

INSTYTUT CYBERNETYKI TECHNICZNEJ
POLITECHNIKI WROCŁAWSKIEJ

Komunikat nr 565

MODEL FORMALNY UCZĄCEGO SIĘ
AUTOMATU PARAMETRYCZNEGO
DŁA POTRZEB SYNTEZY SYSTEMU
OPERACYJNEGO SC

/praca doktorska/

Jan Magott

Promotor:

doc.dr hab.inż. Jan Kazimierczak

Słowa kluczowe: automat uczący się, automat parametryczny, system operacyjny, automat zagregowany, podautomat, teoria grafów

Wrocław 1977

Mgr inż. Jan Magott

Instytut Cybernetyki Technicznej

Politechniki Wrocławskiej

Wrocław, ul. Janiszewskiego 11/17

budynek C-3

Promotorowi - Panu Docentowi
Janowi Kazimierczakowi dziękuję
za inspirację oraz cenne dyskusje
w czasie pisania tej pracy.

SPIS TREŚCI

WSTĘP	6
1. ROLA I MIEJSCE UCZĄCEGO SIĘ AUTOMATU PARAMETRYCZNEGO W SYSTE- MIE OPERACYJNYM SC	15
1.1. Uzasadnienie potrzeby uczenia się systemów operacyjnych SC	15
1.2. Poszukiwanie nowych rozwiązań systemów operacyjnych umoż- liwiających realizację procesu uczenia się	22
1.3. Ogólna charakterystyka koncepcji systemu operacyjnego - - pracującego na zasadach działania automatu parametrycz- nego	26
1.4. Element uczący się w systemie operacyjnym pracującym na zasadzie działania automatu parametrycznego	34
2. MODEL FORMALNY UCZĄCEGO SIĘ AUTOMATU PARAMETRYCZNEGO	40
2.1. Ogólna charakterystyka koncepcji uczenia się automatu pa- rametrycznego	40
2.2. Porównanie przyjętej koncepcji z innymi koncepcjami ucze- nia się	45
2.3. Automaty zagregowane - produkt procesu uczenia się auto- matu parametrycznego	47
2.3.1. Pojęcie automatów zagregowanych i ich klasyfika- cja	47
2.3.2. Synteza formalna automatów zagregowanych pierw- szego rodzaju	49
2.3.3. Własności automatów zagregowanych pierwszego ro- daju	65
2.3.4. Synteza formalna automatów zagregowanych drugiego rodzaju	80
2.3.5. Własności automatów zagregowanych drugiego rodza- ju	95

2.3.6. Proces syntezy formalnej automatu zagregowanego zdolnego do realizacji zadania zgodnie z nowym wa- riantem jego rozwiązania	118
--	-----

3. REALIZACJA ODWZOROWYWANIA AUTOMATÓW ZAGREGOWANYCH W STRUKTU- RZE UCZĄCEGO SIĘ AUTOMATU PARAMETRYCZNEGO	151
ZAKOŃCZENIE	163
LITERATURA	167

W pracy zaproponowano koncepcję uczącego się automatu parametrycznego zdolnego do opanowywania nowych zadań i nowych wariantów rozwiązania tych zadań, do których realizacji automat ten zdolny był przed procesem uczenia się. W celu realizacji nowych zadań lub zadań poprzednio realizowanych zgodnie z nowymi wariantami ich rozwiązania, rozważany automat przeprowadza syntezę automatów nazwanych automatami zagregowanymi. Zasadniczym blokiem tego automatu jest automat parametryczny przedstawiony w pracach [25], [27], oznaczony symbolem AP. Automat AP odwzorowuje swym działaniem wiele automatów, z których każdy można interpretować jako system podautomatów. Stąd automat AP odwzorowuje zbiór podautomatów. Podautomaty te są podstawą do syntezy automatów zagregowanych budowanych w procesie uczenia się. Uczący się automat parametryczny odwzorowuje swym działaniem automaty zagregowane. W pracy przedstawiono podstawy formalne do syntezy strukturalnej tego typu automatu. Zaproponowany automat może wchodzić w skład systemu operacyjnego SC pracującego na zasadzie działania automatu parametrycznego będącego wielozadaniowym systemem sterującym. Wyszczególniono wybrane dziedziny funkcjonowania systemu operacyjnego, w których zakresie problem uczenia się odgrywa istotną rolę.

WSTĘP

W celu zwiększenia efektywności systemów cyfrowych dąży się do zastąpienia pewnych środków programowych systemu operacyjnego - środkami aparaturowymi. Ponieważ system operacyjny SC jest systemem wielofunkcyjnym, uzasadnioną staje się próba jego syntezy bazująca na takim wielozadaniowym organie sterującym jakim jest automat parametryczny. Jedną z funkcji, które powinien wypełniać system operacyjny we współczesnym systemie cyfrowym jest uczenie się. W pracy przedstawiono koncepcję uczącego się automatu parametrycznego.

Współczesne systemy operacyjne systemów cyfrowych nie dysponują, w odpowiednim stopniu, możliwościami rozwoju. Informacje zbierane przez system operacyjny z otoczenia i z systemu cyfrowego służą sterowaniu systemem cyfrowym oraz są podstawą do obliczeń statystycznych. Obliczenia te w zbyt małym stopniu zorientowane są na zastosowanie ich wyników w procesie rozwoju systemu operacyjnego (SO) SC. Systemy operacyjne tego samego typu mogą funkcjonować w różnych otoczeniach. Stąd często w celu określenia optymalnych algorytmów sterowania systemem cyfrowym należy przeprowadzić rozpoznanie otoczenia SC.

Praca składa się z trzech, tematycznie ze sobą powiązanych rozdziałów opatrzonych wstępem i zakończeniem. W rozdziale pierwszym przedstawiono znaczenie i miejsce uczącego się automatu parametrycznego w systemie operacyjnym SC. Rozdział drugi zawiera model uczącego się automatu parametrycznego. W rozdziale trzecim przedstawiono podstawy do syntezy strukturalnej tego automatu.

W podrozdziale 1.1. pracy wymieniono wybrane dziedziny funkcjonowania systemu operacyjnego - godne uwagi ze względu na procesy uczenia się tego systemu.

Ze względu na podobieństwo procesów uczenia się i adaptacji SC, pewne środki umożliwiające realizację procesu uczenia się mogą być zastosowane jako środki umożliwiające adaptację.

Programy systemu operacyjnego obciążają procesor centralny oraz blokują część pamięci operacyjnej. W celu zwiększenia efektywności SC dąży się do urządzeniowej realizacji pewnych funkcji SO dotychczas realizowanych za pomocą oprogramowania. Zapewnienie możliwości uczenia się systemu operacyjnego powoduje zwiększenie rozmiarów tego systemu. Jeśli SO zrealizowany jest programowo, to zwiększenie jego rozmiarów implikuje zwiększenie stopnia zaangażowania zasobów SC przez programy SO. Stąd w celu zwiększenia efektywności systemu cyfrowego sterowanego przez uczący się system operacyjny, uzasadnioną jest próba urządzeniowej realizacji takiego SO.

System operacyjny jest systemem wielofunkcyjnym. Zatem godną uwagi jest próba jego syntezy bazująca na takim wielozadaniowym systemie sterującym, jakim jest automat parametryczny [25], [27]. W celu przeprowadzenia syntezy automatu parametrycznego modelującego system operacyjny, zbiór programów SO rozpatrywany jest jako zbiór zadań realizowanych przez ten automat. Zadaniami mogą być: szeregowanie zadań obliczeniowych, sterowanie wymianą stron między pamięcią operacyjną

a pamięcią zewnętrzną, tłumaczenie programów źródłowych użytkowników itp.

Nie wszystkie algorytmy systemu operacyjnego można bezpośrednio, w możliwy do zastosowania sposób, interpretować jako algorytmy zgodne z automatowym schematem behawioralnym, w myśl którego po podaniu na automat litery alfabetu wejściowego z otoczenia, następuje generacja litery alfabetu wyjściowego. Przykładem algorytmów trudnych do wzmiankowanej interpretacji są sieci działań dla programów. W punkcie 1.3. pracy przedstawiono sposób transformacji sieci działań na graf automatu odwzorowującego tę sieć.

W proponowanej koncepcji funkcjonowania systemu operacyjnego, zadanie realizowane jest zgodnie z przyjętym wariantem rozwiązania tego zadania. Wariant rozwiązania zadania można przedstawić za pomocą grafu. Ponieważ wariant taki jest zwykle systemem podzadań, stąd w jego grafie można wyróżnić grafy podzadań.

Automat parametryczny przedstawiony w pracach [25], [27] - oznaczony symbolem AP, zdolny jest do realizacji wielu zadań zgodnie z przyjętymi, przed procesem uczenia się, wariantami ich rozwiązania. Automat ten nie dysponuje możliwościami uczenia się. W celu realizacji procesu uczenia się systemu operacyjnego, w pracy zaproponowano koncepcję uczącego się automatu parametrycznego. Automat ten zdolny jest do opanowywania nowych zadań oraz do zmiany wariantów rozwiązania tych zadań, do których realizacji zdolny jest automat AP przy pominięciu procesu uczenia się. Stąd można ogólnie stwierdzić, że uczący się automat parametryczny opanowuje nowe warianty rozwiązania zadań.

W celu opanowania nowego zadania lub nowego wariantu rozwiązania zadania budowany jest automat zwany zagregowanym. Synteza tego automatu przeprowadzana jest w uczącym się automacie parametrycznym. Automaty zagregowane zbudowane w procesie uczenia się odwzorowywane są działaniem uczącego się automatu parametrycznego. Zasadniczym blokiem funk-

cjonalnym tego automatu jest automat parametryczny w sensie automatu przedstawionego w pracach [25], [27], będący podstawą do odwzorowywania automatów zagregowanych.

Automat AP odwzorowuje swym działaniem wiele automatów. W pracy automat odwzorowywany działaniem automatu AP interpretowany jest jako system podautomatów. Stąd automat AP odwzorowuje wiele podautomatów. Podautomaty te zdolne są do realizacji podzadań. Współdziałanie wybranych podautomatów jest podstawą do realizacji pewnego zadania, zgodnie z określonym wariantem jego rozwiązania. Automat zagregowany jest systemem podautomatów przynależnych do różnych automatów odwzorowywanych działaniem automatu AP.

Na $(i+1)$ -szym etapie procesu uczenia się, dla nowego zadania lub nowego wariantu rozwiązania takiego zadania, do którego realizacji po „i” etapach zdolny był uczący się automat parametryczny, budowany jest na bazie zbioru podautomatów automat zagregowany zdolny do realizacji nowego zadania lub wymienionego wyżej zadania zgodnie z nowym wariantem jego rozwiązania.

W punkcie 2.2. pracy porównano przyjętą koncepcję uczenia się z innymi koncepcjami uczenia się, ze względu na jej przydatność do zastosowania w systemie operacyjnym SC.

W pracy zdefiniowano automaty zagregowane pierwszego i drugiego rodzaju oraz przedstawiono ich syntezę formalną. Grafy automatów zagregowanych są wynikiem wykonania pewnych operacji na grafach podautomatów odwzorowywanych działaniem automatu AP. Przedstawiono również wybrane własności tych automatów dotyczące ich zachowania i zależności między automatami zagregowanymi obu rodzajów.

W punkcie 2.3.6. pracy przedstawiono proces syntezy formalnej automatu zagregowanego zdolnego do realizacji pewnego zadania zgodnie z nowym wariantem rozwiązania tego zadania.

W rozdziale trzecim pracy rozpatrzono odwzorowywanie automatów

zintegrowanych w uczącym się automacie parametrycznym z zastosowaniem automatu AP. W pracy [24] określono metodę syntezy strukturalnej automatów rozgrywających parametrycznych, która jest zbliżona do metody syntezy automatu AP. Stąd treść tego rozdziału jest podstawą do syntezy strukturalnej uczącego się automatu parametrycznego.

W końcowej części pracy wyszczególniono wybrane problemy dalszych badań, ponieważ praca ta jest pierwszą z zakresu przedstawionej koncepcji uczenia się.

Ważniejsze symbole i definicje

$\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$	- funktory rachunku zdań, odpowiednio, iloczynu, sumy, negacji, implikacji, równoważności
$\forall$	- kwantyfikator ogólny
$\exists$	- kwantyfikator szczegółowy
$a \in A$ ($a \notin A$)	- a jest (nie jest) elementem zbioru A
$\emptyset$	- zbiór pusty
$\cap, \cup, \setminus$	- operacje mnogościowe, odpowiednio, iloczynu, sumy, różnicy
$\bigcup \mathcal{R}$	- suma mnogościowa zbiorów rodziny $\mathcal{R}$
$A \subset B$	- zbiór A jest podzbiorem zbioru B
$\overline{X}$	- moc zbioru X
$\{a : \beta(a)\}$	- zbiór wszystkich elementów a , dla których prawdziwa jest funkcja zdaniowa $\beta(a)$
$\{a\}$	- zbiór, którego jedynym elementem jest a
x	- produkt kartezjański
$\langle a_1, \dots, a_n \rangle$	- n -tka uporządkowana
$\sum$	- suma arytmetyczna
AP	- automat parametryczny w sensie automatu przedstawionego w pracach [25], [27]
UAP	- uczący się automat parametryczny

K_i P_i M_{ji} A_i

$$L = \{ A_i \mid i \in I \}$$

$$G^+ = \{ G_i \mid i \in I \}$$

 S_j
 $\oplus$ T_j
 Δ

$$\{ G_i \mid i \in J \}$$

 A^J G^J

AZI

AZII

 X^* $\bar{x}$ $l|\bar{x}|$

- zadanie
- wartość parametru p
- podzadanie wchodzące w skład zadania K_j
- podautomat odwzorowywany działaniem automatu AP, będący automatem w sensie automatu Moore'a
- zbiór podautomatów odwzorowywanych działaniem automatu AP wyrażonych symbolami, których indeksy "i" są elementami zbioru I
- zbiór grafów podautomatów przynależnych do zbioru L
- operacja wykonywana na grafach przynależnych do podzbioru zbioru G^+ , której wynikiem jest graf automatu zagregowanego pierwszego rodzaju
- operacja wykonywana na grafie będącym wynikiem wykonania operacji $S_j \oplus$. Wynikiem wykonania operacji $T_j \Delta$ jest graf automatu zagregowanego drugiego rodzaju.
- zbiór grafów podautomatów ze zbioru L poddanych operacji $S_j \oplus$ tzn. $J \subset I$
- automat zagregowany
- graf automatu zagregowanego
- automat zagregowany pierwszego rodzaju
- automat zagregowany drugiego rodzaju
- zbiór wszystkich słów skończonej długości z alfabetu X
- skończonej długości sekwencja liter z alfabetu X, tzn. $\bar{x} \in X^*$
- długość sekwencji $\bar{x}$

$E(G)$

- zbiór tych dróg grafu G automatu deterministycznego A , których wierzchołkiem początkowym jest wierzchołek q_0 przyporządkowany stanowi początkowemu tego automatu

 e_j
 $+A^j$

- element zbioru $E(G)$
- automat odwzorowywany działaniem automatu AP , będący automatem Moore'a

 $+G^j$

- graf automatu $+A^j$

 $\bar{G}$

- graf automatu bez wyjścia automatu AP

 $-H^j$

- podgraf cząstkowy grafu $\bar{G}$. Odpowiednie opisanie grafu $-H^j$ cechami funkcjonalnymi umożliwia odwzorowanie grafu $+G^j$.

 H^j

- graf, którego odpowiednie opisanie cechami funkcjonalnymi umożliwia odwzorowanie grafu G^j automatu zagregowanego A^j

 H_i

- podgraf cząstkowy pewnego grafu $-H^j$. Odpowiednie opisanie grafu H_i cechami funkcjonalnymi umożliwia odwzorowanie grafu G_i podautomatu A_i .

 $\mathcal{S}(G, F)$

- zbiór par uporządkowanych, których współrzędne są wzajemnie jednoznacznie przyporządkowanymi sobie wierzchołkami izomorficznych grafów G, F

 $\bar{G}^i$

- graf wariantu rozwiązania zadania

WRZ

- wariant rozwiązania zadania

Automat deterministyczny A z wyjściem - rozpatrywany dalej jako automat Moore'a - można wyrazić następującą "piątką":

$$A = \langle Z, Y, Q, \Phi, \Psi \rangle \quad (0.1)$$

gdzie Z - alfabet wejściowy automatu A , Y - alfabet wyjściowy automatu A , Q - zbiór stanów automatu A , Φ - funkcja przejść automatu A ,

Ψ - funkcja wyjść automatu A.

Jeśli $\langle q_i, z_p, q_j \rangle \in \Phi$, to automat A znajdujący się w stanie q_i , po podaniu na jego wejście litery $z_p \in Z$, osiągnie stan q_j .
Jeśli $\langle q_i, y_s \rangle \in \Psi$, to automat A w stanie q_i generuje literę $y_s \in Y$.

Jeśli automat A ma wyróżniony stan q_0 zwany początkowym, to automat ten można wyrazić następująco:

$$A' = \langle Z, Y, Q, q_0, \Phi, \Psi \rangle \quad (0.2)$$



Graf G automatu A, będący multigrafem skierowanym z obciążeniami wierzchołkami i łukami, można określić następująco:

$$G = \langle Q, \Phi, \Psi \rangle \quad (0.3)$$

gdzie Q - zbiór wierzchołków grafu G, Φ - zbiór trójek uporządkowanych $\langle q_i, z_p, q_j \rangle$ przyporządkowanych łukom grafu G, charakteryzujący obciążenie łuków grafu G literami alfabetu Z, Ψ - zbiór par uporządkowanych $\langle q_i, y_s \rangle$, charakteryzujący obciążenie wierzchołków grafu G literami alfabetu Y.

W celu uproszczenia rysunku grafu G przyjęta zostaje następująca umowa. Jeśli $\langle q_i, z_p, q_j \rangle, \langle q_i, z_s, q_j \rangle, \dots \in \Phi$, to zamiast łuków skierowanych z wierzchołka q_i do wierzchołka q_j , z których jeden opisany jest literą z_p , drugi literą z_s , ..., możemy rysować jeden łuk skierowany z wierzchołka q_i do wierzchołka q_j , opisany następująco:

$$z_p \vee z_s \vee \dots \quad (0.4)$$

Graf H deterministycznego automatu będącego automatem Moore'a bez wyjścia jest multigrafem skierowanym z obciążeniami tylko łukami. Stąd w opisie grafu H nie ma odpowiednika trzeciej współrzędnej trójki określonej wyrażeniem (0.3).

Z rozważanych w pracy grafów, grafy ${}^+G^j, G^j, G_i, \bar{G}^i$ są multi-

grafami skierowanymi z obciążonymi łukami i wierzchołkami, natomiast grafy $\bar{G}$, $\bar{H}^j$, H^j , H_1 są multigrafami skierowanymi z obciążonymi wierzchołkami. W celu uniknięcia definiowania podgrafów i podgrafów cząstkowych multigrafu z obciążonymi łukami i wierzchołkami oraz multigrafu z obciążonymi łukami zostaną przedstawione definicje podgrafu i podgrafu cząstkowego multigrafu skierowanego nieobciążonego.

Niech dany będzie skierowany multigraf nieobciążony:

$$M = \langle V, E \rangle \quad (0.5)$$

gdzie V - zbiór wierzchołków grafu M , E - zbiór łuków grafu M .

Niech symbol $p(x)$, gdzie $x \in E$, oznacza wierzchołek początkowy łuku x , natomiast symbol $k(x)$ - wierzchołek końcowy łuku x .

Definicja 0.1.

Podgrafem skierowanego multigrafu nieobciążonego $M = \langle V, E \rangle$ nazywamy multigraf:

$$M_1 = \langle V_1, E_1 \rangle \quad (0.6)$$

gdzie

$$V_1 \subset V \quad (0.7)$$

$$E_1 = \{x : x \in E \wedge p(x) \in V_1 \wedge k(x) \in V_1\} \quad (0.8)$$

Zbiór E_1 zawiera te łuki zbioru E , których wierzchołek początkowy i wierzchołek końcowy są elementami zbioru V_1 .

Definicja 0.2.

Podgrafem cząstkowym skierowanego multigrafu nieobciążonego $M = \langle V, E \rangle$ nazywamy multigraf:

$$M_2 = \langle V_2, E_2 \rangle \quad (0.9)$$

gdzie

$$V_2 \subset V \quad (0.10)$$

$$E_2 \subset \{x : x \in E \wedge p(x) \in V_2 \wedge k(x) \in V_2\} \quad (0.11)$$

1. ROLA I MIEJSCE UCZĄCEGO SIĘ AUTOMATU PARAMETRYCZNEGO W SYSTEMIE OPERACYJNYM SC

1.1. UZASADNIENIE POTRZEBY UCZENIA SIĘ SYSTEMÓW OPERACYJNYCH SC

Złożony system cybernetyczny powinien dysponować takimi dziedzinami rozwoju jak uczenie się czy adaptacja. Przykładem systemu uczącego się jest perceptron Rosenblatta. Wiele badań poświęconych jest algorytmom rozpoznawania z jednoczesnym uczeniem. Często uważa się, że systemy operacyjne systemów cyfrowych są największymi systemami stworzonymi przez człowieka. Stąd te systemy powinny być zaopatrzone w organy umożliwiające ich rozwój. Niezapewnienie możliwości rozwoju systemu operacyjnego wiąże się z niewykorzystaniem znacznej części informacji jakie można uzyskać z otoczenia.

Dotychczas realizowane systemy cyfrowe nie dysponują w odpowiednim zakresie takimi dziedzinami rozwoju, jak uczenie się czy adaptacja. Rozwój wiąże się z pamiętaniem pewnych informacji, ale jest to proces bardziej złożony od procesu zwykłego zapamiętywania.

W prowadzonych w pracy rozważaniach przyjmuje się, że uczenie się systemu operacyjnego polega na rozwoju tego systemu poprzez zmianę algorytmów jego działania, przy zadanej strukturze fizycznej SC. Natomiast adaptacja jest rozwojem występującym po zmianie tej struktury.

Jedną z funkcji systemu operacyjnego jest zbieranie informacji dotyczących czasu użytkowania poszczególnych podsystemów systemu cyfrowego przez użytkowników. Informacje te służą głównie rozliczaniu z użytkownikami, natomiast w nieznacznym stopniu stosowane są w procesie rozwoju systemu operacyjnego. Wskazuje to na zbyt niski stopień zbadania problematyki rozwoju systemów cyfrowych.

Systemy operacyjne tego samego typu mogą funkcjonować w różnych otoczeniach. Stąd w pewnych przypadkach nie jest możliwe podanie optymalnych algorytmów sterowania systemem cyfrowym bez analizy jego oto-

czenia. W trakcie pracy systemu cyfrowego należy przeprowadzać rozpoznawanie otoczenia SC, mające na celu optymalizację lub budowę nowych algorytmów systemu operacyjnego.

