society, vol. 88, No 2, pp. 301-326, July, 1958.
30. Kynczew P.K.: Metod za pokuczawanie na czastna minimalna forma na jedna rielejna funkcija ot odszczata minimalna forma, Techniczeska mysł, 11-17, No 4, 1966 (w języku bułgarskim).
31. Kukinow A.M.: O odnom metode sintieza normalnych form, Problemy sintieza cyfrowych avtomatow, ss. 73-83, Izd. "Nauka", Moskwa 1967.
32. Necula N.N.: An algorithm for the automatic approximate minimization of Boolean functions, IEEE Trans. on Electronic Computers, vol. EC-16, pp. 770-782, August, 1968.
33. Wasiliew J.Z.: O sravnienii složnosti tupikowych i minimalnych дизjunktywnych normalnych form, Sbornik "Problemy kibiernetiki", wyp. 10, ss. 5-61, 1963.
34. Żurawlew J.I.: Teorietiko-mnożestwiennyje metody w algerbrie logiki, Sbornik "Problemy kibiernetiki", wyp. 8, ss.5-44, 1962.
35. Roth I.P.: Algebraic topological methods in synthesis, Proceedings of an International Symposium on the Theory of Swithing, April 2-5, 1957, in Annals of Computation Laboratory of Harvard University, vol. 29, pp. 57-73, 1959.
36. Żurawlew J.I.: O postrojenii minimalnych дизjunktywnych normalnych form dla funkcij algebry logiki, Dokłady AN SSSR, t. 126, No 2, 1959.
37. Żurawlew J.I.: Ob algorytmach uproszczenija дизjunktywnych normalnych form dla funkcij algebry logiki, DAN 132, No 2, ss. 260-263, 1960.

38. Żurawlew J.I.: Algoritmy s koniecznoj pamiatju pod dizjunktivnymi normalnymi formami, Trudy Instituta matematiki SO AN SSSR. Sbornik "Diskrietnyj analiz", wyp. 1, ss. 5-12, 1963.
39. Shannon C.E.: The synthesis of two - terminal switching circuits Bell. Syst. Techn. J., vol. 28, pp. 59-98, January, 1949.
40. Lupanow O.B.: O sintiezie kontaktnych schiem, DAN 119, No 1, ss. 23-26, 1958.
41. Burkhart W.H.: Theorem of minimization, Proceedings of the ACM Conference, No 2-3, 1952.
42. Siwiński J.: Zkłady przełączające w automatyce, WNT, Warszawa 1968.
43. Żurawlew J.I.: O rozlicznych poniatkach minimalnosti DNF, Sibirskij matematičeskij žurnał, No 4, 1960.
44. Lin Sin-Lian: O srawnienii složnostiej minimalnych i kratczaj-szych dizjunktivnych normalnych form, Sbornik, "Problemy ki-niernietiki", wyp. 8, ss. 11-14, Moskwa 1967.
45. Synthesis of electronic computing and control circuits. Staff of the Harvard Computation Laboratory. Harvard University Press, Cambridge Mass., chap. 5, 1951.
46. Florine J.: La synthèse des machines logiques et son automatisien, Presses Académiques Européennes, Bruxelles 1964.
47. Svoboda A.: Ordering of implicants, IEEE Trans. on Electric Computers, vol. EC-16, pp. 100-104, February, 1967.
48. Harris B.: An algorithm for determining minimal representations of a logic function, IRE Trans. on Electronic Computers, Vol. EC-6, pp. 103-108, June, 1957.
49. McCluskey E.I., Jr.: Minimal Sums for Boolean Functions having many unspecified fundamental products, Princeton University, Technical Report, No 12, June, 1961.
50. Kazakow W.D.: O minimizacji logiczieskich funkcij bolszogo czisła pieriemiennych, Avtomatika i Tieleniechanika, No 9, 1962.
51. Tison P.: Generalization of consensus theory and application to the minimization of Boolean functions, IEEE Trans. on Electronic Computers, vol. EC-16, pp. 446-456, August, 1967.
52. Kazakow W.D.: Nachoźdienie maksimalnowo czista prostych implikantów proizwolnoj logiczeskoj funkcji n pieriemiennych, Automaticzeskoje uprawlenie, Izd. AN SSSR, ss. 330-338, Moskwa 1959.