Na przykład niech otoczenie generuje zadania z k klas. Każdej z tych klas powinien być przyporządkowany pewien priorytet. Przyporządkowanie to może być określone bez dokładnej znajomości charakterystyk tych klas. Określenie przyporządkowania optymalnego, w sensie pewnego kryterium np. minimum strat na oczekiwanie, wymaga wyznaczenia niezsynchronizowanych charakterystyk. Rozwiązanie, wyznaczone bez wymaganej znajomości charakterystyk klas zadań, może znacznie różnić się od rozwiązania optymalnego. W przypadku wielokrotnej realizacji programów z tych klas można wyznaczyć wymagane charakterystyki omawianych klas zadań. Niewykorzystywanie informacji, które można w tym przypadku uzyskać, może powodować nieoptymalne wykorzystanie SC.

Innym przykładem wskazującym na celowość zbierania pewnych informacji dla potrzeb rozwoju SC jest optymalizacja algorytmów dostępu do zbioru. Niech dany będzie zbiór składający się z wielu podzbiorów, parami rozłącznych. W celu odczytania informacji użytkowej rekordu o zadanym kluczu, na pierwszym etapie procesu szukania stosuje się technikę randomizacji. Technika ta określa sposób transformacji klucza w numer podzbioru. Jeśli w podzbiorze o numerze określonym za pomocą techniki randomizacji nie zostanie znaleziony rekord o zadanym kluczu, to problem szukania rozwiązywany jest z wykorzystaniem procedury rozwiązywania kolizji. Procedura ta określa sposób generowania numerów kolejno przeszukiwanych podzbiorów. Rozwiązanie omawiane w pracy [16] uzyskane jest przy silnym założeniu. W myśl tego założenia prawdopodobieństwo przynależenia danego klucza do dowolnego zbioru kluczy jest jednakowe. W zależności od otoczenia w jakim funkcjonuje SC, pewne rekordy wykorzystywane są częściej niż inne. Na przykład w translatorze będącym interpretatorem, pewne fragmenty tego translatora stosowane są w procesie translacji częściej niż inne. Wywołane to jest różną

częstością występowania instrukcji w programach generowanych przez dane otoczenie. Trafnym dostępem do podzbioru zostaje nazwany dostęp, w którego wyniku zostaje znaleziona szukana informacja.

Uzyskane na podstawie obserwacji informacje dotyczące liczb trafnychostępów do poszczególnych podzbiorów rozważanego zbioru będą bazą do szeregowania podzbiorów w procesie szukania rekordu o zadanym kluczu. Podzbiory o większej liczbie trafnychostępów będą przeszukiwane przed podzbiarami o mniejszej liczbie trafnychostępów. W wyniku powyższego ulegnie zmniejszeniu średni czas dostępu do rekordu o zadanym kluczu.

Kolejną dziedziną funkcjonowania SC dla optymalizacji systemu operacyjnego jest segmentacja programów. Różne programy mogą zawierać identyczne segmenty. Okazuje się, że nie zawsze warto odwzorowywać w pamięci operacyjnej identyczny segment w każdym z programów, w których on występuje. Pewne segmenty wspólne dla wielu programów, wykonywanych równocześnie w systemie z podziałem czasu, mogą być zapamiętane w pamięci operacyjnej tylko jeden raz. Wyznaczenie tych segmentów wymaga wielu informacji. Segmenty stosowane przez wielu użytkowników można podzielić na dwa typy. Segmentami użytkowanymi sekwencyjnie nazywane są te segmenty, które mogą być stosowane wielokrotnie, ale kolejne ich wykonanie może być rozpoczęte po zakończeniu danego wykonania. Wzmiankowany sposób użytkowania segmentu realizuje system operacyjny. W celu zwiększenia efektywności systemów wieloprogramowych pewne segmenty powinny być użytkowane równolegle. Niekiedy dopiero po poznaniu otoczenia SC można określić, w jaki sposób powinien być użytkowany dany segment, w celu zwiększenia stopnia efektywności SC.

Istotnym zadaniem systemu operacyjnego jest sterowanie procesami realizowanymi przez SC. Jedną z możliwych zmian stanu procesu jest przejście ze stanu aktywnego w stan zablokowania. Jeśli nawet proces jest długo blokowany, to wiele jego procedur znajduje się w pamięci operacyjnej. Ponieważ procedury te w czasie zablokowania procesu nie

są stosowane, a więc niepotrzebnie zajmują obszary pamięci operacyjnej. Godnym uwagi jest rozpatrzenie możliwości usuwania pewnych rozważanych procedur z tej pamięci, w celu częściowego jej oswobodzenia dla potrzeb innych procesów. Poprzez obserwacje SC i jego otoczenia można szacować czas zablokowania poszczególnych procesów. Uzyskane informacje mogą być podstawą do usuwania z pamięci operacyjnej procedur związanych z długo zablokowanymi procesami. W czasie zablokowania proces może być pozbawiony części innych przydzielonych mu zasobów, które wtedy mogą być użytkowane przez inne procesy.

W przypadku podziału pamięci na strony, część stron programów równocześnie wykonywanych znajduje się w pamięci operacyjnej. Często okazuje się, że w pamięci operacyjnej nie ma strony wymaganej przez aktualnie aktywny proces. Powoduje to zablokowanie tego procesu. Stąd pojawia się konieczność przesłania wymaganej strony z pamięci zewnętrznej do pamięci operacyjnej na miejsce innej strony. Należy zatem dokonać wyboru strony, która będzie usunięta z pamięci operacyjnej. Powinna to być strona o najmniejszym prawdopodobieństwie użytkowania jej w najbliższym czasie. Poprzez rozpoznanie programów można wyznaczyć stronę, którą należy wyrzucić z pamięci operacyjnej. Informacje uzyskane w wyniku wielokrotnego wykonania programu dla różnych danych mogą być podstawą do wyznaczenia prawdopodobieństw przejścia od realizacji określonej strony do realizacji pozostałych stron programu. Wzmiankowane prawdopodobieństwa są podstawą do wyznaczenia strony, dla której prawdopodobieństwo tego, że w najbliższym czasie będzie przetwarzana jest najmniejsze.

Poprzez rozpoznanie programu można również określić strony, dla których prawdopodobieństwo ich użytkowania w najbliższym czasie jest największe. Strony te mogą być wprowadzane do pamięci operacyjnej możliwie wcześnie, aby ich brak nie spowodował zablokowania aktywnego

W celu optymalizacji funkcjonowania tego systemu diagnozującego, należy zapewnić możliwość doskonalenia algorytmów diagnostycznych przez system operacyjny. Obserwacja funkcjonowania SC umożliwia określenie częstości występowania poszczególnych błędów. Uwzględnienie tych częstości może być podstawą do formułowania algorytmów diagnostycznych - zapewniających zmniejszenie średniego czasu lokalizacji przyczyny błędu.

Obecnie w zakres systemu operacyjnego często włącza się translatory

Godną uwagi jest optymalizacja kompilatora w wyniku poznawania otoczenia SC. Wzmiankowana optymalizacja może być przeprowadzona w celu:

- 1) zmniejszenia obszaru pamięci zajmowanego przez program wynikowy napisany w języku maszynowym
- 2) szybszego wykonania programu wynikowego dzięki jego odpowiedniej budowie.

Programista, w większości przypadków, w małym stopniu zna maszynę wykonującą napisany przez niego program. Stąd program napisany przez niego dla rozwiązania zadanego problemu może mieć budowę wymagającą wykonania tego programu w czasie znacznie większym niż czas realizacji takiego programu dla rozwiązania danego problemu, którego autorem jest osoba znająca maszynę. Kompilator dysponujący możliwością uczenia się powinien zbierać informacje dotyczące takich niewłaściwych rozwiązań w zadaniach generowanych przez użytkowników, które powodują niepotrzebne zwiększenie czasu realizacji tych zadań. Informacje te mogą posłużyć jako podstawa do budowy optymalniejszych, ze względu na czas wykonania, programów w języku maszynowym. Optymalizacja kompilatora wiąże się z dodatkowymi kosztami, takimi jak zwiększenie czasu kompilacji i złożoności kompilatora. Różne nieoptymalne rozwiązania programistów występują z różną częstością w ich programach. Stąd pewne algorytmy wchodzące w skład kompilatora dają częściej zmniejszenie czasu wykonania programu wynikowego niż inne.

Częstości występowania niewłaściwych rozwiązań programowych wraz ze stopniem zwiększenia szybkości wykonania programów wynikowych, uzyskanym dzięki algorytmom będącym podstawą do optymalizacji programów wynikowych, mogą być podstawą do wyboru tylko niektórych algorytmów, jako włączanych do kompilatora. Wybrane będą te algorytmy, które dając odpowiedni zysk, nie powodują zbyt dużego rozbudowania kompilatora i zbyt dużego zwiększenia czasu kompilacji.

System operacyjny oprócz optymalizacji algorytmów działania SC powinien dysponować możliwością budowy nowych algorytmów. Na przykład istotnym problemem jest problem konstruowania translatora nowego języka. Niech dany będzie opis nowego języka programowania, dla którego system operacyjny ma zbudować translator. Opis ten zawiera m. in. treści związane z poszczególnymi instrukcjami. Niech dany będzie zbiór translatorów różnych języków wraz z opisami tych języków. Dla zadanej instrukcji nowego języka należy wyznaczyć ciąg instrukcji w języku maszynowym. Poprzez analizę opisów innych języków, system operacyjny może wybrać te instrukcje tych języków, których treść podobna jest do treści wzmiankowanej instrukcji. Z wybranymi instrukcjami związane są pewne podprogramy napisane w języku maszynowym, wchodzące w skład translatorów. Godną uwagi jest próba sformułowania takiego algorytmu systemu operacyjnego, który byłby podstawą do budowy odpowiedniego podprogramu w języku maszynowym dla zadanej instrukcji nowego języka, na bazie podprogramów uzyskanych w przedstawiony wyżej sposób.

Wzmiankowane przykłady uzasadniają celowość gromadzenia informacji o otoczeniu systemu cyfrowego w celu zwiększenia stopnia efektywności funkcjonowania SC w tym otoczeniu. Informacje te powinny być gromadzone pod kontrolą organu sterującego systemem cyfrowego. Organem tym jest system operacyjny. System ten oprócz zdolności gromadzenia informacji o otoczeniu powinien dysponować zdolnością wykorzystania tych informacji w celu optymalizacji lub budowy nowych algorytmów swego działania. Wspomniane modyfikacje wiążą się ze zmianą organizacji syste-

mu operacyjnego. W celu umożliwienia wzmiankowanej zmiany systemu operacyjnego, niezbędne jest rozbudowanie tego systemu o organ umożliwiający tę zmianę. Dla systemu operacyjnego sterującego systemem cyfrowym wbudowanym z określonych elementów fizycznych, organ ten powinien być organem zapewniającym realizację procesu uczenia się. Dla systemu operacyjnego sterującego SC po zmianie struktury fizycznej SC, powinien być to organ zapewniający realizację procesu adaptacji do nowej struktury.

Przyjęte uprzednio określenia uczenia się i adaptacji są bardzo ogólne. Na przykład uczenie się powinno obejmować prognozowanie zorientowane na lepsze wykorzystanie określonej struktury SC w danym otoczeniu.

Rozważmy algorytm wymiany stron. W wyniku rozpoznawania programów należy prognozować, które strony należy wyrzucić z pamięci operacyjnej, aby można było sprowadzać nowe strony. W tym przypadku prognozowanie służy lepszemu dostosowaniu SC o określonej strukturze do otoczenia.

Adaptacja również powinna objąć prognozowanie. Niech do istniejącego SC dołączone będą nowe dyskretne przetworniki informacji. Być może w pewnych przypadkach analiza własności dołączonych urządzeń pozwoli na prognozowanie własności urządzenia, które jako jedno z kolejnych zostanie dołączone do SC. Proces prognozowania może w tym przypadku doprowadzić do lepszej adaptacji SC do nowego przetwornika.

Z powyższego wynika, że prognozowanie powinno być uwzględnione w funkcjonowaniu organu sterującego systemem cyfrowym.

Problematyka uczenia się i adaptacji ze względu na swoją złożoność zawiera elementy heurystyki. Elementy heurystyki są niepożądane w funkcjonowaniu organów sterujących systemów cyfrowych. Stąd też przyjmując heurystyczne algorytmy uczenia się i adaptacji należy określić również i taki algorytm, który w trakcie rozwoju systemu eliminowałby heurystykę z procesów uczenia się i adaptacji.

Rozważania dotyczące uczenia się i adaptacji winny być ze sobą

sprzężone. Każdy z rozważanych procesów rozwoju obejmuje opanowywanie nowych zadań lub opanowywanie optymalniejszych wariantów rozwiązania zadań już realizowanych przez organ sterujący SC. Oznacza to, że pewne metody wykorzystywane w procesie uczenia się mogą być również wykorzystane w procesie adaptacji. Stąd organ systemu operacyjnego umożliwiający zmianę organizacji tego systemu dla potrzeb adaptacji jest organem zawierającym pewne elementy organu umożliwiającego proces uczenia się.

Zapewnienie możliwości uczenia się systemów operacyjnych jest szczególnie istotne w przypadku systemów cyfrowych zrealizowanych w większym stopniu na bazie aparatury niż systemy klasyczne, w których większość algorytmów systemu operacyjnego odwzorowana jest w oprogramowaniu. Systemy w większym stopniu zrealizowane na bazie aparatury początkowo będą zwykle systemami eksperymentalnymi, a zatem powinny dysponować możliwościami modyfikacji i budowy nowych algorytmów.

1.2. POSZUKIWANIE NOWYCH ROZWIĄZAŃ SYSTEMÓW OPERACYJNYCH UMOŻLIWIAJĄCYCH REALIZACJĘ PROCESU UCZENIA SIĘ

Systemem operacyjnym w węższym sensie nazywany jest zbiór programów ułatwiających eksploatację SC wraz z programem ułatwiającym współdziałanie tych programów z innymi składnikami oprogramowania podstawowego.

We współczesnych systemach cyfrowych oprogramowanie podstawowe tworzą następujące części składowe:

- 1) system operacyjny w węższym sensie
- 2) translatory języków programowania
- 3) programy redagowania tekstu
- 4) standardowe programy użytkowe
- 5) standardowe programy usługowe
- 6) środki uruchamiania programów

W pracy będzie stosowane pojęcie systemu operacyjnego w tzw. szerszym sensie. System ten zawiera programy sterujące i przetwarzające.

Programy sterujące stanowią system operacyjny w węższym sensie.

W skład programów przetwarzających wchodzi programy wymienione w punktach 2 - 6.

W celu zwiększenia stopnia efektywności SC dąży się do urządzeniowej realizacji pewnych funkcji, które dotychczas były realizowane za pomocą oprogramowania podstawowego SC. Istnieje obecnie tendencja budowy systemu operacyjnego systemu cyfrowego w postaci odpowiedniego urządzenia. Część programów systemu operacyjnego zawsze znajduje się w pamięci operacyjnej. Niektóre elementy tego systemu sprowadzane są tylko na pewien czas do pamięci operacyjnej.

Oznacza to blokadę tej części pamięci dla programów użytkowników oraz wymaga transmisji między pamięcią zewnętrzną a operacyjną. Translator będący kompilatorem zajmuje w czasie translacji znaczną część pamięci operacyjnej. Rozmiary translatora mogą być tak duże, że w danej chwili w tej pamięci znajduje się tylko jego część. Jeśli programy aktualnie wykonywane napisane są w różnych językach programowania, to może to oznaczać konieczność wymiany translatorów między pamięcią operacyjną a pamięcią pomocniczą. Sytuacja taka powoduje zawieszenie programu na pewien odcinek czasu. Odcinek ten może być duży ze względu na duże rozmiary translatorów.

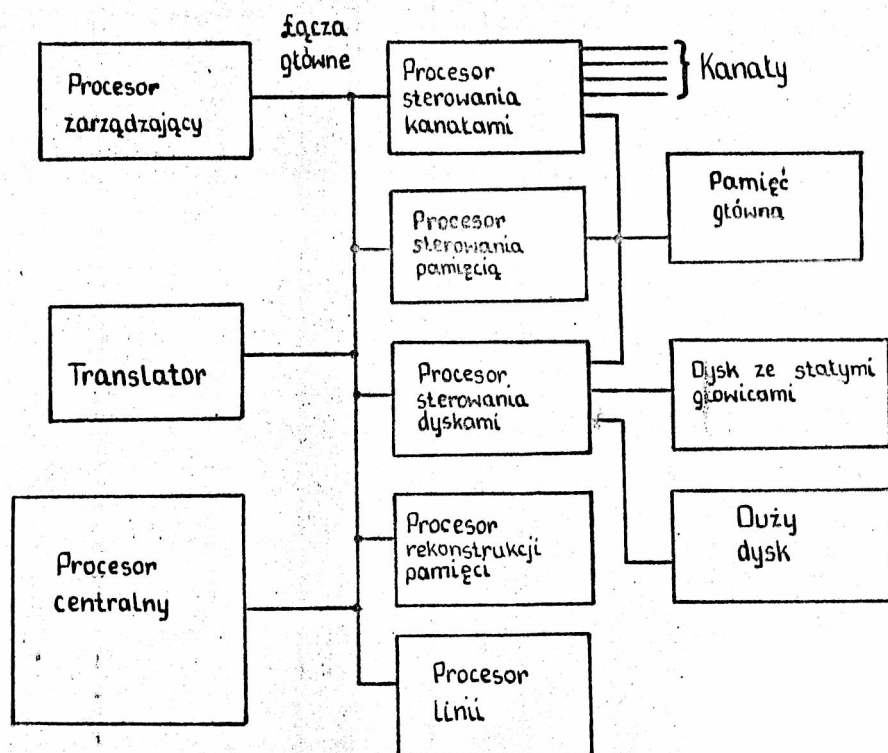
Programy oprogramowania podstawowego dodatkowo obciążają blok arytmetyczno-logiczny. Powoduje to dodatkowe wydłużenie czasu realizacji programów użytkowników. Uzupełnienie oprogramowania podstawowego o środki umożliwiające realizację procesu uczenia się oznacza zwiększenie rozmiarów oprogramowania podstawowego i zwiększenia stopnia zaangażowania bloku arytmetyczno-logicznego realizacją programów tego oprogramowania. Przeniesienie pewnych funkcji oprogramowania podstawowego na odpowiedni układ daje zatem znaczne zwiększenie przepustowości systemu cyfrowego.

System operacyjny jest organem sterującym systemu cyfrowego.

Systemami cybernetycznymi o bardzo wysokim stopniu doskonałości są systemy żywe. W systemach tych organ sterujący posiada pewne cechy elementu fizycznego, natomiast system operacyjny jest systemem programów. Stąd uzasadnione są tendencje zmierzające do zastąpienia systemu operacyjnego odpowiednim układem fizycznym.

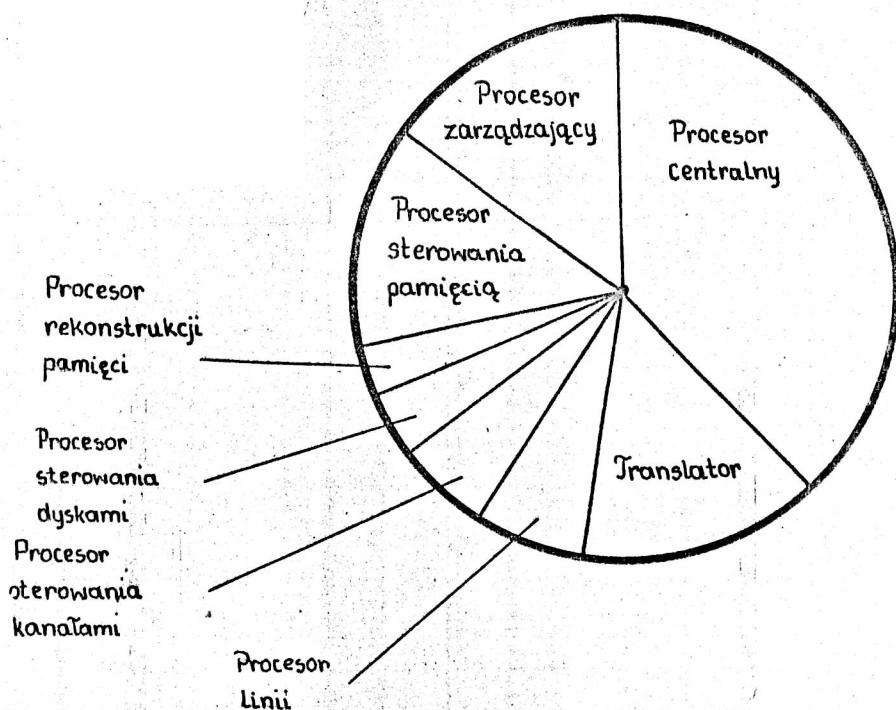
Przykładem systemu cyfrowego, w którym wiele działań wykonywanych jest bez angażowania oprogramowania jest system SYMBOL [46]. Jest to wielki system eksperymentalny dla badania możliwości przeniesienia działań realizowanych z zastosowaniem oprogramowania na aparaturę. System ten jest rezultatem badań zorientowanych na analizę funkcjonalnych możliwości systemów cyfrowych. SYMBOL jest wielodostępnym systemem ogólnego przeznaczenia.

Niżej przedstawiona zostanie ogólna organizacja systemu SYMBOL. System ten zawiera osiem wyspecjalizowanych procesorów. Procesory te połączone są za pomocą łączy zwanych głównymi. Schemat blokowy rozważanego systemu przedstawiony jest na rys. 1.1.



Rys. 1.1. Schemat blokowy systemu SYMBOL

Podstawowa konfiguracja zawiera 18 000 podwojonych mikroukładów tranzystorowych. Względne rozmiary procesorów przedstawione są na rys. 1.2.



Rys. 1.2. Schemat ilustrujący względne rozmiary procesorów systemu SYMBOL

Sterowanie pracą z podziałem czasu.

Procesor zarządzający systemu steruje przejściami między stanami pracy systemu i kolejkami do wszystkich procesorów pracujących z podziałem czasu. Steruje on również organizacją pamięci stronicowanej. Procesor ten steruje transmisjami informacji, niezbędnymi dla łączenia algorytmów zakodowanych w aparaturze z procedurami oprogramowania. Rozważany procesor stosuje tablicę wywołań oprogramowania w celu realizacji zmian stanów systemu. Pewne zmiany stanów systemu realizowane są tylko przez aparaturę - bez wywoływania odpowiedniego oprogramowania. Jako przykład można podać przejście ze stanu ładowania programu do stanu jego kompilacji.

Wnioski.

Według autorów system SYMBOL jest ważnym krokiem na drodze budowy aparatury dysponującej ogromnymi możliwościami. Urządzeniową realizację podziału pamięci i sterowania nią uważają oni za dużą zaletę przedstawionego systemu w porównaniu z klasycznymi systemami cyfrowymi. W pewnych krytycznych przypadkach sterowania pamięcią i jej podziału a także przy podziale zasobów systemu, uzyskana efektywność jest 10 - 100 razy wyższa od zwykłej. Autorzy twierdzą, że pewne aspekty pracy z układami są wielkim osiągnięciem w porównaniu z pracą procesorów pracujących z wykazami programów.

Układ pełniący rolę systemu operacyjnego powinien być dyskretnym przetwornikiem informacji o działaniu zdeterminowanym, a zatem może być interpretowany jako automat deterministyczny. Dalsza analiza prowadzi do wniosku, że jest to automat skończony.

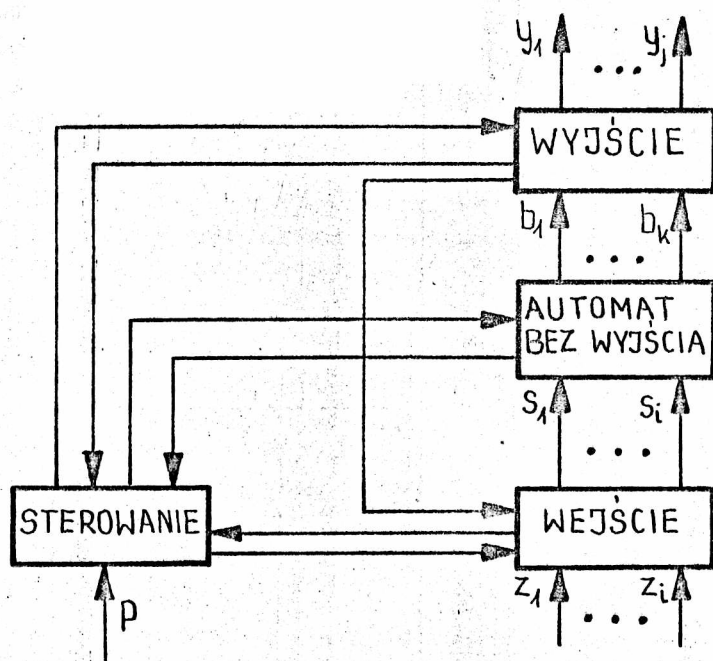
System operacyjny w szerszym sensie jest systemem wielofunkcyjnym. Stąd uzasadnioną wydaje się próba jego syntezy bazująca na takim wielozadaniowym systemie sterującym, jakim jest automat parametryczny [25], [27], będący automatem deterministycznym skończonym.

1.3. OGÓLNA CHARAKTERYSTYKA KONCEPCJI SYSTEMU OPERACYJNEGO - PRACUJĄCEGO NA ZASADACH DZIAŁANIA AUTOMATU PARAMETRYCZNEGO

Schemat blokowy automatu parametrycznego przedstawiony jest na rys. 1.3. a jego działanie i metoda syntezy podane są w pracach [25], [27].

Automat parametryczny zdolny jest do realizacji znacznej liczby zadań. Zbiór tych zadań jest określony następująco:

$$K = \{ K_1, \dots, K_1, \dots, K_n \} \quad (1.1)$$



Rys. 1.3. Schemat blokowy automatu parametrycznego

Każdemu zadaniu K_1 przyporządkowana jest wzajemnie jednoznacznie wartość p_1 parametru p . Po podaniu na wejście automatu wartości p_1 parametru p , automat zdolny jest do realizacji zadania K_1 .

W celu przeprowadzenia syntezy automatu parametrycznego $\langle A \rangle$ modelującego system operacyjny, zbiór K zadań realizowanych przez automat $\langle A \rangle$ traktuje się jako zbiór programów systemu operacyjnego. Zadaniami mogą być: szeregowanie zadań obliczeniowych, sterowanie procesem wyszukiwania informacji systemu, sterowanie wymianą stron między pamięcią operacyjną a pamięcią zewnętrzną, tłumaczenie programów napisanych w określonym języku programowania. Jeśli K jest zbiorem zadań systemu operacyjnego, to na tym zbiorze można określić następującą rodzinę podzbiorów:

$$K^* = \{ K', K^o \} \quad (1.2)$$

gdzie K' - zbiór zadań realizowanych przez $\langle A \rangle$ na korzyść procesora centralnego SC, K^o - zbiór zadań realizowanych przez $\langle A \rangle$ na korzyść pozostałych procesorów lub innych elementów fizycznych struk-

tury SC.

Bloki funkcjonalne struktury SC są elementami następującego zbioru:

$$Q = \{ X_0, X_1, \dots, X_v \} \quad (1.3)$$

gdzie X_0 - system operacyjny w postaci automatu $\langle A \rangle$, X_1 - procesor centralny, $X_2, \dots, X_v$ - pozostałe procesory lub inne bloki funkcjonalne.

W danej chwili dyskretnej stan SC można rozpatrywać jako wektor o postaci:

$$m = \langle m_0, m_1, \dots, m_v \rangle \quad (1.4)$$

Współrzędne wektora m przyporządkowane są stanom poszczególnych bloków $X_j \in Q$. Stanowi SC inicjującemu pracę systemu operacyjnego przyporządkowana jest ściśle określona wartość p_1 parametru p . Zdarzenia występujące w SC (np. przerwania) powodują zmianę stanu SC, co może oznaczać zmianę wartości parametru p . Podanie wartości p_1 parametru p na wejście automatu $\langle A \rangle$ powoduje realizację zadania K_1 (np. obsługiwane przerwania).

Zastosowanie automatu parametrycznego $\langle A \rangle$ jako systemu operacyjnego (SO) posiada tę zaletę względem mniej rozwiniętych systemów operacyjnych, że zwalnia pamięć operacyjną od pamiętania programów SO oraz w znacznym stopniu odciąża procesor centralny od obsługi SO. Działanie automatu $\langle A \rangle$ jako SO polega na tym, że rozkazy programów dotychczasowego SO określone są przez stan automatu bez wyjścia automatu $\langle A \rangle$ i wartość parametru p .

Dokładniejsze przedstawienie modelu systemu operacyjnego działającego na bazie automatu parametrycznego znajduje się w pracy [26].

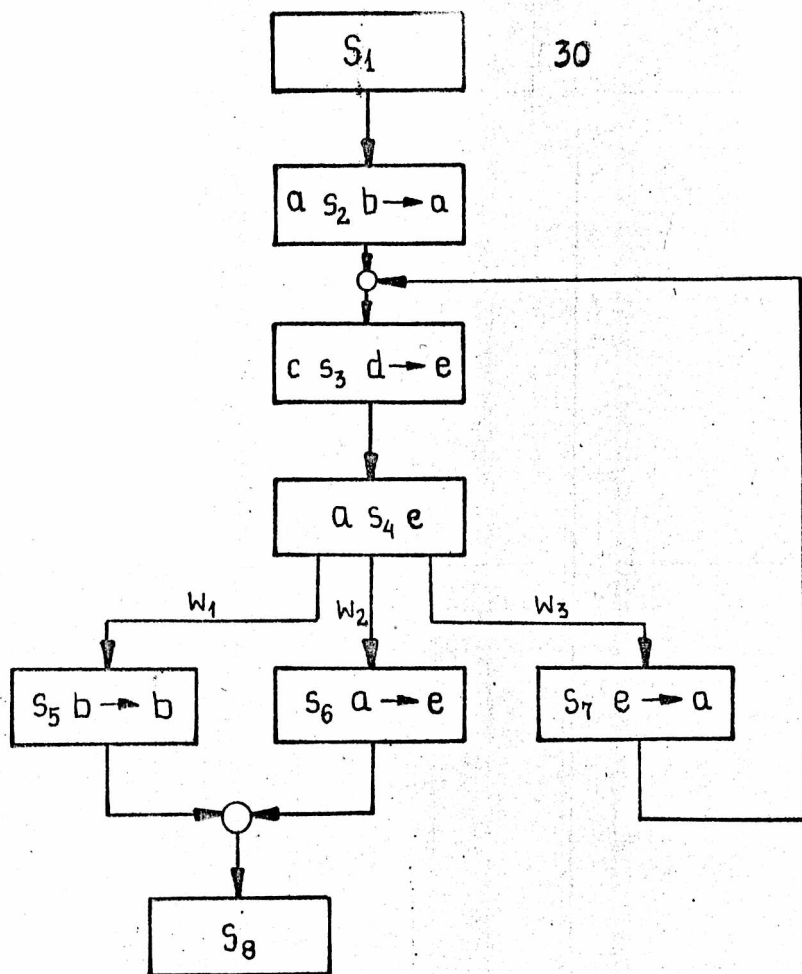
Nie wszystkie algorytmy systemu operacyjnego można bezpośrednio w możliwy do zastosowania sposób interpretować jako algorytmy zgodne

z automatowym schematem behawioralnym, w myśl którego po podaniu na automat litery alfabetu wejściowego z otoczenia, następuje generacja litery alfabetu wyjściowego. Przykładem algorytmów, dla których opis "bodziec - reakcja" jest zbyt ubogi, są algorytmy zapisywane w postaci sieci działań. Urządzenia działające na bazie tych sieci reagują na bodziec z otoczenia, ale następuje to po wykonaniu wielu działań. Należy zatem przetransformować sieć działań na graf automatu. Przykładowa transformacja sieci działań algorytmu szeregowego na graf automatu zostanie przedstawiona poniżej. Analizowana sieć działań przedstawiona jest na rys. 1.4.

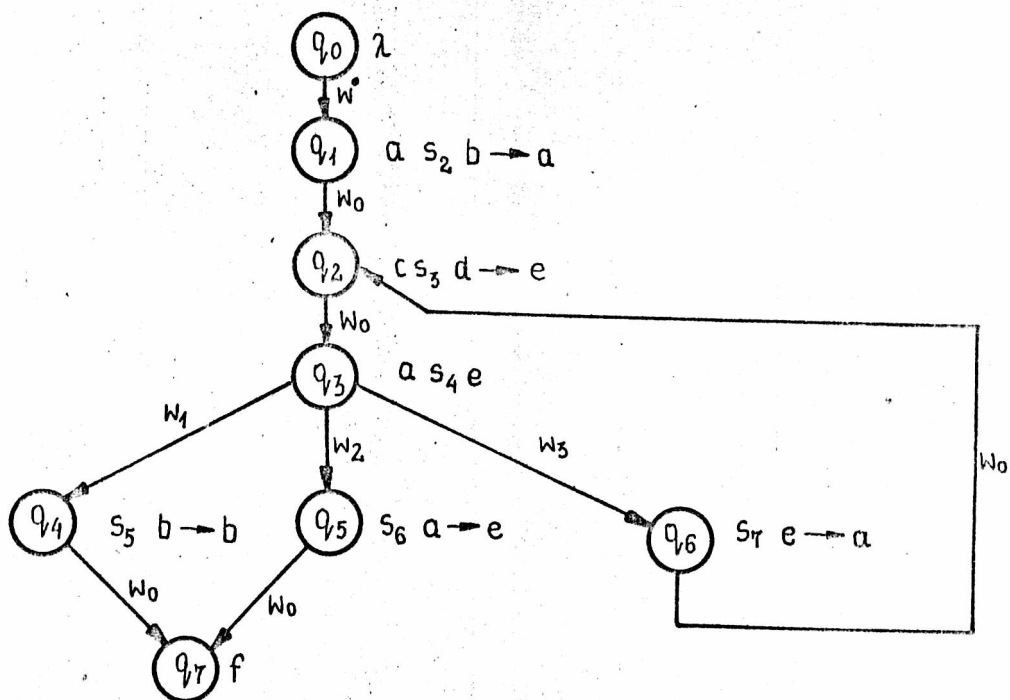
W rozważanej sieci działania s_1 , s_8 są działaniami zeroargumentowymi, działania s_5 , s_6 , s_7 - jednoargumentowymi a działania s_2 , s_3 , s_4 - dwuargumentowymi. Działaniu s_1 odpowiada początek sieci działań, a działaniu s_8 - jej koniec. Symbole w_1 , w_2 , w_3 opisują różne wyniki, działania decyzyjnego s_4 .

Wynikiem transformacji rozważanej sieci działań na graf automatu - działającego według algorytmu zapisanego w postaci tej sieci - jest graf przedstawiony na rys. 1.5.

Gotowość bloku arytmetyczno-logicznego do wykonania następnego działania sygnalizowana jest symbolami w_0 , w_1 , w_2 , w_3 . Symbol w^* inicjuje proces realizacji algorytmu. Symbol λ jest symbolem litery pustej. Automat, wyrażony grafem przedstawionym na rys. 1.5., będąc w stanie początkowym q_0 generuje literę pustą. W stanie końcowym - q_7 , automat ten generuje literę f , co sygnalizuje, że działanie wg rozważanego algorytmu zostało zakończone. Pojawienie się na wejściu automatu będącego w stanie q_1 litery w_0 powoduje przejście automatu do stanu q_2 i wygenerowanie litery alfabetu wyjściowego, którą jest litera c s_3 $d \rightarrow e$. Powoduje to wykonanie działania s_3 przez blok arytmetyczno-logiczny oraz niezbędnych przesłań między tym blokiem a pamięcią. W podobny sposób interpretowane są litery alfabetu wyjś-



Rys. 1.4. Przykładowa sieć działań



Rys. 1.5. Graf automatu odpowiadający sieci działań z rys. 1.4.

ciowego analizowanego automatu opisujące wierzchołki $q_1, q_3 - q_6$ grafu tego automatu.

Sterowanie wg równoległej sieci działań za pomocą jednego automatu jest bardziej złożone.

Zadanie K_1 jest realizowane zgodnie z wariantem rozwiązania tego zadania. Symbolem WRZ oznaczony jest wariant rozwiązania zadania. WRZ można opisać za pomocą grafu izomorficznego z grafem pewnego automatu zdolnego do realizacji tego zadania zgodnie z jego wariantem rozwiązania. WRZ można rozbić na podzadania. W pracy rozważane są dwa przypadki podziału grafu WRZ na grafy podzadań. W pierwszym ze wspomnianych przypadków graf podzadania jest podgrafem grafu zadania, natomiast w drugim - podgrafem cząstkowym grafu WRZ. Niżej analizowana będzie transformacja przykładowej sieci działań, podzielonej na podsieci, na graf WRZ.

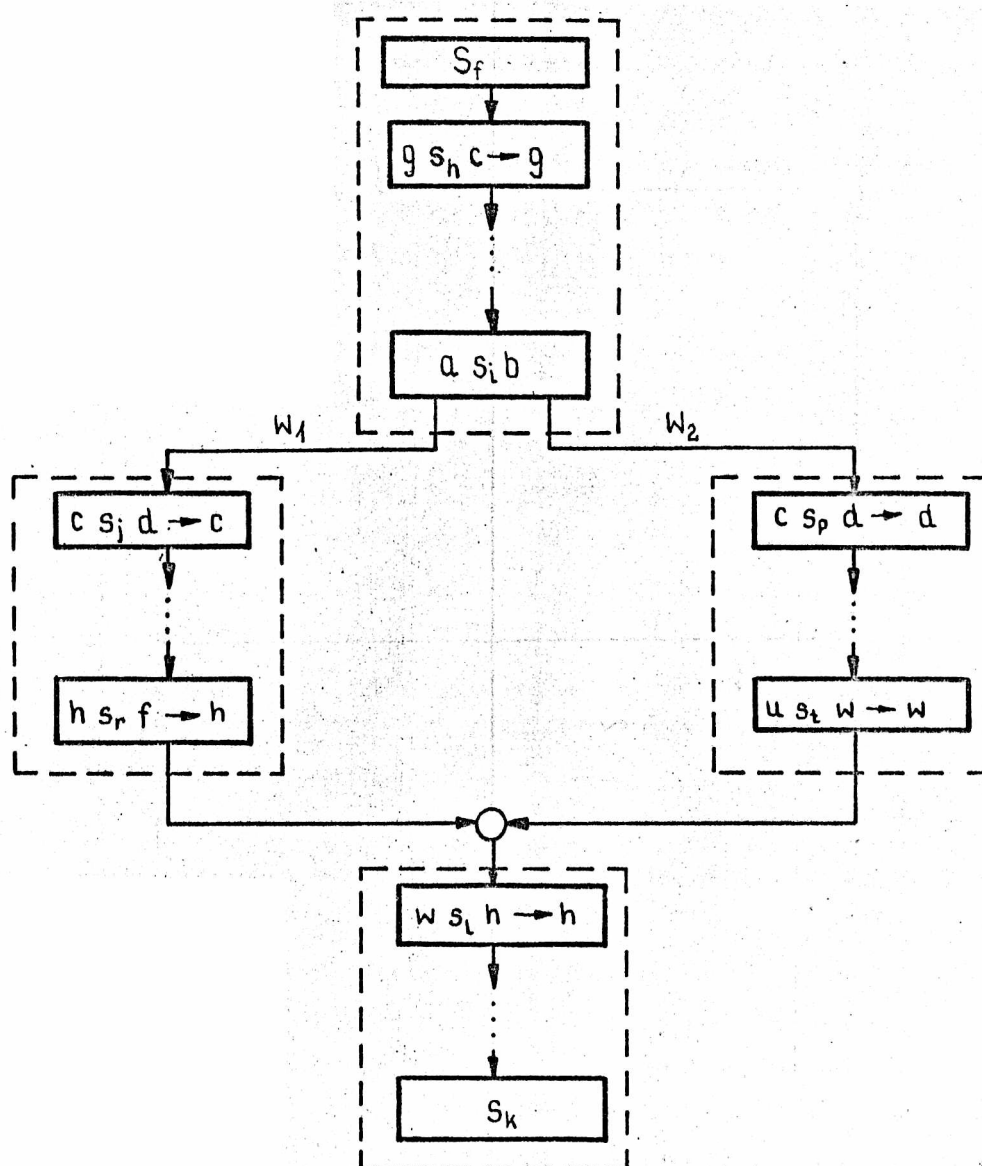
Rozważaną sieć działań ilustruje rys. 1.6., natomiast graf wariantu rozwiązania zadania odpowiadający tej sieci przedstawiony jest na rys. 1.7.. Graf podzadania, w omawianym przykładzie jest podgrafem grafu wariantu rozwiązania zadania.

Graf wariantu rozwiązania zadania będący wynikiem transformacji sieci działań z rys. 1.6. przedstawiony jest na rys. 1.7..

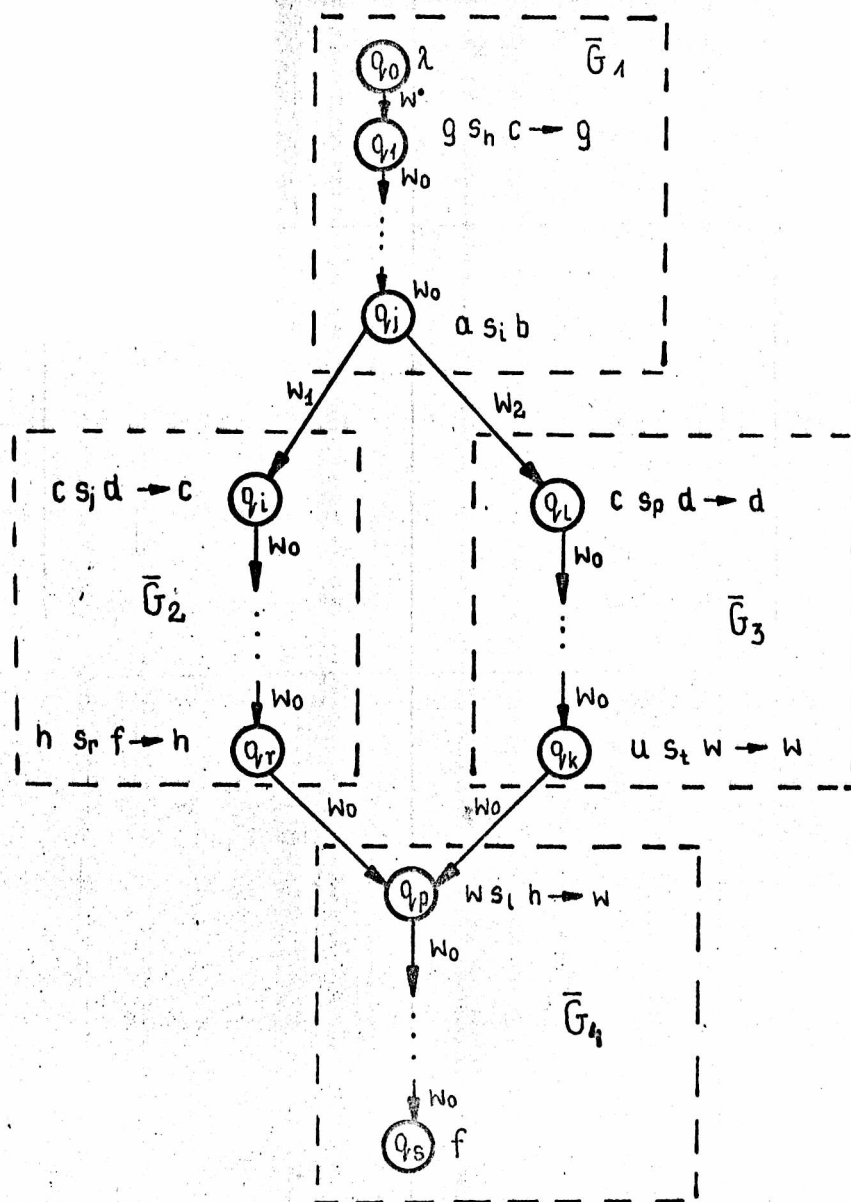
Wariant rozwiązania zadania o grafie przedstawionym na rys. 1.7. składa się z czterech podzadań. Łuki łączące grafy podzadań nie należą do grafów opisujących podzadania.

Niżej rozpatrzony zostanie sposób uzyskania grafu $\bar{G}^k$ WRZ na bazie grafów $\bar{G}_1, \bar{G}_2, \bar{G}_3, \bar{G}_4$ podzadań i odpowiedniego zbioru S . Niech zbiór S ma następującą postać:

$$S = \left\{ \langle q_j, w_1, q_1 \rangle, \langle q_j, w_2, q_1 \rangle, \langle q_r, w_0, q_p \rangle, \right. \\ \left. \langle q_k, w_0, q_p \rangle \right\} \quad (1.5)$$



Rys. 1.6. Przykładowa sieć działań z wyróżnionymi podsieciami objętymi liniami przerywanymi



Rys. 1.7. Graf $\bar{G}^k$ wariantu rozwiązania zadania odpowiadający sieci działań z rys. 1.6. z wyróżnionymi podgrafami $\bar{G}_1, \bar{G}_2, \bar{G}_3, \bar{G}_4$ opisującymi podzadania

Elementy zbioru S wskazują na sposób łączenia grafów podzadań w graf $\bar{G}^k$ wariantu rozwiązania zadania.