53. Urbano R.H., Mueller R.K.: A topological method for the determination of the minimal forms of a Boolean Function. IRE Trans. on Electronic Computers, vol. EC-5, pp. 126-132, September 1956.
54. Ghazala M.I.: Irredundant disjunctive and conjunctive forms of a Boolean function, IBM Journal of research and development, vol. 1, pp. 171-176, 1957.
55. Howard R.L.: New methods of simplifying Boolean functions, AIEE Trans. part 2, Applic. and Industry, July, 1959.
56. Gawrzyłow M.A.: Minimizacija buliewych funkcij charakterizujuszczich relejnyje cepl, Avtomatika i Tielemechanika, No 9, 1959.
57. Butler K.I. Jr., Warfield I.N.: A digital computer program for reducing logical statements to a minimal form; Proceedings of the national electronics conference, vol. XV, pp. 456-466, October 12-14, 1959.
58. Quine W.V.: On cores and prime implicants of truth functions, Americ. Math. Monthly, vol. 66, pp. 755-760, November, 1960.
59. Stogij A.A.: O minimizacii buliewych funkcij na elektronnych cifrowych wycislitielnych maszinach, Woprosy Wycislitelnoj Tehniki, ss. 65-70, 1961.
60. Waligórski S.: Calculation of the Quine's table for truth functions, Prace Zakładu Aparatów Matematycznych PAN, Praca A 15, Warszawa, t. II, 1961.
61. Hockney R.: An intersection algorithm giving all irredundant normal forms from a prime implicant list, IRE Trans. on Electronic Computers, vol. EC-11, March, 1962.
62. Scheinman A.R.: A method for simplifying Boolean functions, Bell Sys. Tech. J. vol. 41, pp. 1337-1346, July, 1962.
63. Florine I.: Comparaison des calculateurs universels et des calculateurs spécialisés concernant la simplification automatique des fonctions et des circuits logiques, Presses Universitaires de Bruxelles 1962.
64. Żurawlew J.I.: Ocenka złożoności algorytmów postrojenia minimalnych дизjunktywnych normalnych form dla funkcji algebry logiki, sb. Diskrietnyj analiz, Nowosibirsk, No 3, 1964.
65. Mileto F., Putzolu G.: Average values of quantities appearing in Boolean function minimization, IEEE on Electronic Computers,



- vol. EC-13, pp. 87-92, April, 1964.
66. Mukhopadhyay A.: On determination of the irredundant covers of a prime implicant table, Princeton University, Technical Report, No 36, July, 1964.
  67. Mott H. and Carroll C.C.: Numerical procedures for Boolean function minimization, IEEE Trans. on Electronic Computers, vol. EC-13, August, 1964.
  68. Cheng D.M.Y., Mott T.H., Jr.: Computing irredundant normal forms from abbreviated presence functions IEEE Trans. on Electronic Computers, vol. EC-14, pp. 335-342, June, 1965.
  69. Pospiechowski D.A.: Woprosy minimizacji normalnych form funkcji algebraicznych logiki na sowremiennych wycislitelnych maszynach dyskretnego dzienstwa, Teoria relejnych struktur, Trudy IFAK, Izd. "Nauka", Moskwa 1965.
  70. Natapoff A.: Irreducible topological Components of an arbitrary Boolean truth function and generation of their minimal coverings, Journ. of the Association for computing machinery, vol. 14, No. 2, pp. 376-381, April, 1967.
  71. Niekuzka N.N.: Avtomaticheskaja minimizacija dwuznacznich buliewych funkcij, Avtomatika i Teliemechanika, No 5, ss. 80-85, 1967.
  72. Florin Z.: Optimizacija dwocznnych funkcij c pomoszczju specializirovannoj elektronnej wycislitelnoj masziny, Avtomatika i Teliemechanika, No 6, ss. 112-121, 1967.
  73. Kynczew P.: Metod za nienosredstawieno opriedielinie na czastna minimalna forma na relejni funkcii z otdzieljanie na jadroto, Izwiestia na Instituta po Techniczieska Kibiernietika, Izd. na Bołgarskata Akademia na Naukite, t. VI, ss.15-14, Sofia 1967.
  74. Krisztal B.B.: Minimizacija odnowo klasa simmetriczeskich buliewych funkcij, Teoria diskretnych avtomatow, Riga, 1967.
  75. Wizariew I.: Metod za pouczawanie na czastni minimalni formi na buliewi funkcii ot broj promienliwi, Izwiestia na Instituta po Techniczieska Kibiernietika, Izd. na Bołgarskata Akademia na Naukite, t. VI, ss. 15-26, Sofia 1967.
  76. Gorobowiockaja N.A., Pinsker I.Sz.: Postrojenie minimalnych dziunktiwnych normalnych form, Problemy sintieza cifrowych av-