Ogólnie graf podzadania jest grafem znacznie bardziej złożonym niż droga elementarna. Wywołane to jest złożonością grafów automatów.

Poprzednie rozważania dotyczyły modelu podzadania w postaci grafu będącego podgrafem grafu WRZ. Modelem ogólniejszym jest model podzadania w postaci grafu będącego podgrafem częściowym grafu wariantu rozwiązania zadania.

W zakresie grafów wariantów rozwiązania zadań przyporządkowanych sieciom działań nie jest uzasadnione analizowanie podzadań opisanych grafami będącymi podgrafami częściowymi grafu WRZ. Jednak w zakresie zadań, które mogą być realizowane za pomocą automatów, należy uwzględnić podzadania wyrażone grafami będącymi podgrafami częściowymi grafu wariantu rozwiązania zadania. Uzasadnić to można tym, że pewne fragmenty grafu WRZ mogą być wspólne dla co najmniej dwu grafów podzadań.

1.4. ELEMENT UCZĄCY SIĘ W SYSTEMIE OPERACYJNYM PRACUJĄCYM NA ZASADZIE DZIAŁANIA AUTOMATU PARAMETRYCZNEGO

Zbiór zadań, do których realizacji zdolny jest system operacyjny, przy pominięciu procesu uczenia się, zadany jest wyrażeniem (1.1). Zadania te realizowane są zgodnie z przyjętymi wariantami ich rozwiązania. Zadaniu K_1 przyporządkowana jest wzajemnie jednoznacznie wartość p_1 parametru p . Stąd zbiorowi K odpowiada zbiór P wartości parametru p :

$$P = \{ p_1, \dots, p_1, \dots, p_n \} \quad (1.6)$$

Zadaniu K_1 przyporządkowany jest graf $\bar{G}^1$ automatu zdolnego do realizacji tego zadania. Graf ten odwzorowuje przyjęty wariant rozwiązania zadania K_1 . Na przykład zadaniu K_j diagnozowania pewnego podsystemu SC początkowo przyporządkowany jest graf $\bar{G}^j$ automatu odwzoro-

wujący stosowany w początkowej fazie pracy SC sposób diagnozowania tego podsystemu. W wyniku rozpoznawania wzmiankowanego podsystemu, ze względu na ilość uszkodzeń jego elementów, formułowany jest nowy wariant rozwiązania zadania K_j , umożliwiający uzyskanie mniejszego średniego czasu lokalizacji przyczyny błędu. Jeśli uczący się automat parametryczny przeprowadzi syntezę automatu zdolnego do realizacji rozważanego zadania wg nowego wariantu jego rozwiązania, to graf G^j zostanie zastąpiony grafem G^B zbudowanego automatu. W omawianym wyżej przypadku mówimy, że nastąpiła zmiana wariantu rozwiązania zadania.

W celu opanowania realizacji nowego zadania również podejmowane są przez uczący się automat parametryczny (UAP) działania, których celem jest synteza automatu zdolnego do realizacji tego zadania zgodnie z przyjętym wariantem jego rozwiązania.

Nowo budowane automaty w obu rozważanych przypadkach są tzw. automatami zagregowanymi (przedstawionymi w podrozdziale 2.3. pracy) odwzorowywanymi działaniem automatu UAP. Automat ten zawiera automat parametryczny w sensie automatu przedstawionego w pracach [25], [27] oznaczonego symbolem AP.

Zbiór zadań i zbiór wartości parametru p nie ulega zmianie w przypadku zmiany wariantu rozwiązania zadania, natomiast zbiór grafów automatów poprzednio odwzorowywanych działaniem uczącego się automatu parametrycznego zostaje zastąpiony zbiorem o tej samej mocy, ale zawierającym jeden inny element. Mogą również wystąpić sytuacje, w których następuje zastąpienie zbioru grafów automatów odwzorowywanych działaniem UAP, zbioru zadań i zbioru wartości parametru p - zbiorami o większej mocy. Na przykład w przypadku opanowania przez uczący się automat parametryczny translacji nowego języka, do każdego z trzech uprzednio rozważanych zbiorów zostaje dołączony jeden element, w wyniku czego powstaną trzy nowe zbiory.

Definicja 1.1.

Etapem procesu uczenia się nazywany jest proces prowadzący do opanowania przez automat UAP nowego zadania lub zmodyfikowanego wariantu rozwiązania zadania, do którego realizacji automat ten był uprzednio zdolny.

Niech K^i będzie symbolem zbioru zadań, do których realizacji zdolny jest automat UAP po i -tym etapie procesu uczenia się. Zbiór ten ma postać:

$$K^i = \{K_1, \dots, K_l, \dots, K_n, K_{n+1}, \dots, K_{n+j}\} \quad j \leq i$$

Nierówność $j < i$ jest prawdziwa, jeśli na co najmniej jednym z etapów procesu uczenia się nastąpiła wymiana grafu automatu dla realizacji tego samego zadania. Jeśli na każdym z etapów procesu uczenia się UAP opanowywał nowe zadanie, to $j = i$.

Elementy $K_1, \dots, K_l, \dots, K_n$ zbioru K^i są identyczne z elementami zbioru K , mimo że pewne z tych elementów mogą być realizowane zgodnie ze zmodyfikowanymi wariantami ich rozwiązania. Element $K_c \in K$, dla którego nie nastąpiła zmiana wariantu rozwiązania zadania, po i -tym etapie procesu uczenia się ale przed $(i+1)$ -szym realizowany jest za pomocą automatu odwzorowywanego działaniem AP. Elementy $K_{n+1}, \dots, K_{n+j}$ są nowymi zadaniami, których realizację opanował automat UAP.

Elementy zbioru K , dla których nastąpiła zmiana wariantu rozwiązania zadania i nowe zadania opanowane przez UAP realizowane są za pomocą automatów zbudowanych w procesie uczenia się - zwanych automatami zagregowanymi. Automaty te odwzorowywane są działaniem UAP, który w tym celu stosuje AP. Stąd automat AP jest podstawą do realizacji zadań ze zbioru K , dla których nie nastąpiła zmiana wariantu ich rozwiązania oraz zadań realizowanych za pomocą automatów zagregowanych.

Zbiór wartości parametru p po i -tym etapie procesu uczenia się ma postać:

$$P^i = \{p_1, \dots, p_1, \dots, p_n, p_{n+1}, \dots, p_{n+j}\} \quad j \leq i$$

Zbiór P^i można rozbić na dwa rozłączne podzbiory oznaczone symbolami P_1^i , P_2^i . Podzbiór P_1^i jest zbiorem wartości parametru p przyporządkowanych zadaniom ze zbioru K , dla których w trakcie i etapów procesu uczenia się nie nastąpiła zmiana wariantu ich rozwiązania. Podzbiór P_2^i jest zbiorem wartości parametru p przyporządkowanych zadaniom, które po i -tym etapie procesu uczenia się realizowane są za pomocą automatów zagregowanych.

Jeśli na wejście automatu UAP po i -tym etapie procesu uczenia się ale przed $(i+1)$ -szym podana zostanie wartość $p_k \in P_1^i$ to automat ten poda na wejście automatu AP wartość p_k , co spowoduje realizację zadania K_k za pomocą AP. W przypadku podania na wejście UAP wartości $p_r \in P_2^i$, automat ten będzie sterował automatem AP w celu zrealizowania zadania K_r . Wspomniane sterowanie polega na wyodrębnieniu fragmentów automatów odwzorowywanych działaniem AP. Fragmenty te realizują fragmenty zadania K_r realizowanego przez uczący się automat parametryczny.

Uczący się automat parametryczny, na podstawie grafu wariantu rozwiązania nowego zadania lub nowego wariantu rozwiązania zadania uprzednio realizowalnego przez rozważany automat, podejmuje działania, których celem jest synteza odpowiedniego automatu zagregowanego. Jeśli zostanie zbudowany automat zagregowany zdolny do realizacji nowego zadania lub nowego wariantu rozwiązania zadania, to zostanie przeprowadzony kolejny etap procesu uczenia się.

Odrębnym problemem jest sposób uzyskiwania grafu wariantu rozwiązania nowego zadania lub grafu zmodyfikowanego wariantu rozwiązania zadania uprzednio realizowalnego przez automat UAP.

W prostszym modelu uczenia się wzmiankowany graf formułowany

jest przez otoczenie automatu UAP, które pełni rolę nauczyciela. Wymienioną rolę może pełnić człowiek. Model ten jest pewnym analogiem uczenia się przez naśladowanie. W przypadku bardziej złożonego modelu uczenia się, graf nowego wariantu rozwiązania zadania może być budowany przez UAP. W celu zilustrowania tego przypadku zostanie przedstawiony przykład.

Rozważmy jako zadanie - szukanie rekordu w rozważanym poprzednio zbiorze. Niech każdy z jego podzbiorów będzie zorganizowany w inny sposób niż pozostałe podzbiory. Warunek ten wymaga zapewnienia dla wyszukiwania rekordu w każdym podzbiorze odrębnego podzadania. Każde z tych podzadań opisane jest za pomocą odrębnego grafu. Na początkowym etapie procesu wyszukiwania rekordu we wspomnianym zbiorze wykorzystywane są technika randomizacji i procedura unikania kolizji. W wyniku poznawania charakterystyki zbioru następuje uporządkowanie jego podzbiorów, ze względu na liczbę trafnych dostępów do każdego z podzbiorów. Uporządkowanie to jest podstawą do wyznaczenia kolejności przeszukiwania podzbiorów. Stąd wspomniane uporządkowanie jest podstawą do procesu budowy, przeprowadzanego przez automat UAP, grafu opanowywanego wariantu rozwiązania zadania z grafów podzadań zapewniających szukanie rekordów w poszczególnych podziorach.

Sformułowanie algorytmu umożliwiającego budowę grafu wariantu rozwiązania zadania dla tych zadań, dla których odpowiedni graf może być zbudowany przez człowieka jest obecnie niemożliwe, chociaż w przyszłości automat uczący się powinien być zdolny do samodzielnego budowania grafu wariantu rozwiązania zadania dla możliwie dużej klasy zadań. Wzmiankowany algorytm musiałby bazować na metodach heurystycznych. Metody te są w większości przypadków metodami wyspecjalizowanymi, przydatnymi do rozwiązywania wąskich problemów. Nie znane są ogólne zasady heurystyki, a tzw. Ogólny Rozwiązywacz Problemów przydatny jest w rozwiązywaniu tylko niektórych problemów i wymaga złożo-

nego opisu tych problemów. Ponieważ nie jest możliwe sformułowanie ogólnego algorytmu budowy grafu wariantu rozwiązania zadania, a więc dla każdej z rozważanych klas zadań należy sformułować odrębny algorytm konstrukcji tego grafu. Algorytm ten musi wykorzystywać aparat formalny opisu zadania. Na przykład jeśli zadaniem jest tłumaczenie programów napisanych w pewnym nowym języku programowania, to wspomniany aparat powinien opisywać ten język, język wewnętrzny jednostki centralnej.

Oba przedstawione wyżej modele uczenia się wymagają rozwiązania problemu budowy automatu zagregowanego na podstawie grafu wariantu rozwiązania zadania. W pracy przedstawiono propozycję rozwiązania tego problemu.

2. MODEL FORMALNY UCZĄCEGO SIĘ AUTOMATU PARAMETRYCZNEGO

2.1. OGÓLNA CHARAKTERYSTYKA KONCEPCJI UCZENIA SIĘ AUTOMATU PARAMETRYCZNEGO

Automat parametryczny w sensie automatu przedstawionego w pracach [25] , [27] zdolny jest do realizacji zadań ze zbioru K , określonego za pomocą wyrażenia (1.1). Automat ten w dalszych rozważaniach oznaczany będzie symbolem AP. Automat AP odwzorowuje swym działaniem zbiór automatów:

$$A^+ = \{ +A^1, \dots, +A^j, \dots, +A^n \} \quad (2.1)$$

Każdy z automatów $+A^j \in A^+$ zdolny jest do realizacji pewnego zadania ze zbioru K zgodnie z przyjętym, przed procesem uczenia się, wariantem jego rozwiązania.

Uczący się automat parametryczny (UAP), na $(i+1)$ -szym etapie procesu uczenia się, zdolny jest do opanowania nowego zadania lub nowego wariantu rozwiązania takiego zadania, do którego realizacji automat ten zdolny jest po i etapach procesu uczenia się.

Uwaga:

Opanowanie nowego zadania związane jest z opanowaniem pewnego wariantu rozwiązania tego zadania. Stąd można ogólnie stwierdzić, że automat UAP opanowuje nowe warianty rozwiązania zadań. W celu skrócenia zapisów, w pracy będziemy posługiwali się określeniem "nowy wariant rozwiązywania zadania" jako terminem obejmującym wariant rozwiązania nowego zadania i nowy wariant rozwiązania takiego zadania, do którego realizacji automat UAP był uprzednio zdolny.

Wariant rozwiązania zadania można interpretować jako system podzadań. Stąd, przyjętemu przed procesem uczenia się, wariantowi rozwiązania zadania $K_j \in K$ można przyporządkować zbiór wchodzących w jego

skład podzadań. Zbiór ten określony jest następująco:

$$M_j^+ = \{ M_{j1}, \dots, M_{j1}, \dots, M_{js} \} \quad (2.2)$$

gdzie M_{ji} jest podzadaniem przyjętego przed procesem uczenia się wariantu rozwiązania zadania K_j .

Zadanie $K_j \in K$ realizowane jest, wg przyjętego przed procesem uczenia się wariantu jego rozwiązania, przez automat $^+A^j$ odwzorowywany działaniem automatu AP. Stąd elementy zbioru M_j^+ realizowane są za pomocą automatu AP. Podzadanie M_{ji} realizowane jest za pomocą podautomatu wchodzącego w skład automatu $^+A^j$. Zbiór podzadań realizowanych przez automat AP określony jest następująco:

$$M^+ = \bigcup_{j=1}^n M_j^+ \quad (2.3)$$

Pewne podzadania wchodzi w skład więcej niż jednego, przyjętego przed procesem uczenia się, wariantu rozwiązania zadania $K_j \in K$. Stąd podzadania o wzmiankowanej własności realizowane są za pomocą różnych automatów ze zbioru A^+ , w zależności od tego, w skład jakiego przyjętego przed procesem uczenia się wariantu rozwiązania zadania $K_j \in K$ wchodzi to podzadanie. Należy dokonać wyboru tylko jednego podautomatu dla realizacji danego podzadania. W wyniku wymienionego wyboru otrzymujemy zbiór podautomatów:

$$L = \{ A_i \} \quad i \in I \quad (2.4)$$

gdzie I - zbiór indeksów symboli podautomatów uzyskanych dzięki przeprowadzonemu wyborowi.

Zbiór L podautomatów odwzorowywanych działaniem automatu AP jest podstawą do budowy automatów nazwanych automatami zagregowanymi.

Budując automaty zagregowane na bazie zbioru L , można otrzymywać automaty realizujące zadania zgodnie z takimi nowymi wariantami ich rozwiązania, które składają się z podzadań wchodzących w skład przyjętych przed procesem uczenia się wariantów rozwiązania zadań ze zbioru K . Na przykład jeśli nowy język programowania zawiera elementy języków istniejących, to translator tego języka będzie zawierał fragmenty translatorów języków istniejących.

Wariant rozwiązania zadania można przedstawić za pomocą grafu. Graf ten jest izomorficzny z grafem pewnego automatu, który jest zdolny do realizacji tego zadania zgodnie z wymienionym wariantem jego rozwiązania. W grafie wariantu rozwiązania zadania można wyróżnić fragmenty opisujące podzadania. Fragmenty te, w zależności od rozpatrywanego w pracy przypadku, są podgrafami lub podgrafami cząstkowymi grafu wariantu rozwiązania zadania. Podgraf grafu jest jednocześnie podgrafem cząstkowym tego grafu. Stąd, w celu uproszczenia opisu, we fragmentach treści pracy wspólnych dla obu przypadków używany będzie termin podgraf cząstkowy.

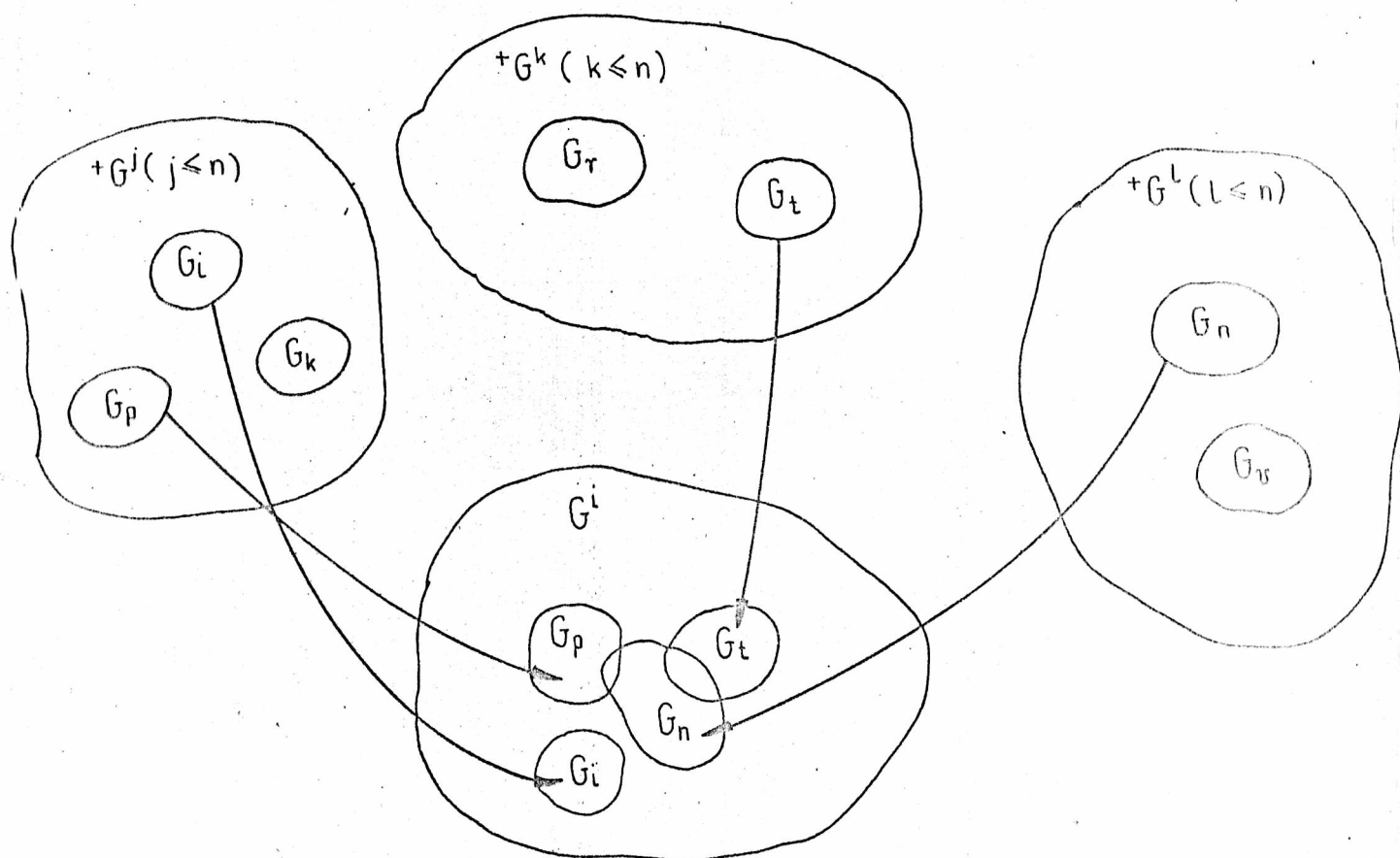
Podzadaniu $M_{ji} \in M^+$ przyporządkowany jest podautomat $A_c \in L$ o grafie izomorficznym z grafem tego podzadania. Grafy podautomatów ze zbioru L są podgrafami cząstkowymi grafów automatów ze zbioru A^+ automatów odwzorowywanych działaniem AP . Każdy z automatów odwzorowywanych działaniem AP można rozpatrywać jako system współpracujących ze sobą podautomatów, będących elementami zbioru L .

Automat zagregowany jest systemem podautomatów przynależnych do różnych automatów odwzorowywanych działaniem AP . Grafy podautomatów są podgrafami cząstkowymi grafu automatu zagregowanego. Automat ten budowany jest w celu realizacji nowego zadania lub takiego zadania, które poprzednio mogło być realizowane przez automat UAP ale zgodnie z nowym wariantem jego rozwiązania.

Automat AP odwzorowuje wiele automatów $A^j \in A^+$ będących sys-

temami podautomatów. Zbiór podautomatów odwzorowywanych działaniem AP jest podstawą do określenia możliwości automatu UAP w zakresie odwzorowywania automatów zagregowanych. Automaty zagregowane, budowane w procesie uczenia się, stanowią małą część zbioru automatów zagregowanych, które mogą być odwzorowywane działaniem automatu UAP. Zbiór L podautomatów jest podstawą do określenia możliwości automatu UAP w zakresie opanowywania nowych wariantów rozwiązania zadań.

Sposób konstrukcji automatu zagregowanego z automatów odwzorowywanych działaniem automatu AP ilustruje rys. 2.1.



Rys. 2.1. Ilustracja sposobu budowy grafu G^i automatu zagregowanego z grafów automatów odwzorowywanych działaniem automatu AP

Grafy $+G^j$, $+G^k$, $+G^l$, przedstawione szkicowo na rys.2.1, są grafami

automatów odwzorowywanych działaniem AP. Grafy $G_i, G_p, G_k, G_r, G_t, G_n, G_v$ są grafami podautomatów ze zbioru L. Graf G^i automatu zagregowanego jest wynikiem wykonania pewnych operacji na grafach $G_i, G_p, G_n, G_t, \dots$.

Zasada działania automatu zagregowanego różni się bardzo istotnie od zasady działania automatów budowanych z takich automatów składowych, które istnieją jako odrębne obiekty fizyczne. Stan automatu zbudowanego z odrębnych obiektów fizycznych wyznaczony jest przez stany automatów składowych. W przypadku automatu zagregowanego, zbudowanego na bazie zbioru podautomatów odwzorowywanych działaniem automatu AP, w danej chwili istnieje tylko jeden podautomat, a mianowicie ten, który w tej chwili jest odwzorowywany działaniem AP. Stan automatu zagregowanego wyznaczony jest przez stan podautomatu odwzorowywanego aktualnie działaniem automatu AP.

Graf opanowywanego wariantu rozwiązania zadania jest podstawą procesu syntezy automatu zagregowanego zdolnego do realizacji tego zadania zgodnie z nowym wariantem jego rozwiązania. Przyjmuje się, że w grafie wariantu rozwiązania zadania graf danego podzadania występuje tylko jeden raz. Uwzględnienie możliwości wielokrotnego występowania grafu pewnego podzadania w grafie wariantu rozwiązania zadania nie powoduje trudności, ale implikuje zbędne zwiększenie stopnia szczegółowości rozważań. Automat UAP może opanować te warianty rozwiązania zadań, których graf jest izomorficzny z grafem pewnego automatu zagregowanego możliwego do uzyskania za pomocą zbioru L podautomatów.

W procesie uczenia się, w grafie nowego wariantu rozwiązania zadania, wyszukiwane są podgrafy cząstkowe opisujące podzadania. Następnie uzyskanym, w przedstawiony wyżej sposób, podzadaniom zostają przyporządkowane zdolne do ich realizacji podautomaty. Zbiór otrzymanych w ten sposób i odpowiednio ze sobą współpracujących pod-

automatów tworzy taki automat zagregowany, który zdolny jest do realizacji rozważanego zadania zgodnie z nowym wariantem jego rozwiązania.

Uzyskane w procesie uczenia się automaty zagregowane odwzorowywane są działaniem automatu UAP, który w celu ich odwzorowywania wykorzystuje podautomaty odwzorowywane działaniem automatu AP.

2.2. PORÓWNANIE PRZYJĘTEJ KONCEPCJI Z INNYMI KONCEPCJAMI UCZENIA SIĘ

Kryterium wymienionego porównania jest przydatność koncepcji do zastosowania w systemie operacyjnym SC.

W badaniach teoretycznych procesów uczenia się, w zakresie teorii automatów, wykorzystywane są automaty probabilistyczne [9], automaty rozmyte [52], automaty skłonnościowe [51]. W celu uzyskania pożądanego zachowania się tych automatów, w rozważanym w pracy przypadku - w celu opanowania wariantu rozwiązania zadania, niezbędne jest przeprowadzenie długotrwałego procesu uczenia się. Na taki proces nie zawsze możemy sobie pozwolić w zakresie formułowania algorytmów wchodzących w skład systemu operacyjnego. Badania procesów uczenia się wymienionych wyżej automatów obejmują złożone badania zbieżności zachowania uzyskanego po określonej liczbie kroków procesu uczenia się do zachowania optymalnego. Analiza wzmiankowanej zbieżności powinna być przeprowadzona w celu określenia liczby kroków procesu uczenia się.

W przypadku budowania grafu opanowywanego wariantu rozwiązania zadania przez otoczenie automatu UAP, w zakresie przedstawionej w pracy koncepcji, proces budowy automatu zagregowanego dla zadanego grafu przeprowadzany jest tylko jeden raz.

W przypadku syntezy wzmiankowanego grafu za pomocą odpowiedniego algorytmu, należy rozważany algorytm wyposażyć w elementy

umożliwiający minimalizację liczby kroków procesu uczenia się. Nie można podać ogólnych zasad zapewniających tę minimalizację, ponieważ - z przyczyn podanych w podrozdziale 1.4. pracy - nie można sformułować ogólnego algorytmu budowy grafu opanowywanego wariantu rozwiązania zadania. Można jedynie próbować określać, dla algorytmu budowy grafu wariantu rozwiązania zadania dla danej klasy zadań, sposoby minimalizacji liczby kroków procesu uczenia się.

System operacyjny jest systemem wykonującym znaczną liczbę zadań. Próba realizacji każdego z tych zadań za pomocą automatu będącego odrębnym obiektem fizycznym posiada te wady, że nie uwzględnia się podobieństw między grafami rozważanych automatów. W związku z czym uwzględnienie części tych podobieństw umożliwia uzyskanie automatu odwzorowującego wspomniane automaty, będącego urządzeniem prostszym niż urządzenie będące zbiorem rozważanych automatów. Pewne cechy wspólne dla podzbiorów zbioru automatów odwzorowywanych działaniem automatu AP odwzorowywane są za pomocą struktury tego automatu tylko jeden raz.

Niżej zostanie przedstawione uzasadnienie tego, że automat AP może mieć prostszą strukturę niż zbiór automatów zrealizowanych jako odrębne obiekty fizyczne.

Nazwijmy grafem nieobciążonym automatu graf tego automatu pozbawiony opisu łuków - literami alfabetu wejściowego i opisu wierzchołków - literami alfabetu wyjściowego rozważanego automatu. W grafie automatu bez wyjścia automatu AP uwzględnione są podobieństwa grafów nieobciążonych automatów odwzorowywanych działaniem automatu AP.

W strukturze bloku wejściowego automatu AP uwzględnione są pewne podobieństwa w zakresie opisu literami alfabetów wejściowych łuków grafów automatów.

W strukturze bloku wyjściowego automatu AP uwzględnione są niektóre podobieństwa w zakresie opisu literami alfabetów wyjścio-

wych wierzchołków grafów automatów.

W wyniku procesu syntezy automatu zagregowanego dla nowego wariantu rozwiązania zadania, zostaje odwzorowany w uczącym się automacie parametrycznym sposób współpracy podautomatów wchodzących w skład uzyskanego automatu zagregowanego.

Natomiast w celu zapewnienia możliwości uczenia się realizacji wszystkich wariantów rozwiązania zadań opisywalnych grafami skończonymi, a nie tylko tych wariantów rozwiązania zadań, których realizację może opanować automat UAP przy danym zbiorze podautomatów, należy zapewnić możliwość odrębnego odwzorowania każdego łuku i wierzchołka grafu automatu zdolnego do realizacji pewnego zadania zgodnie z określonym wariantem jego rozwiązania. Odwzorowanie to wymaga pamiętania całego grafu wymienionego automatu. Stąd rozważania dotyczące tego przypadku zostaną w pracy pominięte.

2.3. AUTOMATY ZAGREGOWANE - PRODUKT PROCESU UCZENIA SIĘ AUTOMATU PARAMETRYCZNEGO

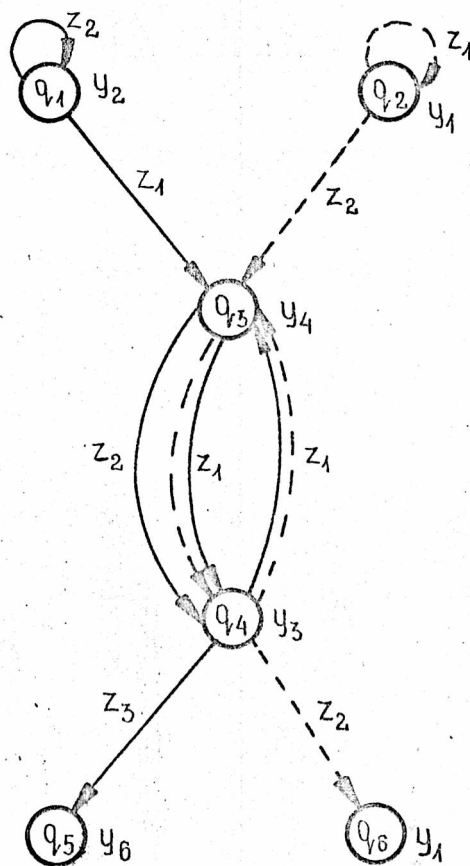
2.3.1. Pojęcie automatów zagregowanych i ich klasyfikacja

Dla opanowywanego w procesie uczenia się nowego zadania lub nowego wariantu rozwiązania zadania, które mogło być uprzednio realizowane za pomocą automatu UAP konstruowany jest automat zagregowany.

W pracy przedstawione zostaną definicje automatów zagregowanych pierwszego i drugiego rodzaju. Automaty zagregowane obu rodzajów są systemami podautomatów przynależnych do różnych automatów odwzorowywanych działaniem automatu AP. Do podziału automatów zagregowanych na dwa rodzaje wykorzystuje się pojęcie grafu automatu.

Automat zagregowany pierwszego rodzaju charakteryzuje się tym, że jego graf składa się z grafów podautomatów będących podgrafami tego grafu.

Nie zawsze podautomat realizujący podzadanie wyrażany jest grafem będącym podgrafem grafu automatu realizującego zadanie, w którego skład wchodzi wymienione podzadanie. Niżej zostanie przedstawiony przykład automatu, w którym pewien podautomat wyrażony jest grafem nie będącym podgrafem tego automatu. Niech dany będzie graf G' automatu przedstawiony na rys. 2.2.



Rys. 2.2. Graf G' automatu

Na rys. 2.2. łuki grafu G_1 podautomatu narysowane są linią ciągłą, natomiast łuki grafu G_2 - linią przerywaną. Graf G_1 jest podgrafem grafu G' , ale podgrafem grafu G' nie jest graf G_2 będący podgrafem cząstkowym grafu G' .

Automat zagregowany drugiego rodzaju charakteryzuje się tym, że jego graf zawiera takie podgrafy cząstkowe, które są izomorficzne z grafami podautomatów.

2.3.2. Synteza formalna automatów zagregowanych pierwszego rodzaju

Podstawą do uzyskiwania automatów zagregowanych pierwszego rodzaju są podautomaty odwzorowywane działaniem automatu AP - wyrażone grafami ze zbioru:

$$G^+ = \{G_i\} \quad i \in I \quad (2.5)$$

W wyniku wykonania operacji oznaczonej symbolem $\oplus^j$ na podzbiorze zbioru G^+ otrzymujemy graf automatu zagregowanego pierwszego rodzaju (AZI). Grafy podautomatów są podgrafami grafu automatu AZI.

W początkowej części tego punktu przedstawiona zostanie operacja $\oplus^j$. Graf G_i jest grafem podautomatu A_i będącego deterministycznym automatem Moore'a. Podautomat ten określony jest następująco:

$$A_i = \langle Z_i, Y_i, Q_i, \Phi_i, \Psi_i \rangle \quad (2.6)$$

gdzie Z_i - alfabet wejściowy A_i , Y_i - alfabet wyjściowy A_i , Q_i - zbiór stanów A_i , Φ_i - funkcja przejść A_i , Ψ_i - funkcja wyjść A_i .

Funkcja przejść Φ_i podautomatu A_i określona następująco:

$$q_j = \Phi_i(q_s, z_p) \quad (2.7)$$

jest relacją [31], [44] :

$$\Phi_i \subset (Q_i \times Z_i) \times Q_i \quad (2.8)$$

Stąd elementami Φ_i są pary uporządkowane $\langle \langle q_s, z_p \rangle, q_j \rangle$. Dla tych par prawdziwa jest następująca zależność:

$$\langle \langle q_s, z_p \rangle, q_j \rangle = \langle q_s, z_p, q_j \rangle \quad (2.9)$$

Z wyrażenia (2.9) wynika, że $\langle q_s, z_p, q_j \rangle \in \Phi_i$ czyli $\Phi_i \subset Q_i \times Z_i \times Q_i$.

Funkcja wyjść Ψ_i podautomatu A_i określona następująco:

$$y_r = \Psi_i(q_t) \quad (2.10)$$

jest relacją:

$$\Psi_i \subset Q_i \times Y_i \quad (2.11)$$

Zatem elementami Ψ_i są pary uporządkowane $\langle q_t, y_r \rangle$.

Niżej zostanie przedstawiona definicja deterministycznego automatu Moore'a.

W danym stanie q_i deterministycznego automatu Moore'a A może być generowana tylko jedna litera alfabetu wyjściowego automatu A . Po podaniu litery alfabetu wejściowego na wejście automatu A znajdującego się w stanie q_i , automat ten osiągnie stan q_j jednoznacznie określony przez relację charakteryzującą przejścia między stanami automatu A .

Definicja 2.1.

Deterministycznym automatem Moore'a nazywamy automat A , dla którego relacji Φ charakteryzującej przejścia między stanami prawdziwa jest zależność:

$$(\forall \langle q_i, z_p, q_j \rangle, \langle q_a, z_s, q_b \rangle \in \Phi) ((q_i = q_a \wedge z_p = z_s) \Rightarrow q_j = q_b) \quad (2.12)$$

natomiast dla relacji Ψ opisującej reakcje tego automatu prawdziwe jest wyrażenie:

$$(\forall \langle q_i, y_p \rangle, \langle q_t, y_s \rangle \in \Psi) (q_i = q_t \Rightarrow y_p = y_s) \quad (2.13)$$

Z powyższego wynika, że relacje Φ, Ψ są funkcjami.

Ponieważ relacje Φ_i, Ψ_i dla podautomatu A_i są funkcjami, a więc A_i jest automatem deterministycznym.

Z kolei zostanie określony graf G_i podautomatu A_i .

Wierzchołki grafu G_i przyporządkowane są stanom podautomatu A_i . Funkcja wyjść Ψ_i podautomatu A_i określa sposób opisanie wierzchołków grafu G_i literami alfabetu wyjściowego Y_i . Elementowi $\langle q_i, z_p, q_j \rangle \in \Phi_i$

przyporządkowany jest łuk grafu G_i skierowany z wierzchołka q_i do wierzchołka q_j a opisany literą $z_p \in Z_i$.

Z powyższych rozważań wynika, że graf G_i można określić następująco:

$$G_i = \langle Q_i, \bar{\Phi}_i, \Psi_i \rangle \quad (2.14)$$

gdzie Q_i jest zbiorem wierzchołków G_i , $\bar{\Phi}_i \subset Q_i \times Z_i \times Q_i$ jest zbiorem trójek uporządkowanych $\langle q_i, z_p, q_j \rangle$ przyporządkowanych łukom G_i ,

$\Psi_i \subset Q_i \times Y_i$ jest zbiorem par uporządkowanych $\langle q_i, y_s \rangle$ charakteryzującym obciążenie wierzchołków G_i literami Y_i .

Relacje $\bar{\Phi}_i$, Ψ_i spełniają ograniczenia wynikające z tego, że podautomat A_i jest automatem deterministycznym.

Żadna para wierzchołków grafów ze zbioru G^+ nie jest opisana tym samym symbolem q_i , to znaczy:

$$(\forall i, j \in I)(i \neq j \Rightarrow q_i \cap q_j = \emptyset) \quad (2.15)$$

Wierzchołki grafu podautomatu wchodzącego w skład danego automatu AZI można podzielić na wierzchołki wewnętrzne i zewnętrzne. Podział ten nie jest podziałem bezwzględnym, ale uzależnionym od automatu AZI zawierającego rozważany podautomat.

Wierzchołkiem wewnętrznym grafu podautomatu w danym automacie AZI jest taki wierzchołek, z którego wszystkie wychodzące łuki skierowane są do wierzchołków grafu tego podautomatu. Wierzchołkiem zewnętrznym grafu podautomatu w danym automacie AZI jest wierzchołek, z którego wychodzi co najmniej jeden łuk skierowany do wierzchołka grafu innego podautomatu wchodzącego w skład tego automatu AZI.

Niech każdy z podautomatów wyrażonych grafami ze zbioru G^+ (wyrażenie 2.5) zdolny będzie do realizacji pewnego podzadania. Łuk skierowany z danego grafu podautomatu wchodzącego w skład automatu AZI do grafu innego podautomatu opisany jest literą alfabetu wejścio-

wego automatu AZI powodującą przejście do realizacji innego podzadania w czasie realizacji pewnego zadania przez automat AZI. Niech grafy G_k, G_l będą grafami podautomatów pewnego automatu AZI. Każdemu z łuków łączących grafy różnych podautomatów można przyporządkować trójkę uporządkowaną postaci $\langle q_i, z_p, q_j \rangle$. W trójce $\langle q_i, z_p, q_j \rangle$ element q_i jest takim wierzchołkiem podgrafu G_k grafu rozpatrywanego automatu AZI, z którego wychodzi łuk opisany literą z_p i skierowany do wierzchołka q_j podgrafu będącego grafem G_l .

W zależności od zbioru rozważanych trójek uzyskuje się różne grafy automatów zagregowanych pierwszego rodzaju. Stąd w celu wykonania operacji $\oplus_j$ na grafach podautomatów niezbędne jest określenie zbioru tych trójek. Zbiór ten oznaczony jest literą S_j . Jeśli zbiór wyznaczający wynik operacji oznaczony jest literą S_j , to symbolem rozważanej operacji jest symbol $\oplus_j$.

W trójkach uporządkowanych $\langle q_i, z_p, q_j \rangle \in S_j$, pierwsza i trzecia współrzędna są wierzchołkami różnych grafów, to znaczy:

$$(\forall \langle q_i, z_p, q_j \rangle \in S_j) ((q_i \in Q_k \wedge q_j \in Q_l) \Rightarrow Q_k \neq Q_l) \quad (2.16)$$

gdzie Q_k, Q_l są zbiorami wierzchołków grafów G_k, G_l przynależnych do zbioru grafów poddanych operacji $\oplus_j$.

Z powyższego wynika, że łuki otrzymane w wyniku wykonania operacji $\oplus_j$ łączą wierzchołki różnych grafów podautomatów.

Z wierzchołka q_i grafu pewnego podautomatu do wierzchołków innych grafów podautomatów, w grafie automatu AZI, może być skierowanych więcej łuków niż jeden. Z wierzchołka q_i nie mogą wychodzić łuki opisane tą samą literą alfabetu wejściowego automatu AZI a skierowane do dwu różnych wierzchołków grafów innych podautomatów. Warunek ten zwany jest pierwszym warunkiem determinizmu dla operacji $\oplus_j$.

Warunek 2.1.

Pierwszy warunek determinizmu dla operacji $\oplus_j$:

$$(\forall \langle q_i, z_s, q_t \rangle, \langle q_j, z_p, q_r \rangle \in S_j) ((q_i = q_j \wedge z_s = z_p) \Rightarrow q_t = q_r) \quad (2.17)$$

Łuki skierowane z wierzchołka q_a grafu G_p podautomatu do grafu G_r innego podautomatu mogą być oznaczone tylko tymi literami, którymi nie są opisane łuki skierowane z wierzchołka q_a do wierzchołków grafu G_p . Warunek ten zwany jest drugim warunkiem determinizmu dla operacji $\oplus_j$.

Niech $Z_{q_j}^p$ będzie zbiorem liter alfabetu wejściowego automatu AZI, którymi oznaczone są łuki podgrafu G_p grafu tego automatu skierowane z wierzchołka q_j .

Natomiast niech $Z_{q_j}^{p,r}$ będzie zbiorem liter alfabetu wejściowego automatu AZI, którymi oznaczone są łuki skierowane z wierzchołka q_j podgrafu G_p grafu automatu AZI do podgrafu będącego grafem G_r .

Niech symbol $\text{crd}^2 \langle q_t, z_i, q_s \rangle$ oznacza drugą współrzędną trójki $\langle q_t, z_i, q_s \rangle$.

Zbiór $Z_{q_j}^{p,r}$ określony jest na podstawie zbioru S_j w następujący sposób:

$$Z_{q_j}^{p,r} = \{ z_i : z_i = \text{crd}^2 \langle q_j, z_i, q_s \rangle \wedge \langle q_j, z_i, q_s \rangle \in S_j \wedge q_s \in Q_r \} \quad (2.18)$$

gdzie Q_r jest zbiorem wierzchołków grafu G_r .

Niech zbiór $Z_{q_j}^{p,-}$ będzie określony w następujący sposób:

$$Z_{q_j}^{p,-} = \bigcup_{r \in J \setminus \{p\}} Z_{q_j}^{p,r}$$

Warunek 2.2.

Drugi warunek determinizmu dla operacji $\oplus_j$:

$$(\forall p \in J) (\forall q_j \in Q_p) (Z_{q_j}^p \cap Z_{q_j}^{p,-} = \emptyset) \quad (2.19)$$

gdzie J jest zbiorem dolnych indeksów symboli grafów podautomatów pod-

danych operacji $\oplus^{S_j}$, natomiast Q_p - zbiorem wierzchołków grafu G_p podautomatu A_p .

Warunek ten może być zapisany w następującej postaci:

$$(\forall p, r \in J)(\forall q_j \in Q_p)(Z_{q_j}^p \cap Z_{q_j}^{p,r} = \emptyset) \quad (2.20)$$

Niżej zostanie przedstawiona definicja operacji $\oplus^{S_j}$.

Graf $G_i \in \{G_j\}_{j \in J}$ określony jest za pomocą wyrażenia (2.14).

Graf G^j otrzymany w wyniku wykonania operacji $\oplus^{S_j}$ można określić następująco:

$$G^j = \langle \mathcal{Q}^j, \mathcal{G}^j, L^j \rangle \quad (2.21)$$

gdzie $\mathcal{Q}^j$ - zbiór wierzchołków grafu G^j , $\mathcal{G}^j$ - zbiór trójek uporządkowanych $\langle q_i, z_p, q_r \rangle$ przyporządkowanych łukom grafu G^j , L^j - zbiór par $\langle q_i, y_s \rangle$ charakteryzujący obciążenie wierzchołków grafu G^j .

Definicja 2.2.

Operacją $\oplus^{S_j}$ nazywa się taką operację wykonywaną na grafach ze zbioru $\{G_i\}_{i \in J}$, gdzie $J \subset I$, przy spełnieniu dwu warunków determinizmu dla tej operacji, w której wyniku zostaje uzyskany graf:

$$G^j = \oplus_{i \in J}^{S_j} G_i = \langle \mathcal{Q}^j, \mathcal{G}^j, L^j \rangle \quad (2.22)$$

gdzie

$$\mathcal{Q}^j = \bigcup_{i \in J} Q_i \quad (2.23)$$

$$\mathcal{G}^j = \left(\bigcup_{i \in J} \Phi_i \right) \cup S_j \quad (2.24)$$

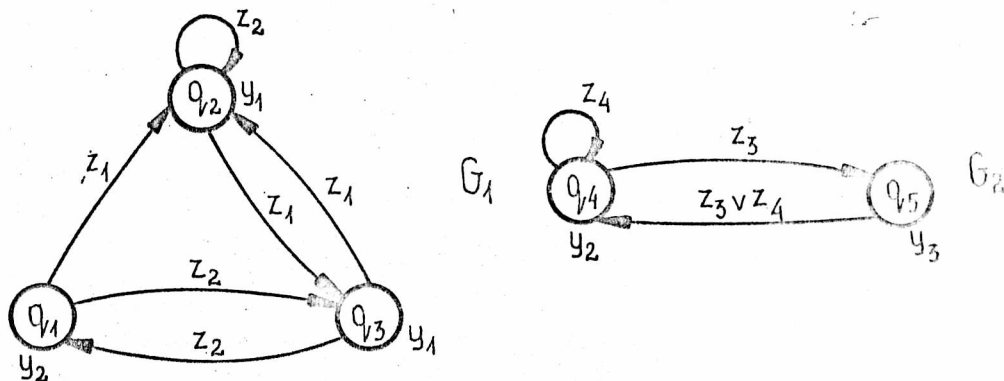
$$L^j = \bigcup_{i \in J} \Psi_i \quad (2.25)$$

Interpretacja zbiorów Θ^j , L^j jest prosta. Zbiór łuków grafu G^j zawiera łuki grafów ze zbioru $\{G_i\}_{i \in J}$ i łuki wyznaczone przez trójki uporządkowane zbioru S_j oraz nie zawiera innych elementów.