- tomatów, ss. 54-72, Izd. "Nauka", Moskwa 1967.
77. Zakriewskij A.D., Ostrowskij W.I.: Optimizacja poszuka kratozaj-szewo pokrytia, Problemy sintieza cifrowych awtomatów, ss.84-95, Izd. "Nauka", Moskwa 1967.
  78. Głagoliew W.W.: Niekotoryje ocienki dizjunktivnych normalnych form funkcij algebrj logiki, Problemy Kibiernietiki, wyp.19, ss. 75-94, 1967.
  79. Marquand A.: "A logical diagram for n terms", Philosophical Magazine, vol. 12, p. 266, October, 1881.
  80. Venn I.: "Symbolic logic", 2-nd ed., London 1894, (pp.138-140).
  81. Veitch E.W.: A chart method for simplifying truth functions, Proc. of Association for Computing Machinery, Pittsburgh, Pennsylvania, Meeting May 2-3, pp. 127-133, 1952.
  82. Karnaugh M.: The map method for synthesis of combinational logic circuits, AIEE Trans. part I, Communications and Electronics, vol. 72, pp. 593-599, November, 1953.
  83. Svoboda A.: Grafico - mechanicke pomucky uzirane pri analise a synthese kontaktowych obvodu. Stroje Zpracov. Inf. No 4, 1956 (w jz. czeskim).
  84. Zakriewskij A.D.: Wizualno-matrycznyj mietod minimizacii bu-liewych funkcij, Avtomatika i Tielemechanika, No 3, 1960.
  85. Basu M.S., Das S.R., Choudhury A.K.: A chart method for the determination of one the minimal forms of a switching function, J. of Electronics and Control, No 4, pp. 425-440, April, 1964.
  86. Bromirski J.: Macierzowa metoda minimalizacii funkcii bulowskich 5 i 6 zmiennych. Prace III Kraj. Konfer. Automatyki, t.2, Politechnika Śląska, Gliwice 1966.
  87. Czomarow S.: Wizualno matriczen mietod za minimizacja na logi-cheski schemi, rabotata na kaito sie opisawa s pomoszczta na bu-liewy funkcii, Trudowie na II nacjonalna konferencija po avto-matika, t. I, cz II, ss. 372-382, Warna 21-25.IX.1967.
  88. Michaleki R.S.: Graficzna minimalizacja w klasie alternatyw-nych normalnych wyrażen funkcji logicznych na podstawie tablic typu Veitcha-Karnauga, Prace Instytutu Automatyki PAN, zeszyt 52, 1967.
  89. Michiejew W.M.: O mnożestwach codierzaszczich najboszeje czis-ło poparno niesrawninnych buliewych wektorow, Problemy Ki-biernietiki, wyp. 2, ss. 69-71, 1959.