Dla przykładu rozpatrzona zostanie operacja $\bigoplus_{S_i}$ wykonywana na dwu grafach podautomatów.

Przykład 2.1.

Niech dane będą dwa grafy podautomatów przedstawione na rys. 2.3.



Rys. 2.3. Grafy G_1 , G_2 podautomatów

Niech dany będzie zbiór S_1 określony następująco:

$$S_1 = \{ \langle q_3, z_3, q_5 \rangle, \langle q_4, z_1, q_1 \rangle, \langle q_5, z_2, q_2 \rangle \} \quad (2.26)$$

Na grafach G_1 , G_2 zostaje wykonana następująca operacja:

$$G^i = \bigoplus_{j \in \{1,2\}}^{S_i} G_j \quad (2.27)$$

Z postaci zbioru S_1 wynika, że spełniony jest pierwszy warunek determinizmu dla operacji $\bigoplus_{S_i}$. Należy sprawdzić, czy spełniony jest drugi warunek determinizmu dla tej operacji. Każda trójka uporządkowana zbioru S_1 wyznacza element pewnego zbioru $Z_q^{p,x}$. Zbiory $Z_q^{p,x}$ generowane przez zbiór S_1 są następujące:

$$z_{q_3}^{1,2} = \{z_3\}$$

$$z_{q_4}^{2,1} = \{z_1\}$$

$$z_{q_5}^{2,1} = \{z_2\}$$

(2.28)

Dla każdego ze zbiorów typu $z_q^{p,r}$ należy wyznaczyć zbiór typu z_q^p .
Odpowiednie zbiory z_q^p mają w tym przykładzie następujące postaci:

$$z_{q_3}^1 = \{z_1, z_2\}$$

$$z_{q_4}^2 = \{z_3, z_4\}$$

$$z_{q_5}^2 = \{z_3, z_4\}$$

(2.29)

Iloczynny zbiorów które należy poddać analizie sprawdzając, czy spełniony jest drugi warunek determinizmu, określony za pomocą wyrażenia (2.20) są następujące:

$$z_{q_3}^1 \cap z_{q_3}^{1,2} = \emptyset$$

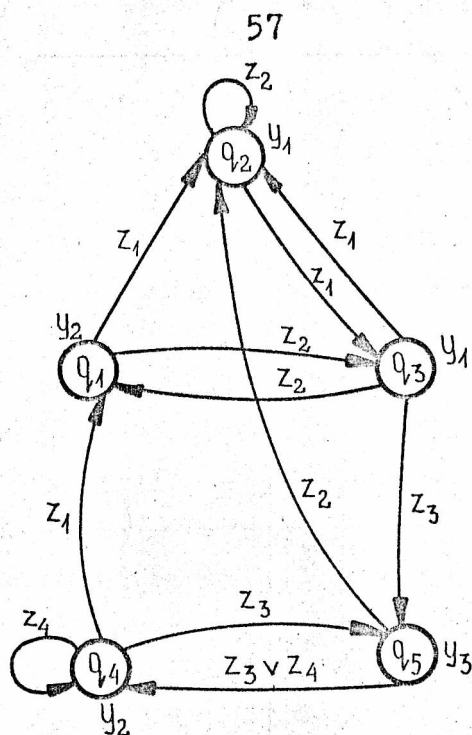
$$z_{q_4}^2 \cap z_{q_4}^{2,1} = \emptyset$$

$$z_{q_5}^2 \cap z_{q_5}^{2,1} = \emptyset$$

(2.30)

Stąd spełniony jest drugi warunek determinizmu dla operacji $\oplus_i$.

Graf G^i będący wynikiem wykonania operacji zadanej wyrażeniem (2.27) przedstawiony jest na rys. 2.4.



Rys. 2.4. Graf G^i

Niech symbol G^+ będzie symbolem wyrażenia analitycznego [25], [26] grafu G . Rozważana operacja na grafach ma odpowiednika w zbiorze operacji wykonywanych na wyrażeniach analitycznych. Operacja ta oznaczona zostaje symbolem $\oplus$. Jej zapis w rozważanym wyżej przykładzie ma postać:

$$G_1^+ \oplus_{S_1} G_2^+ = G^{i+} \quad (2.31)$$

Wynik wykonania operacji $\oplus_{S_j}$ można przedstawić w postaci znanej postacią zagregowaną. Postać zagregowana grafu G^j oznaczona jest symbolem ZG^j . Postać ZG^j jest grafem, w którym każdy wierzchołek reprezentuje sobą graf G_k będący elementem zbioru $\{G_i\}_{i \in J}$ grafów poddanych operacji $\oplus_{S_j}$. Łuki grafu ZG^j opisane są trójkami uporządkowanymi $\langle q_p, z_r, q_t \rangle \in S_j$ i skierowane są z wierzchołka reprezentującego graf zawierający wierzchołek q_p do wierzchołka reprezentującego graf, którego wierzchołkiem jest q_t .

Niech symbol $\langle G_s, \langle q_p, z_r, q_t \rangle, G_n \rangle$ będzie przyporządkowany łukowi grafu ZG^j skierowanemu z wierzchołka reprezentującego graf

G_s do wierzchołka reprezentującego graf G_n , opisanemu trójką uporządkowaną $\langle q_p, z_r, q_t \rangle \in S_j$.

Definicja 2.3.

Postacią zagregowaną grafu $G^j = \bigoplus_{i \in J}^{S_j} G_i$ nazywamy graf:

$$ZG^j = \langle X^j, U^j \rangle \quad (2.32)$$

gdzie

$$X^j = \{G_i\}_{i \in J} \quad (2.33)$$

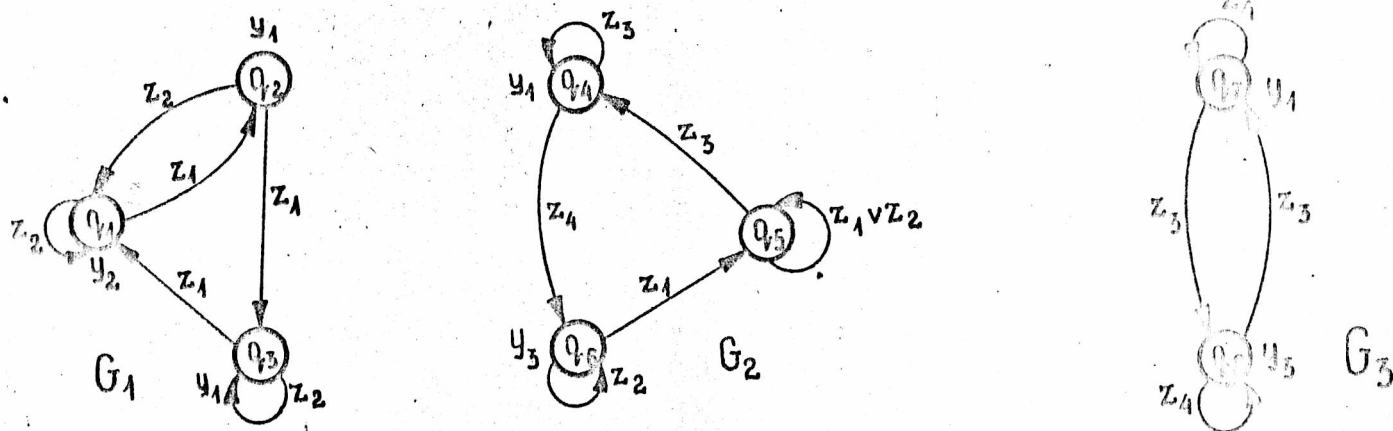
$$U^j = \{ \langle G_s, \langle q_p, z_r, q_t \rangle, G_n \rangle : \langle q_p, z_r, q_t \rangle \in S_j \wedge G_s, G_n \in \{G_i\}_{i \in J} \wedge q_p \in Q_s \wedge q_t \in Q_n \}$$

przy czym Q_s, Q_n są zbiorami wierzchołków grafów G_s, G_n . (2.34)

Postać zagregowaną grafu G^j można opisać za pomocą wyrażenia analitycznego. W celu zilustrowania postaci zagregowanej przedstawiony zostanie przykład.

Przykład 2.2.

Niech dane będą trzy grafy podautomatów przedstawione na rys. 2.5.



Rys. 2.5. Grafy G_1, G_2, G_3 podautomatów A_1, A_2, A_3

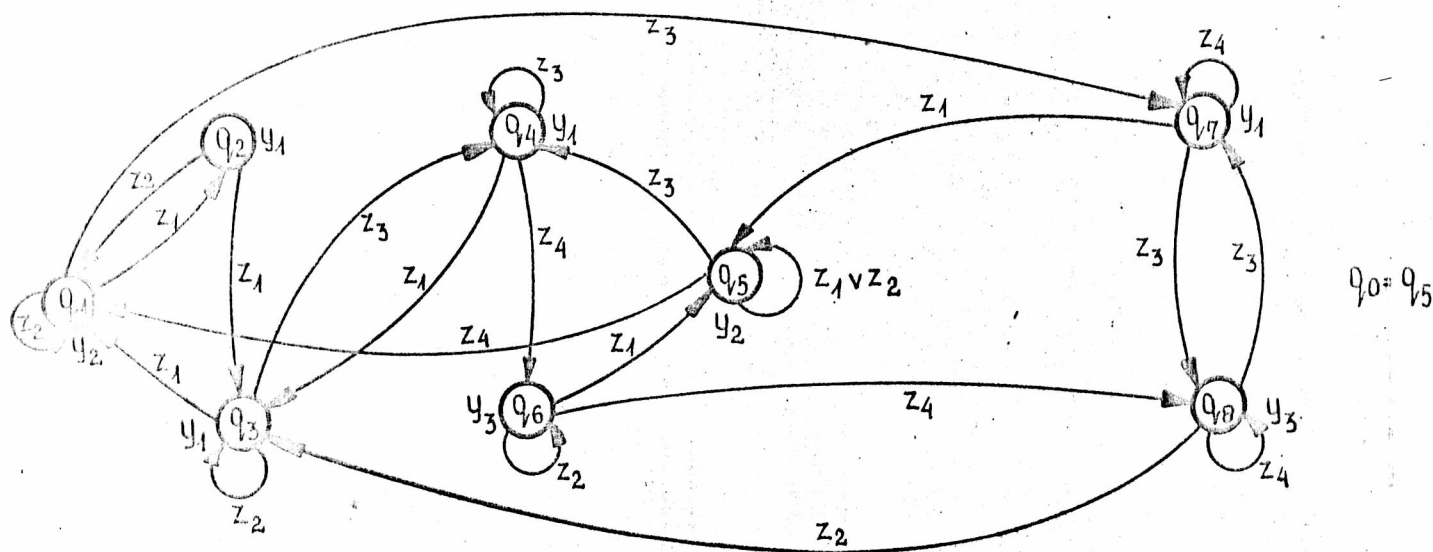
W celu wykonania operacji $\oplus_{S_c}$ na grafach G_1, G_2, G_3 niech dany będzie zbiór:

$$S_c = \{ \langle q_3, z_3, q_4 \rangle, \langle q_1, z_3, q_7 \rangle, \langle q_5, z_4, q_1 \rangle, \langle q_4, z_1, q_3 \rangle, \langle q_6, z_4, q_8 \rangle, \\ \langle q_8, z_2, q_3 \rangle, \langle q_7, z_1, q_5 \rangle \} \quad (2.35)$$

Niech symbol B oznacza zbiór $\{1, 2, 3\}$. Graf G^c zadany jest wyrażeniem:

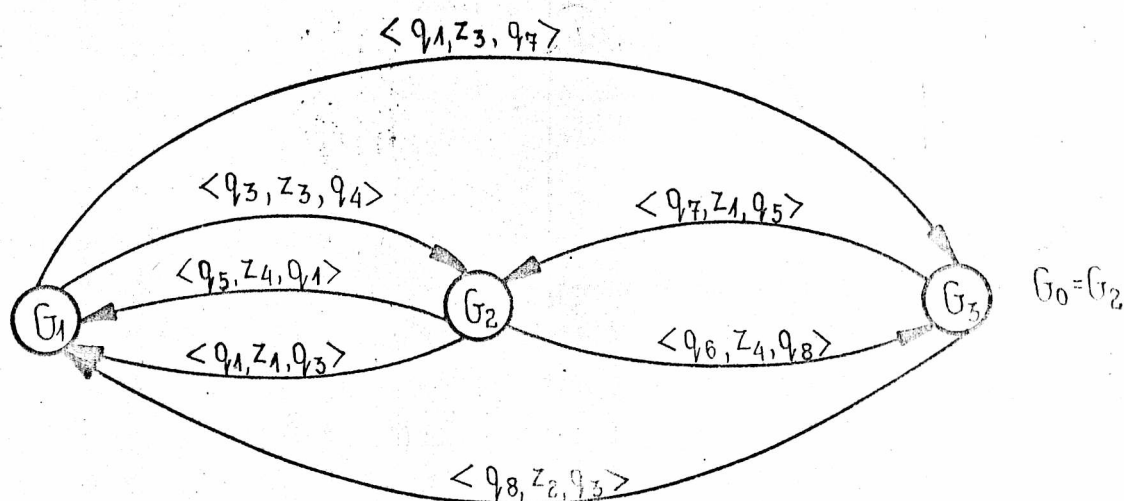
$$G^c = \bigoplus_{i \in B}^{S_c} G_i \quad (2.36)$$

Graf G^c przedstawiony jest na rys. 2.6.



Rys. 2.6. Graf G^c

Niech wierzchołkiem początkowym grafu G^c będzie wierzchołek q_5 grafu G_2 . Graf G^c można przedstawić w postaci zagregowanej ZG^c . Wierzchołkiem początkowym grafu ZG^c jest wierzchołek reprezentujący graf G_2 , ponieważ wierzchołkiem początkowym grafu G^c jest wierzchołek q_5 grafu G_2 . Postać ZG^c ilustruje rys. 2.7.



Rys. 2.7. Graf ZG^C - postać zagregowana grafu G^C

Graf ZG^C można przedstawić za pomocą wyrażenia analitycznego oznaczonego symbolem ZG^{C+} zaznaczając, że wierzchołkiem początkowym grafu G^C jest wierzchołek q_5 . Wyrażenie to ma następującą postać:

$$\begin{aligned}
 ZG^{C+} = & \begin{pmatrix} 0 & 1 & 2 \\ G_2 & \left(\begin{matrix} < q_6, z_4, q_8 > & G_3 \end{matrix} \right) & \begin{pmatrix} < q_7, z_1, q_5 > & G_2, < q_8, z_2, q_3 > \\ G_1 & \left(\begin{matrix} < q_3, z_3, q_4 > & G_2, < q_1, z_3, q_7 > & G_3 \end{matrix} \right)^3 & \begin{matrix} < q_4, z_1, q_3 > & G_1, \\ < q_5, z_4, q_1 > & G_1 \end{matrix} \end{pmatrix}^2 \\
 & \begin{pmatrix} 1 & 0 \\ & \end{pmatrix} \end{pmatrix} ; \quad q_0 = q_5 \quad (2.37)
 \end{aligned}$$

Wyrażenie ZG^{C+} umożliwia uproszczone zapisywanie grafu G^C .

Niżej zostanie podana definicja automatu zagregowanego pierwszego rodzaju.

Definicja 2.4.

Automatem zagregowanym pierwszego rodzaju nazywamy automat z wyróżnionym stanem zwanym początkowym, wyrażony grafem będącym wynikiem wykonania operacji $\oplus_{S_j}$ na podzbiorze zbioru grafów podautomatów odwzorowywanych działaniem automatu AP, przy czym $S_j \neq \emptyset$.

Z powyższej definicji wynika, że automat AZI jest systemem podautomatów przynależnych do różnych automatów odwzorowywanych działaniem

niem automatu A^j .

Niżej zostaną przedstawione twierdzenia charakteryzujące automaty zagregowane pierwszego rodzaju.

Automat zagregowany pierwszego rodzaju A^j można określić następująco:

$$A^j = \langle Z^j, Y^j, Q^j, q_0^j, \Phi^j, \Psi^j \rangle \quad (2.38)$$

gdzie Z^j - alfabet wejściowy A^j , Y^j - alfabet wyjściowy A^j , Q^j - zbiór stanów A^j , q_0^j - stan początkowy A^j , Φ^j jest relacją charakteryzującą przejścia między stanami A^j , Ψ^j jest relacją opisującą reakcje automatu A^j .

Niech symbol $\text{crd}^2 S_j$ oznacza zbiór drugich współrzędnych trójek $\langle q_i, z_p, q_j \rangle \in S_j$.

Twierdzenie 2.1.

Jeśli graf G^j automatu zagregowanego pierwszego rodzaju A^j zadany jest wyrażeniem:

$$G^j = \bigoplus_{i \in J}^{S_j} G_i$$

to automat A^j można określić następująco:

$$A^j = \langle Z^j, Y^j, Q^j, q_0^j, \Phi^j, \Psi^j \rangle$$

gdzie

$$Z^j = \left(\bigcup_{i \in J} Z_i \right) \cup \text{crd}^2 S_j \quad (1^0)$$

$$Y^j = \bigcup_{i \in J} Y_i \quad (2^0)$$

$$Q^j = \bigcup_{i \in J} Q_i \quad (3^0)$$

$$\Phi^j = \left(\bigcup_{i \in J} \Phi_i \right) \cup S_j \quad (4^0)$$

$$\Psi^j = \bigcup_{i \in J} \Psi_i \quad (5^0)$$

Dowód.

Alfabet wejściowy automatu A^j zawiera litery alfabetów wejściowych podautomatów ze zbioru $\{A_i\}_{i \in J}$ podautomatów oraz litery, którymi opisane są łuki otrzymane w wyniku wykonania operacji $\oplus$ i innych elementów nie zawiera. Stąd otrzymujemy wyrażenie (1^0) .

Graf G^j nie zawiera żadnych wierzchołków poza wierzchołkami grafów ze zbioru $\{G_i\}_{i \in J}$. Z powyższego i z wyrażenia (2.25) wynika wyrażenie (2^0) .

Z wyrażenia (2.23) otrzymujemy zależność (3^0) .

Zbiór $\mathcal{G}^j$ określony za pomocą wyrażenia (2.24) zawiera trójki $\langle q_i, z_p, q_j \rangle$ przyporządkowane łukom grafu G^j . Trójki te należą do zbioru Φ^j charakteryzującego przejścia między stanami automatu A^j . Ponieważ zbiór Φ^j innych elementów nie zawiera czyli $\Phi^j = \mathcal{G}^j$, a więc otrzymujemy wyrażenie (4^0) .

Dla zbioru Ψ^j określającego reakcje automatu A^j i funkcji L^j charakteryzującej obciążenie wierzchołków grafu G^j literami alfabetu Y^j prawdziwa jest zależność:

$$\Psi^j = L^j$$

Z powyższej równości i z wyrażenia (2.25) otrzymujemy zależność (5^0) . $\square$

Z wyrażenia (5^0) tezy twierdzenia 2.1. wynika następujący wniosek.

Wniosek 2.1.

Automat zagregowany pierwszego rodzaju jest automatem w sensie automatu Moore'a.

Twierdzenie 2.2.

Automat zagregowany pierwszego rodzaju jest automatem deterministycznym.

Dowód.

Relacja Φ^J dla automatu A^J określona jest wyrażeniem (4^0) tezy twierdzenia 2.1.

Podautomat A_i odwzorowywany działaniem automatu AP jest automatem deterministycznym. Stąd stwierdzamy:

$$(\forall \langle q_c, z_p, q_s \rangle, \langle q_t, z_r, q_a \rangle \in \Phi_i) ((q_c = q_t \wedge z_p = z_r) \Rightarrow q_s = q_a) \quad (1^0)$$

Ponieważ $J \subset I$, a więc na mocy wyrażenia (2.15) otrzymujemy:

$$(\forall i, j \in J) (i \neq j \Rightarrow q_i \cap q_j = \emptyset) \quad (2^0)$$

Z (2^0) wynika, że pierwsze współrzędne trójek będących elementami relacji Φ_i , Φ_j dla różnych podautomatów A_i , A_j są różne, to znaczy:

$$(\forall \langle q_c, z_p, q_s \rangle \in \Phi_i) (\forall \langle q_d, z_t, q_r \rangle \in \Phi_j) (i \neq j \Rightarrow q_c \neq q_d) \quad (3^0)$$

Z wyrażeń (1^0), (3^0) wynika, że dla zbioru $\Phi^{J'} = \bigcup_{i \in J} \Phi_i$

prawdziwa jest zależność:

$$(\forall \langle q_a, z_s, q_v \rangle, \langle q_p, z_r, q_u \rangle \in \Phi^{J'}) ((q_a = q_p \wedge z_s = z_r) \Rightarrow q_v = q_u) \quad (4^0)$$

Ponieważ zbiór S_j spełnia pierwszy warunek determinizmu dla operacji $\oplus^j$, a więc:

$$(\forall \langle q_i, z_s, q_t \rangle, \langle q_c, z_a, q_m \rangle \in S_j) ((q_i = q_c \wedge z_s = z_a) \Rightarrow q_t = q_m) \quad (5^0)$$

Z $(4^0), (5^0)$ i drugiego warunku determinizmu dla operacji $\oplus^j$ wynika, że dla zbioru $\Phi^j = \Phi^{j'} \cup S_j$ prawdziwe jest wyrażenie:

$$(\forall \langle q_a, z_s, q_v \rangle, \langle q_p, z_r, q_u \rangle \in \Phi^j) ((q_a = q_p \wedge z_s = z_r) \Rightarrow q_v = q_u) \quad (6^0)$$

Z (6^0) wynika, że spełniony jest warunek zadany wyrażeniem (2.12).

Relacja Ψ^j dla automatu A^j zadana jest wyrażeniem (5^0) tezy twierdzenia 2.1.

Ponieważ podautomat A_i jest automatem deterministycznym, a więc:

$$(\forall \langle q_c, y_p \rangle, \langle q_d, y_s \rangle \in \Psi_i) (q_c = q_d \Rightarrow y_p = y_s) \quad (7^0)$$

Z wyrażenia (2^0) przeprowadzanego dowodu wnioskujemy:

$$(\forall \langle q_c, y_p \rangle \in \Psi_i) (\forall \langle q_s, y_r \rangle \in \Psi_j) (i \neq j \Rightarrow q_c \neq q_s) \quad (8^0)$$

Z $(7^0), (8^0)$ otrzymujemy:

$$(\forall \langle q_c, y_p \rangle, \langle q_d, y_s \rangle \in \Psi^j) (q_c = q_d \Rightarrow y_p = y_s) \quad (9^0)$$

Z (9^0) wnioskujemy, że spełniony jest warunek zadany wyrażeniem (2.13).

Ponieważ automat A^j spełnia warunki określone za pomocą wyrażeń (2.12), (2.13), a więc jest automatem deterministycznym. $\square$

2.3.3. Własności automatów zagregowanych pierwszego rodzaju

Rozważania tego punktu będą dotyczyły zachowania automatów zagregowanych pierwszego rodzaju.

Dla grafu G automatu A zostanie zdefiniowany zbiór oznaczony symbolem $\bar{B}(G)$. W celu określenia tego zbioru należy wprowadzić definicję uogólnionej funkcji wyjść automatu.

Niech symbol $\bar{z}$ oznacza skończonej długości sekwencję liter z alfabetu Z automatu A , tzn. $\bar{z} \in Z^*$, gdzie Z^* jest zbiorem wszystkich słów skończonej długości uzyskanych z liter alfabetu Z . W podobny sposób interpretowany jest symbol $\bar{y}$.

Sekwencje $\bar{z}, \bar{y}$ mają następujące postaci:

$$\bar{z} = z_i \dots z_j \dots z_k \quad (2.39)$$

$$\bar{y} = y_p \dots y_r \dots y_s \quad (2.40)$$

gdzie $z_i, \dots, z_j, \dots, z_k \in Z, y_p, \dots, y_r, \dots, y_s \in Y$. Niech symbol $l|\bar{x}|$ oznacza długość sekwencji $\bar{x}$.

Dla automatów Moore'a uzasadnione jest analizowanie dwu zależności między długością sekwencji $\bar{z}$ a długością reakcji $\bar{y}$ automatu znajdującego się w stanie początkowym na sekwencję $\bar{z}$. Pierwsza z tych zależności określona jest następująco:

$$l|\bar{z}| = l|\bar{y}| \quad (2.41)$$

Z wyrażenia (2.41) wynika, że w przypadku pojawienia się sekwencji $\bar{z}$ na wejściu automatu A znajdującego się w stanie początkowym, kolejne litery sekwencji $\bar{y}$ są reakcjami na kolejno pojawiające się na wejściu automatu A litery sekwencji $\bar{z}$. Dla automatu Moore'a A jako pierwszą literę reakcji tego automatu można rozważać również literę alfabetu wyjściowego generowaną w stanie początkowym. Kolejne

litery sekwencji $\bar{y}$ są w tym przypadku reakcjami na kolejno pojawiające się na wejściu automatu A litery sekwencji $\bar{z}$. Zatem zamiast wyrażenia (2.41) można przyjąć następującą zależność:

$$l|\bar{z}| + 1 = l|\bar{y}| \quad (2.42)$$

Dalsze rozważania prowadzone będą dla przypadku równej długości sekwencji $\bar{z}$ i $\bar{y}$. Rozważania dla zależności (2.42) byłyby analogiczne.

Definicja 2.5.

Uogólnioną funkcją wyjść deterministycznego automatu $A = \langle Z, Y, Q, q_0, \Phi, \Psi \rangle$ nazywamy funkcję:

$$\bar{\Psi}: \{q_0\} \times Z^* \rightarrow Y^* \quad (2.43)$$

określającą reakcję $\bar{y} \in Y^*$ automatu A znajdującego się w stanie początkowym q_0 na sekwencję $\bar{z} \in Z^*$, przy czym $l|\bar{z}| = l|\bar{y}|$.

Zbiór $\bar{B}(G)$ dla automatu A zawiera pary uporządkowane, których pierwszą współrzędną jest sekwencja $\bar{z} \in Z^*$, natomiast drugą współrzędną - reakcja automatu A na sekwencję $\bar{z}$, określona za pomocą uogólnionej funkcji wyjść $\bar{\Psi}$.

Definicja 2.6.

$$\bar{B}(G) = \{ \langle \bar{z}, \bar{y} \rangle : \bar{z} \in Z^* \wedge \bar{y} \in Y^* \wedge \bar{y} = \bar{\Psi}(q_0, \bar{z}) \} \quad (2.44)$$

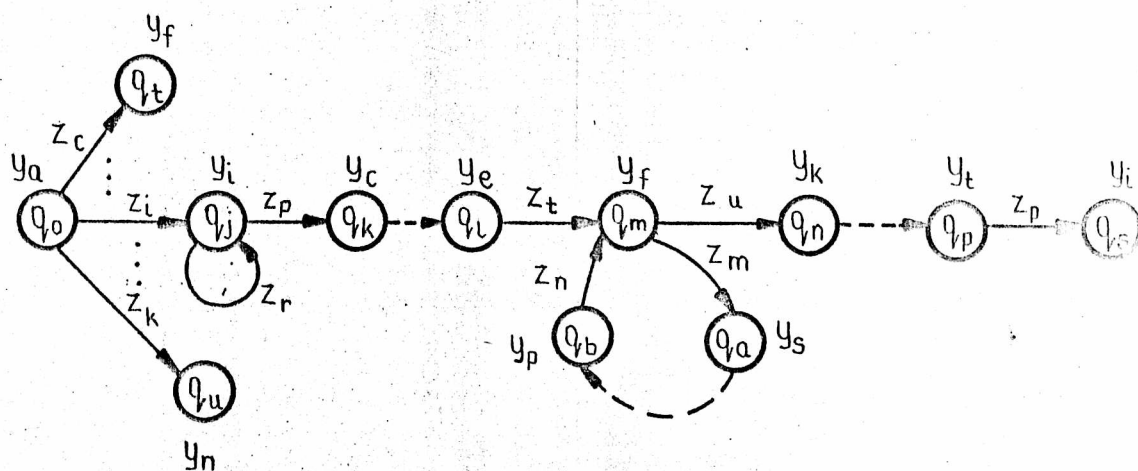
gdzie G jest grafem deterministycznego automatu A.

Zbiór $\bar{B}(G)$ charakteryzuje zachowanie automatu A.

Niech dany będzie graf G automatu A. Każda para $\langle \bar{z}, \bar{y} \rangle \in \bar{B}(G)$ związana jest z pewną drogą w grafie G. Wierzchołkiem początkowym tej drogi jest wierzchołek q_0 . Łuki rozważanej drogi wyznaczone są przez

elementy sekwencji $\bar{z}$. Droga ta wyznaczona jest jednoznacznie, ponieważ po podaniu litery $z_j \in Z$ na wejście automatu A , będącego automatem deterministycznym i znajdującego się w stanie q_p , kolejnym stanem może być tylko jeden stan. Wierzchołki tej drogi opisane są tymi literami alfabetu wyjściowego, z których zbudowana jest sekwencja $\bar{y}$.

Możliwe jest wielokrotne przechodzenie automatu przez jeden stan w wyniku podania na jego wejście sekwencji $\bar{z}$. Niech w grafie G istnieje droga przedstawiona na rys. 2.8.



Rys. 2.8. Fragment grafu G automatu A

Niech na wejście automatu A podana będzie następująca sekwencja:

$$\bar{z} = z_1 z_r^m z_p \dots z_t (z_m \dots z_n)^k z_u \dots z_p \quad (2.45)$$

gdzie symbol z_r^m oznacza sekwencję składającą się z m liter z_r , natomiast symbol $(z_m \dots z_n)^k$ jest symbolem sekwencji składającej się z k sekwencji $z_m \dots z_n$. Po podaniu na wejście automatu A znajdującego się w stanie q_0 sekwencji z opisanej wyrażeniem (2.45), automat ten $k+1$ razy będzie znajdował się w stanie q_m . Po raz pierwszy wspomniane zdarzenie nastąpi po podaniu sekwencji $z_1 z_r^m z_p \dots z_t$. Kolejne k przejść wspomnianego automatu przez stan q_m wywołane będą k -krotnym podaniem na wejściu tego automatu znajdującego się w sta-

nie q_m sekwencji $z_m \dots z_n$. Każda para $\langle \bar{z}, \bar{y} \rangle$ związana jest tylko z jedną drogą w grafie G , ponieważ rozważany automat jest automatem deterministycznym. Z daną drogą może być związanych wiele sekwencji. Na przykład ze wzmiankowaną drogą w grafie G związane są sekwencje $\bar{z}$ opisane wyrażeniem (2.45), w których liczby m, k przyjmują wartości ze zbioru $\{0, 1, \dots\}$.

Istnieją w grafie automatu drogi związane tylko z jedną sekwencją wejściową. Drogi te pozbawione są cykli i pętli.

Z podsekwencjami danych sekwencji liter alfabetu wejściowego automatu A związane są poddrogi odpowiednich dróg w grafie G . Przykładowo z podsekwencją $z_1 z_r^m z_p$ sekwencji $\bar{z}$ opisanej wyrażeniem (2.45) związana jest droga rozpoczynająca się w wierzchołku q_0 a zawierająca łuki opisane literami z_1, z_r, z_p .

Wierzchołkiem początkowym rozważanych w tym podrozdziale dróg jest wierzchołek q_0 . W celu skrócenia zapisu w dalszej analizie nie będzie to podkreślane.

Uwaga:

Dla wielu automatów, nie wszystkie litery alfabetu wejściowego mogą pojawić się na wejściu automatu znajdującego się w wybranym stanie. Na przykład na wejściu automatu wyrażonego grafem przedstawionym na rys. 1.5. i znajdującego się w stanie q_2 może pojawić się tylko litera w_0 , natomiast litery w_1, w_2, w_3 - nie mogą się pojawić. Stąd nie jest uzasadnionym uzupełnienie wymienionego grafu o pętlę wokół wierzchołka q_2 opisaną literami w_1, w_2, w_3 . Powyższa uwaga dotyczy również innych automatów, a nie tylko automatów przyporządkowanych sieciom działań. Automaty o wspomnianej własności nazywane są automatami z ograniczonym wejściem [18]. Przeprowadzone rozważania wskazują, że nie wszystkie sekwencje skończonej długości, które można uzyskać na bazie alfabetu wejściowego automatu mogą być pierwszymi współrzędnymi par $\langle \bar{z}, \bar{y} \rangle$. Prowadzona dalej analiza dotyczyć będzie

par $\langle \bar{z}, \bar{y} \rangle$ wyznaczonych przez drogi w grafie automatu. Dla automatu o grafie G' , zbiór tych par oznaczony jest symbolem $B(G')$.

Niech symbol $Z(G')$ oznacza zbiór tych sekwencji o skończonej długości z alfabetu wejściowego automatu opisanego grafem G' , które są związane z drogami w grafie G' rozpoczynającymi się w wierzchołku przyporządkowanym stanowi początkowemu tego automatu. Niech $\bar{\Psi}'$ będzie uogólnioną funkcją wyjść rozważanego automatu, natomiast Y' - jego alfabetem wyjściowym.

Definicja 2.7.

$$B(G') = \{ \langle \bar{z}, \bar{y} \rangle : \bar{z} \in Z(G') \wedge \bar{y} \in Y'^* \wedge \bar{y} = \bar{\Psi}'(q_0, \bar{z}) \}$$

gdzie G' jest grafem automatu deterministycznego.

Grafowi G' automatu deterministycznego można przyporządkować zbiór $E(G')$ dróg tego grafu, których wierzchołkiem początkowym jest wierzchołek q_0 . Droga będąca elementem zbioru $E(G')$ oznaczona jest symbolem e_k .

Niżej zostanie przedstawiona definicja d-równości dróg będących drogami różnych grafów automatów deterministycznych.

Droga $e_p \in E(G')$ wyznacza zbiór $U_p \subset B(G')$ par uporządkowanych $\langle \bar{z}, \bar{y} \rangle$. Zależność ta zostaje oznaczona symbolem $e_p \rightarrow U_p$.

Definicja 2.8.

Drogi e_k, e_s dwu różnych grafów automatów deterministycznych nazywamy d-równymi ($e_k \stackrel{d}{=} e_s$), jeśli dla zbiorów U_k, U_s wyznaczonych przez te drogi prawdziwa jest równość:

$$U_k = U_s. \quad (2.46)$$

Przykładem drogi d-równej drodze przedstawionej na rys. 2.8. jest droga różniąca się od wymienionej drogi tylko sposobem określe-

nia wierzchołków symbolami q_c .

Dla rozważań prowadzonych w tym punkcie istotne jest pojęcie d-równości dla dróg grafów automatów zagregowanych pierwszego rodzaju. Na bazie jednego zbioru grafów $\{G_i\}_{i \in J}$ podautomatów odwzorowywanych działaniem automatu AP można uzyskać dużo różnych automatów zagregowanych pierwszego rodzaju. Stąd wierzchołki grafów tych automatów określone są tymi samymi symbolami q_j . Szczególnym przypadkiem dróg d-równych są drogi nie różniące się nawet sposobem określenia wierzchołków symbolami q_j .

Niech graf G_1 podautomatu A_1 wchodzi w skład grafów G^i, G^j automatów zagregowanych pierwszego rodzaju. Zatem w grafach G^i, G^j istnieją pewne drogi będące drogami grafu G_1 . Drogi te są nie tylko d-równymi ale ich wierzchołki określone są w taki sam sposób symbolami q_j .

Z kolei zostanie przedstawiona definicja relacji d-zawierania między zbiorami dróg grafów automatów deterministycznych oznaczonej symbolem $\overset{d}{\subset}$.

Dla grafów G', G'' prawdziwa jest relacja $E(G') \overset{d}{\subset} E(G'')$, jeśli dla każdego elementu zbioru $E(G')$ istnieje d-równy mu element zbioru $E(G'')$.

Definicja 2.9.

Dla zbiorów dróg grafów G', G'' automatów deterministycznych prawdziwa jest relacja d-zawierania:

$$E(G') \overset{d}{\subset} E(G'') \quad (2.47)$$

jeśli prawdziwe jest zdanie:

$$(\forall e_k \in E(G')) (\exists e_s \in E(G'')) (e_k \overset{d}{=} e_s) \quad (2.48)$$

Niżej zostanie przedstawiony lemat charakteryzujący relację d-zawierania.

Lemat 2.1.

$$Z.E(G') \stackrel{d}{\subset} E(G'')$$

$$T.B(G') \subset B(G'')$$

D.

Droga e_k wyznacza zbiór U_k par $\langle \bar{z}, \bar{y} \rangle$.

Zbiory $B(G')$, $B(G'')$ można określić następująco:

$$B(G') = \bigcup \{ U_k : e_k \rightarrow U_k \wedge e_k \in E(G') \} \quad (1^0)$$

$$B(G'') = \bigcup \{ U_k : e_k \rightarrow U_k \wedge e_k \in E(G'') \} \quad (2^0)$$

Z $(1^0), (2^0)$, założenia i definicji relacji d-zawierania wynika, że:

$$B(G') \subset B(G'')$$

□

Związek par $\langle \bar{z}, \bar{y} \rangle$ z drogami w grafie automatu zagregowanego jest podstawą do formułowania twierdzeń, które będą podane w dalszej części tego punktu. W celu określenia założeń tych twierdzeń przedstawiony zostanie lemat charakteryzujący warunki determinizmu dla operacji $\bigoplus_{S_j}$.

Niech symbol:

$$\text{Det} (S_j, \{G_i\}_{i \in J}) \quad (2.49)$$

oznacza, że zbiór S_j trójek uporządkowanych i zbiór $\{G_i\}_{i \in J}$ grafów podautomatów odwzorowywanych działaniem automatu AP spełniają dwa warunki determinizmu dla operacji $\bigoplus_{S_j}$, tzn., że automat o grafie $G^j = \bigoplus_{i \in J} G_i$ jest automatem deterministycznym.

Zbiory $Z_{q_j}^{p,-}$ określone w punkcie 2.3.2. pracy można na podstawie zbioru S_j wyznaczyć w następujący sposób:

$$Z_{q_j}^{p,-} = \{z_1 : z_1 = \text{crd}^2 \langle q_j, z_1, q_r \rangle \wedge \langle q_j, z_1, q_r \rangle \in S_j\} \quad (2.50)$$

gdzie symbol $\text{crd}^2 \langle q_t, z_1, q_r \rangle$ oznacza drugą współrzędną trójki uporządkowanej $\langle q_t, z_1, q_r \rangle$. Na przykład niech jedynymi elementami zbioru S_k , których pierwszą współrzędną jest $q_j \in Q_p$ będą następujące trójki:

$$\langle q_j, z_1, q_m \rangle, \langle q_j, z_n, q_d \rangle \quad (2.51)$$

Zbiór $Z_{q_j}^{p,-}$ określony jest w tym przypadku następująco:

$$Z_{q_j}^{p,-} = \{z_1, z_n\} \quad (2.52)$$

Prowadzone w tym punkcie rozważania dotyczą różnych zbiorów S_i . Zbiór typu zbioru $Z_{q_j}^{p,-}$ wyznaczony na podstawie zbioru S_i będzie oznaczony symbolem $Z_{q_j}^{p,i}$ (i).

Niżej zostanie przedstawiony lemat charakteryzujący warunki determinizmu dla operacji $\oplus_j^{S_j}$. Z wymienionego lematu wynika, że jeśli zbiory S_n i $\{G_i\}_{i \in J}$ spełniają warunki determinizmu dla operacji $\oplus_n^{S_n}$, to zbiory S_m i $\{G_i\}_{i \in J}$, gdzie $S_m \subset S_n$, spełniają warunki determinizmu dla operacji $\oplus_m^{S_m}$.

Lemat 2.2.

Z. $S_m \subset S_n \wedge \text{Det}(S_n, \{G_i\}_{i \in J})$

T. $\text{Det}(S_m, \{G_i\}_{i \in J})$

D.

W pierwszej części dowodu udowodnimy, że zbiory $S_m, \{G_i\}_{i \in J}$ spełniają pierwszy warunek determinizmu dla operacji $\oplus_m^{S_m}$.

Z założenia mamy:

$$\text{Det} (S_n, \{G_i\}_{i \in J}) \quad (1^0)$$

Z (1^0) wynika, że spełniony jest pierwszy warunek determinizmu dla operacji $\oplus^{S_n}$, a więc prawdziwe jest zdanie:

$$(\forall \langle q_i, z_s, q_t \rangle, \langle q_j, z_p, q_r \rangle \in S_n) ((q_i = q_j \wedge z_s = z_p) \Rightarrow q_t = q_r) \quad (2^0)$$

Ponieważ $S_m \subset S_n$, a więc z (2^0) otrzymujemy:

$$(\forall \langle q_s, z_p, q_n \rangle, \langle q_z, z_s, q_t \rangle \in S_m) ((q_s = q_z \wedge z_p = z_s) \Rightarrow q_n = q_t) \quad (3^0)$$

czyli zbiory $S_m, \{G_i\}_{i \in J}$ spełniają pierwszy warunek determinizmu dla operacji $\oplus^{S_m}$.

Z (1^0) wynika, że zbiory $S_n, \{G_i\}_{i \in J}$ spełniają drugi warunek determinizmu dla operacji $\oplus^{S_n}$, czyli dla zbiorów $z_{q_j}^{p,-(n)}$ prawdziwe jest zdanie:

$$(\forall p \in J) (\forall q_j \in Q_p) (z_{q_j}^p \cap z_{q_j}^{p,-(n)} = \emptyset) \quad (4^0)$$

Ponieważ $S_m \subset S_n$, a więc na podstawie wyrażenia (2.50) stwierdzamy, że:

$$(\forall p \in J) (\forall q_j \in Q_p) (z_{q_j}^{p,-(m)} \subset z_{q_j}^{p,-(n)}) \quad (5^0)$$

Z (4^0) , (5^0) wnioskujemy, że prawdziwe jest zdanie:

$$(\forall p \in J) (\forall q_j \in Q_p) (z_{q_j}^p \cap z_{q_j}^{p,-(m)} = \emptyset) \quad (6^0)$$

tzn., że spełniony jest drugi warunek determinizmu dla operacji $\oplus^{S_m}$.

Z powyższych rozważań otrzymujemy:

$$\text{Det} \left(S_m, \{ G_i \}_{i \in J} \right)$$

Z lematów 2.1., 2.2. wynikają następujące twierdzenia.

Twierdzenie 2.3.

$$Z. S_m \neq \emptyset \wedge S_m \subset S_n \wedge \text{Det} \left(S_n, \{ G_i \}_{i \in J} \right)$$

$$T. B \left(\bigoplus_{i \in J}^m G_i \right) \subset B \left(\bigoplus_{i \in J}^n G_i \right)$$

D.

Z założenia i lematu 2.2. wynika, że

$$\text{Det} \left(S_m, \{ G_i \}_{i \in J} \right) \quad (1^0)$$

Z założenia wynika, że zbiory S_m, S_n są zbiorami niepustymi. Stąd wobec spełnienia warunków determinizmu przez zbiory S_m i $\{ G_i \}_{i \in J}$ oraz przez zbiory S_n i $\{ G_i \}_{i \in J}$ stwierdzamy, że istnieją automaty zagregowane pierwszego rodzaju o grafach będących wynikiem wykonania operacji $\bigoplus^m, \bigoplus^n$ na grafach ze zbioru $\{ G_i \}_{i \in J}$ grafów podautomatów odwzorowywanych działaniem automatu AP, wyrażone grafami:

$$G^J = \bigoplus_{i \in J}^{S_m} G_i \quad (2^0)$$

$$G^K = \bigoplus_{i \in J}^{S_n} G_i \quad (3^0)$$

Z relacji $S_m \subset S_n$ oraz definicji operacji $\bigoplus^{S_j}$ i definicji relacji d-zawierania wnioskujemy, że

$$E(G^J) \stackrel{d}{\subseteq} E(G^k) \quad (4^0)$$

$Z(2^0)$, (3^0) , (4^0) i lematu 2.1. wynika, że

$$B\left(\bigoplus_{i \in J}^{S_m} G_i\right) \subset B\left(\bigoplus_{i \in J}^{S_n} G_i\right) \quad \blacksquare$$

Twierdzenie 2.4.

$$Z. S_m \neq \emptyset \wedge S_n \neq \emptyset \wedge S = S_m \cup S_n \wedge \text{Det}(S, \{G_i\}_{i \in J})$$

$$T. B\left(\bigoplus_{i \in J}^{S_m} G_i\right) \cup B\left(\bigoplus_{i \in J}^{S_n} G_i\right) \subset B\left(\bigoplus_{i \in J}^S G_i\right)$$

D.