90. Łupanow O.B.: K woprosu o realizacji simmetriczeskich funkcij algebry logiki kontaktnymi schiemami, Problemy Kibernetiki, wyp. 15, ss. 85-99, 1965.
91. Epstein C.: Synthesis of Electronic Circuits for Symmetric Functions, IRE Trans. on Electronic Computers, vol. EC-7, pp. 60-67, March, 1958.
92. Sheng C.L.: A graphical interpretation of realization of symmetric Boolean functions with threshold logic elements, IEEE Trans. on Electronic Computers, vol. EC-14, pp. 8-18, February 1965.
93. Caldwell S.H.: The recognition and identification of symmetric switching functions, Trans. AIEE, vol. 73, Pt. I, May, 1954.
94. Powanow G.N.: O metodkie analiza simmetriczeskich kontaktnych schiem, Avtomatika i Tielemechanika, t. XVI, No 4, ss. 364-366, 1955.
95. McCluskey E.J., Jr.: Detection of group invariance or total symmetry of a Boolean function, Bell Sys. Tech. J. vol. 35, pp. 1445-1453, November, 1956.
96. Marcus M.P.: The detection and identification of symmetric switching functions with the use of tables of combinations, IRE Trans. on Electronic Computers, vol. EC-5, pp. 237-239, December 1956.
97. Choudhury A.K., Basu M.S.: On detection of group invariance or total symmetry of a Boolean function, Indian J. Phys., vol. 36, pp. 31-42, January, 1962.
98. Mukhopadhyay A.: Detection of total or partial symmetry of a switching function with the use of decomposition charts. IRE Trans. on Electronic Computers, vol. EC-12, pp. 553-557, October 1963.
99. Sheng C.L.: Detection of totally symmetric Boolean functions, IEEE Trans. on Electrical Computers, vol. EC-14, pp. 924-926, December, 1965.
100. Dietmeyer D.L., Schneider P.R.: Identyfication of symmetry, redundancy and equivalence of Boolean functions. IEEE Trans. on Electronic Computers, vol. EC-16, pp. 804-817, December, 1967.
101. Kazakow W.D.: Wld minimalnych form simetricznych buliewych

funkcji proizwolnogo sziala pieremiennych, sb. "Avtomaticzeskoje regulirovanie i upravlenie", ss. 468-473, Izd. AN SSSR, Moskwa 1962.

Р. С. Михальски

СИНТЕЗ МИНИМАЛЬНЫХ ФОРМ  
И РАСПОЗНАВАНИЕ СИММЕТРИИ ЛОГИЧЕСКИХ ФУНКЦИЙ

Резюме

В статье описаны новые концепции решения некоторых существенных вопросов теории комбинационных автоматов: вопроса покрытия при синтезе минимальных ДНФ, проблемы распознавания симметрии (полной или частичной) логических функций и использования информации о симметрии функции для облегчения синтеза их минимальных форм.

Описаны два алгоритма синтеза покрытий:  $A^M$  - дающий точно минимальное покрытие, но требующий иногда осуществления операции перебора и  $A^Q$  - не требующий перебора и дающий т.н. квазимиимальное покрытие вместе с оценкой максимально возможной разницы между полученным и минимальным покрытием.

Для описания разработанных алгоритмов применена двумерная геометрическая модель логической функции т.н. образ функции  $T(f)$ .

R. S. Michalski

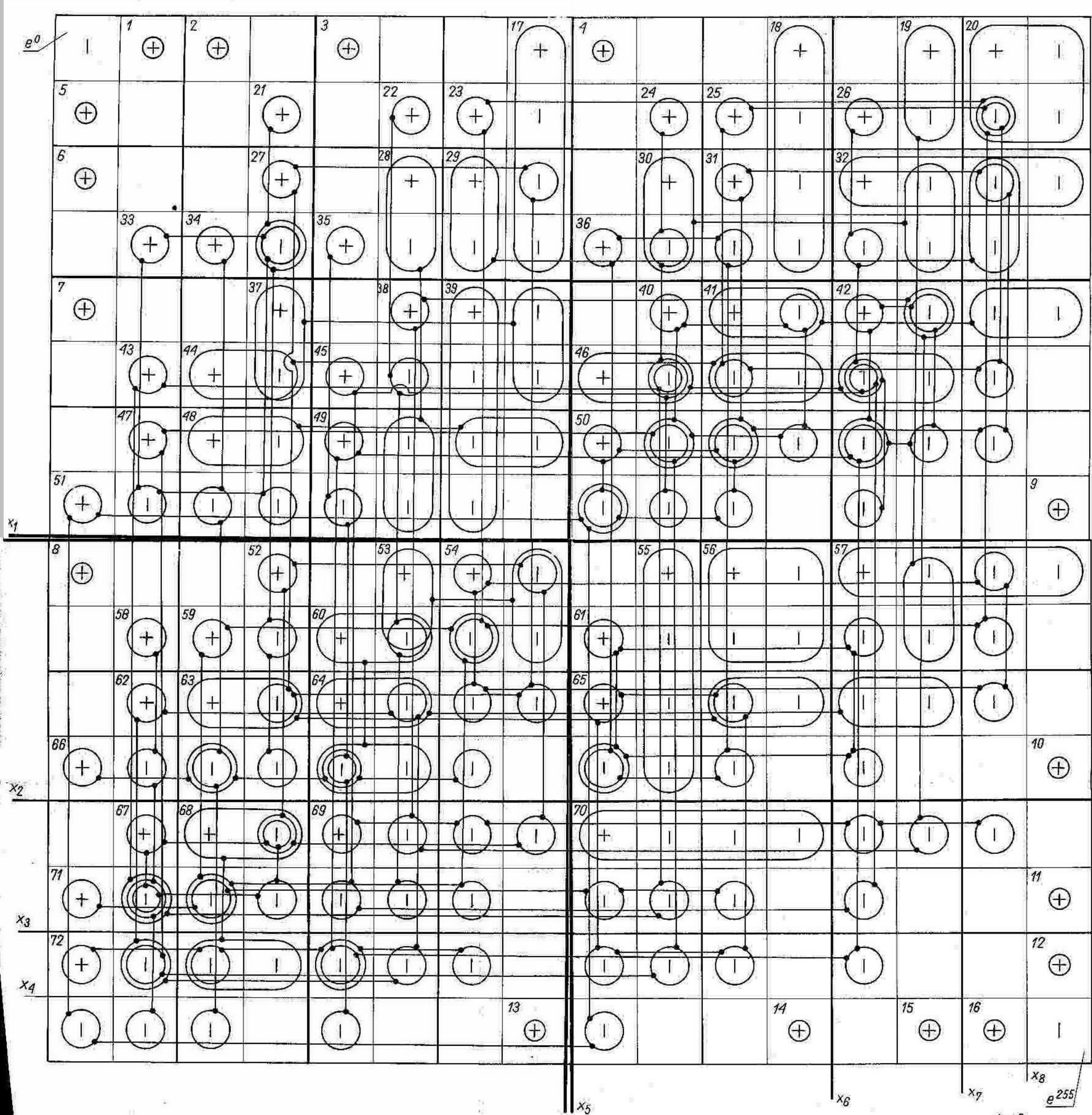
SYNTHESIS OF MINIMAL STRUCTURES OF COMBINATIONAL AUTOMATA  
AND RECOGNITION OF SYMMETRY IN SWITCHING FUNCTIONS

Summary

The paper describes new ideas of resolving some of the significant problems in switching circuits theory: the covering problem connected with the synthesis of minimal forms of switching functions, the recognition of the generally stated symmetry of switching functions and the use of symmetry properties to simplify the synthesis of minimal forms.

Two algorithms for synthesis of a cover are derived:  $A^M$  - which yields a minimal cover, but may require some inspection operations and  $A^Q$  - which does not require inspection and gives a so-called quasi-minimal cover with an estimate of the maximal possible distance to the minimal cover.

As a tool for algorithms was used a two-dimensional geometric model of a switching function, called the image  $T(f)$ .



Rys. D.1. Pokrycie  $M^q(T)$  funkcji  $S_{0,1,3,4,5,7,8}^8$

$\Delta=0$   
 $\delta=0$