Na mocy założenia i lematu 2.2. stwierdzamy, że dla zbioru

$\{A_i\}_{i \in J}$ podautomatów odwzorowywanych działaniem automatu AP można rozpatrywać automaty zagregowane pierwszego rodzaju wyrażone grafami:

$$G^J = \bigoplus_{i \in J}^{S_m} G_i \quad (1^0)$$

$$G^k = \bigoplus_{i \in J}^{S_n} G_i \quad (2^0)$$

$$G^D = \bigoplus_{i \in J}^S G_i \quad (3^0)$$

Z zależności $S_m \subset S$ i $S_n \subset S$ oraz twierdzenia 2.3. wynika, że

$$B\left(\bigoplus_{i \in J}^{S_m} G_i\right) \subset B\left(\bigoplus_{i \in J}^S G_i\right) \quad (4^0)$$

$$B\left(\bigoplus_{i \in J}^{S_n} G_i\right) \subset B\left(\bigoplus_{i \in J}^S G_i\right) \quad (5^0)$$

Z (4^0) , (5^0) wnioskujemy, że

$$B\left(\bigoplus_{i \in J}^{S_m} G_i\right) \cup B\left(\bigoplus_{i \in J}^{S_n} G_i\right) \subset B\left(\bigoplus_{i \in J}^S G_i\right) \quad \blacksquare$$

Przy spełnieniu założenia twierdzenia 2.4. ogólnie nie jest prawdziwa następująca zależność:

$$B\left(\bigoplus_{i \in J}^S G_i\right) \subset B\left(\bigoplus_{i \in J}^{S_m} G_i\right) \cup B\left(\bigoplus_{i \in J}^{S_n} G_i\right) \quad (2.53)$$

Stwierdzenie to zostanie uzasadnione za pomocą przykładu.

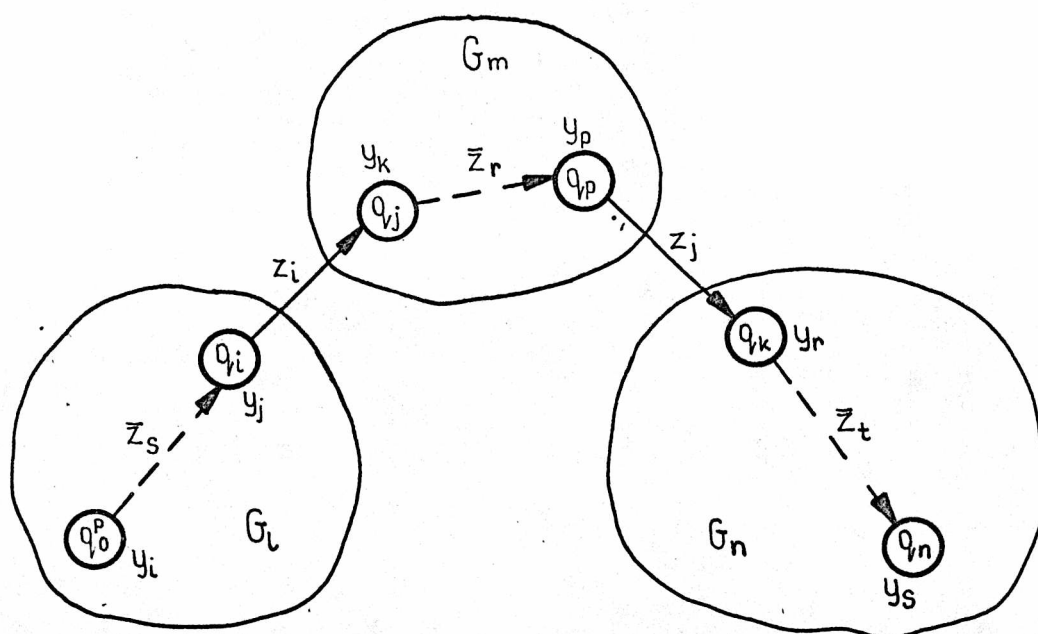
Przykład 2.3.

Niech fragment grafu $G^p = \bigoplus_{i \in J}^S G_i$ ma postać przedstawioną na rys. 2.9.

Przyjmujemy, że:

$$\langle q_i, z_i, q_j \rangle \in S_m \wedge \langle q_p, z_j, q_k \rangle \notin S_m \quad (2.54)$$

$$\langle q_i, z_i, q_j \rangle \notin S_n \wedge \langle q_p, z_j, q_k \rangle \in S_n \quad (2.55)$$

Rys. 2.9. Fragment grafu G^p

Dwie rozważane wyżej trójki uporządkowane są elementami zbioru

$$S = S_m \cup S_n.$$

W grafie G^p istnieje droga, której łuki opisane są literami następującej sekwencji:

$$\bar{z}_v = \bar{z}_s z_i \bar{z}_r z_j \bar{z}_t \quad (2.56)$$

Drogi, której łuki opisane są literami sekwencji $\bar{z}_v$ nie ma w grafie $G^j = \bigoplus_{i \in J}^m G_i$ oraz drogi o tej własności nie ma w grafie $G^k = \bigoplus_{i \in J}^n G_i$, co można uzasadnić przeprowadzając następujące rozważania.

Przyjmujemy, że droga opisana literami sekwencji $\bar{z}_v$ istnieje w grafie G^j . Ponieważ $\langle q_p, z_j, q_k \rangle \notin S_m$, a więc zbiór S_m musi zawierać element $\langle q_p, z_j, q_s \rangle$, przy czym $q_k \neq q_s$ i wierzchołek q_s musi być początkiem drogi grafu G^j opisanej literami sekwencji $\bar{z}_t$.

Jeśli $\langle q_p, z_j, q_s \rangle \in S_m$, to trójka ta jest elementem zbioru S zawierającego również trójkę $\langle q_p, z_j, q_k \rangle$. Z ostatniego zdania wynika, że

zbiór S nie spełnia pierwszego warunku determinizmu dla operacji $\oplus$, co jest sprzeczne z założeniem twierdzenia 2.4. Stąd w grafie G^j nie istnieje droga, której łuki opisane są literami sekwencji $\bar{z}_v$. Podobna analiza przeprowadzona dla grafu G^k jest podstawą do sformułowania wniosku, że w tym grafie również nie ma drogi opisanej literami sekwencji $\bar{z}_v$. Ponieważ drogi opisanej literami sekwencji $\bar{z}_v$ nie ma w żadnym z grafów G^j, G^k , a więc ogólnie nie jest prawdziwa zależność opisana wyrażeniem (2.53).

Twierdzenie 2.5.

$$Z. S \neq \emptyset \wedge S = S_m \cap S_n \wedge \text{Det}(S_m, \{G_i\}_{i \in J}) \wedge \text{Det}(S_n, \{G_i\}_{i \in J})$$

$$T. B\left(\bigoplus_{i \in J} G_i\right) \subset B\left(\bigoplus_{i \in J} G_i\right) \cap B\left(\bigoplus_{i \in J} G_i\right)$$

D. Z założenia i lematu 2.2. wnioskujemy, że można rozpatrywać automaty zagregowane pierwszego rodzaju opisane grafami:

$$G^j = \bigoplus_{i \in J}^{S_m} G_i \quad (1^\circ)$$

$$G^k = \bigoplus_{i \in J}^{S_n} G_i \quad (2^\circ)$$

$$G^r = \bigoplus_{i \in J}^S G_i \quad (3^\circ)$$

Ponieważ $S = S_m \cap S_n$, a więc

$$S \subset S_m \quad (4^\circ)$$

$$S \subset S_n \quad (5^\circ)$$

Z (4°), (5°) i twierdzenia 2.3. wynika, że

$$B \left(\bigoplus_{i \in J}^{S} G_i \right) \subset B \left(\bigoplus_{i \in J}^{S_m} G_i \right) \quad (6^\circ)$$

$$B \left(\bigoplus_{i \in J}^{S} G_i \right) \subset B \left(\bigoplus_{i \in J}^{S_n} G_i \right) \quad (7^\circ)$$

Na podstawie (6°), (7°) stwierdzamy, że

$$B \left(\bigoplus_{i \in J}^{S} G_i \right) \subset B \left(\bigoplus_{i \in J}^{S_m} G_i \right) \cap B \left(\bigoplus_{i \in J}^{S_n} G_i \right) \quad \blacksquare$$

Przy spełnieniu założenia twierdzenia 2.5., ogólnie nie jest prawdziwe wyrażenie:

$$B \left(\bigoplus_{i \in J}^{S_m} G_i \right) \cap B \left(\bigoplus_{i \in J}^{S_n} G_i \right) \subset B \left(\bigoplus_{i \in J}^{S} G_i \right) \quad (2.57)$$

Niech dane będzie następujące zdanie:

$$S_n \neq \emptyset \wedge S \neq \emptyset \wedge S = S_m \setminus S_n \wedge \text{Det}(S_m, \{G_i\}_{i \in J}) \wedge \text{Det}(S_n, \{G_i\}_{i \in J}) \quad (2.58)$$

Jeśli zdanie opisane wyrażeniem (2.58) jest prawdziwe, to może nie być prawdziwa żadna z dwu następujących relacji:

$$B \left(\bigoplus_{i \in J}^{S} G_i \right) \subset B \left(\bigoplus_{i \in J}^{S_m} G_i \right) \setminus B \left(\bigoplus_{i \in J}^{S_n} G_i \right) \quad (2.59)$$

$$B \left(\bigoplus_{i \in J}^{S_m} G_i \right) \setminus B \left(\bigoplus_{i \in J}^{S_n} G_i \right) \subset B \left(\bigoplus_{i \in J}^S G_i \right) \quad (2.60)$$

Twierdzenia tego punktu można uogólnić rezygnując z ograniczenia, zgodnie z którym zbiory S , S_m , S_n są zbiorami niepustymi. Wtedy podane twierdzenia będą dotyczyły nie tylko automatów zagregowanych pierwszego rodzaju, których grafy są wynikiem wykonania operacji $\bigoplus^{S_j}$ na grafach podautomatów, przy założeniu $S_j \neq \emptyset$.

2.3.4. Synteza formalna automatów zagregowanych drugiego rodzaju

Podstawą do uzyskiwania automatów zagregowanych drugiego rodzaju są podautomaty odwzorowywane działaniem automatu AP - wyrażone grafami ze zbioru G^+ opisanego zależnością (2.5). W wyniku wykonania operacji $\bigoplus^{S_j}$, przedstawionej w punkcie 2.3.2. pracy, otrzymujemy graf G^j . Graf podautomatu jest podgrafem grafu G^j . Nie zawsze można graf automatu podzielić na takie części nie mające wspólnych wierzchołków, jakimi są podgrafy tego grafu będące grafami podautomatów. Na rys. 2.2. w punkcie 2.3.1. pracy przedstawiono graf G' automatu, w którym pewien graf podautomatu nie jest podgrafem grafu G' , ale jest tylko podgrafem cząstkowym grafu rozważanego automatu. W celu uzyskania takiego grafu automatu na bazie zbioru L podautomatów, w którym istnieją podgrafy cząstkowe izomorficzne z grafami podautomatów, nie będące podgrafami grafu tego automatu zostanie wprowadzona operacja oznaczona symbolem Δ^{T_i} . Operacja ta wykonywana jest na grafie G^j , a wynikiem jej wykonania jest graf G^k automatu zagregowanego drugiego rodzaju.

Operacja Δ^{T_i} polega na nakładaniu na siebie wierzchołków grafu G^j w celu uzyskania wierzchołka grafu automatu zagregowanego drugiego rodzaju. Symbol T_i jest symbolem zbioru charakteryzującego sposób nakładania na siebie wierzchołków grafu G^j . Aby operacja Δ^{T_i} mogła

być wykonana, niezbędne jest spełnienie przez zbiór T_i i graf G^j trzech warunków, które będą przedstawione w dalszej części tego punktu.

W celu skrócenia zapisów, automat zagregowany drugiego rodzaju oznaczony będzie symbolem AZII.

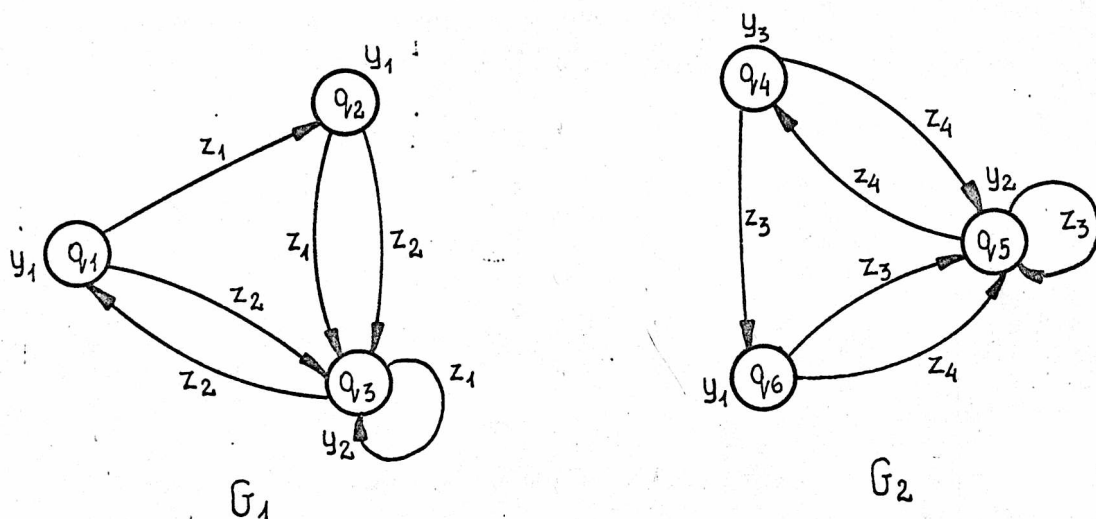
Proces, w którego wyniku zostaje uzyskany graf G^j nazywany jest pierwszym etapem syntezy formalnej automatu AZII, natomiast proces otrzymywania grafu G^k z grafu G^j - drugim etapem syntezy formalnej wymienionego automatu.

Wierzchołek grafu automatu AZI wyznaczony jest przez tylko jeden wierzchołek grafu pewnego podautomatu ze zbioru L (wyrażenie 2.4.) . Wierzchołek grafu automatu AZII może być wyznaczony przez zadany podzbiór zbioru wierzchołków grafu G^j , będącego wynikiem wykonania operacji $\oplus$ na grafach ze zbioru $\{G_i\}_{i \in J}$. Niżej zostanie przedstawiony, za pomocą przykładu, sposób nakładania na siebie wierzchołków dwu grafów podautomatów bez uwzględnienia operacji $\oplus$.

Przykład 2.4.

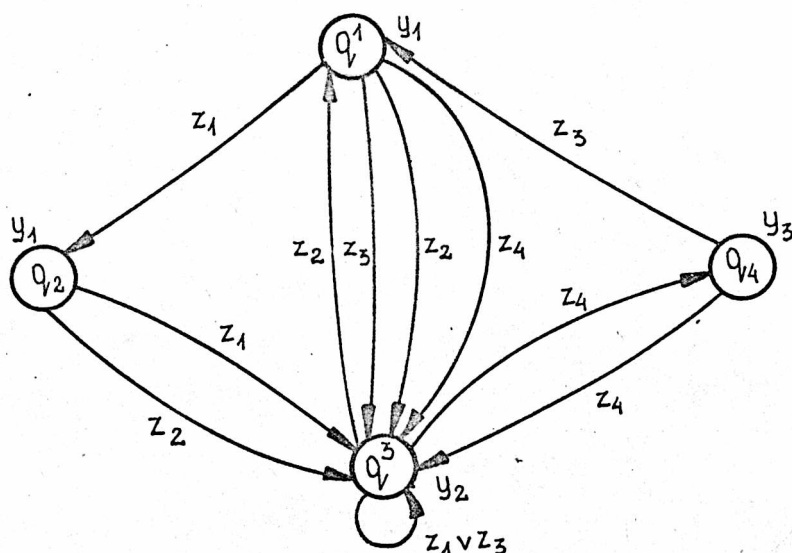
Niech dane będą dwa grafy podautomatów przedstawione na rys.

2.10.



Rys. 2.10. Grafy G_1 , G_2 podautomatów

Niech wierzchołki q_1 i q_6 wyznaczają wierzchołek grafu G^* pewnego automatu AZII oznaczony symbolem q^1 , a otrzymany w wyniku nałożenia na siebie wierzchołków q_1 , q_6 . Inny wierzchołek grafu G^* oznaczony symbolem q^3 niech będzie wyznaczony przez wierzchołki q_3 i q_5 . W wyniku odpowiedniego nałożenia na siebie wierzchołków grafów G_1 , G_2 otrzymujemy graf G^* przedstawiony na rys. 2.11.



Rys. 2.11. Graf G^*

Graf G_1 z rys. 2.10 jest izomorficzny z takim podgrafem cząstkowym grafu G^* , którego wierzchołkami są wierzchołki q^1, q_2, q^3 . Graf G_1 nie jest podgrafem cząstkowym grafu G^* , ponieważ zbiór wierzchołków G_1 równy $\{q_1, q_2, q_3\}$ nie jest identyczny ze zbiorem $\{q^1, q_2, q^3\}$. Graf G_2 jest izomorficzny z takim podgrafem cząstkowym grafu G^* , którego wierzchołkami są wierzchołki q^1, q^3, q_4 .

Z wierzchołkiem q_2 grafu G^* incydentne są te łuki podgrafu cząstkowego grafu G^* izomorficznego z grafem G_1 , które przyporządkowane są łukom grafu G_1 incydentnym z wierzchołkiem q_2 grafu G_1 . W podobny sposób określone są łuki grafu G^* incydentne z wierzchołkiem q_4 tego grafu.

Z wierzchołkiem q^1 grafu G^* wyznaczonym przez zbiór:

$$\bar{q}^1 = \{q_1, q_6\} \quad (2.61)$$

incydentne są te łuki podgrafu cząstkowego grafu G^* izomorficznego z grafem G_1 , które przyporządkowane są łukom grafu G_1 incydentnym z wierzchołkiem q_1 grafu G_1 i te łuki podgrafu cząstkowego grafu G^* izomorficznego z grafem G_2 , które są przyporządkowane łukom grafu G_2 incydentnym z wierzchołkiem q_6 grafu G_2 . W podobny sposób określone są łuki grafu G^* incydentne z wierzchołkiem q^3 tego grafu wyznaczonym przez zbiór:

$$\bar{q}^3 = \{q_3, q_5\} \quad (2.62)$$

Wierzchołek q_2 grafu G^* przyporządkowany jest wierzchołkowi q_2 grafu G_1 , natomiast wierzchołek q_4 grafu G^* - wierzchołkowi q_4 grafu G_2 . O wierzchołkach q_2 grafu G_1 i q_4 grafu G_2 mówimy, że samodzielnie wyznaczają wierzchołki grafu G^* .

Ogólnie jeśli tylko jednemu wierzchołkowi q_i grafu G^j , uzyskanego w wyniku pierwszego etapu syntezy automatu AZII, przyporządkowany jest wierzchołek q_i grafu G^k będącego wynikiem drugiego etapu syntezy tego automatu; to mówimy, że wierzchołek q_i grafu G^j samodzielnie wyznacza wierzchołek q_i grafu G^k .

W celu wykonania operacji Δ^{τ_i} na grafie G^j należy wyszczególnić te zbiory wierzchołków grafu G^j , które wyznaczają wierzchołki grafu G^k . Nie jest uzasadnione wymienianie tych wierzchołków grafu G^j , które samodzielnie wyznaczają wierzchołki grafu G^k .

Zbiór tych wierzchołków $q_1, \dots, q_j, \dots, q_k$ grafu G^j , które wyznaczają pewien wierzchołek grafu G^k oznaczony jest symbolem $\bar{q}^i$ to znaczy:

$$\bar{q}^i = \{q_1, \dots, q_j, \dots, q_k\} \quad (2.63)$$

Górny indeks symbolu $\bar{q}^i$ jest indeksem jednego z symboli $q_1, \dots, \dots, q_j, \dots, q_k$. Zbiór $\bar{q}^i$ wyznacza wierzchołek grafu G^k oznaczony symbolem q^i .

Zbiór tych zbiorów wierzchołków grafu G^j , które wyznaczają wierzchołki grafu G^k , z pominięciem zbiorów zawierających tylko jeden element samodzielnie wyznaczający wierzchołek grafu G^k , oznaczony jest symbolem T_i . Jest to zbiór następującej postaci:

$$T_i = \{ \bar{q}_v^a, \dots, \bar{q}_v^c, \dots, \bar{q}_v^n \} \quad (2.64)$$

W przykładzie 2.4. zbiór T_i ma następującą postać:

$$T_i = \{ \{q_1, q_6\}, \{q_3, q_5\} \} \quad (2.65)$$

Operacje prowadzące do uzyskania grafu automatu AZII wykonywane są na zbiorze $\{G_i\}_{i \in J}$ grafów podautomatów odwzorowywanych działaniem automatu AP.

W wyniku pierwszego etapu syntezy formalnej automatu AZII otrzymujemy graf G^j zadany za pomocą wyrażeń (2.22), (2.23), (2.24), (2.25), przy czym zbiór S_j i zbiór $\{G_i\}_{i \in J}$ grafów podautomatów odwzorowywanych działaniem automatu AP spełniają dwa warunki determinizmu dla operacji $\oplus$ przedstawione w punkcie 2.3.2. pracy.

Na drugim etapie syntezy automatu AZII wykonywana jest operacja nakładania na siebie wierzchołków grafu G^j oznaczona symbolem Δ^{T_i} , w wyniku czego otrzymujemy graf G^k automatu zagregowanego drugiego rodzaju.

W celu skrócenia zapisów dla wierzchołków q_t grafu G^j samodzielnie wyznaczających wierzchołki grafu G^k , przyjmujemy oznaczenie:

$$\bar{q}_v^t = \{q_t\} \quad (2.66)$$

a więc możemy pisać $q_t \in \bar{q}_v^t$.

Zbiorem wierzchołków grafu G^k jest zbiór składający się z wierzchołków wyznaczonych przez zbiory będące elementami zbioru T_i oraz z tych wierzchołków grafu G^j , które samodzielnie wyznaczają wierzchołki grafu G^k . Łuk grafu G^j skierowany z wierzchołka q_i do wierzchołka q_k wyznacza łuk grafu G^k skierowany z wierzchołka wyznaczonego przez zbiór $\bar{q}_v^1$ zawierający q_i do wierzchołka wyznaczonego

przez zbiór $\bar{q}^j$ zawierający q_k . Ze względu na oznaczenie określone wyrażeniem (2.66), w szczególnym przypadku $q_i \in \bar{q}^i$ lub $q_k \in \bar{q}^k$.

Na przykład łuk grafu G_1 z rys. 2.10. skierowany z wierzchołka q_1 do q_2 opisany literą z_1 wyznacza łuk grafu G^* z rys. 2.11. skierowany z wierzchołka q^1 do wierzchołka q_2 , opisany literą z_1 .

Aby operacja Δ^{T_i} była wykonalna, muszą być spełnione trzy warunki zwane warunkami determinizmu dla tej operacji.

Niżej zostanie przedstawiony pierwszy warunek determinizmu dla operacji Δ^{T_i} .

Niech wierzchołek q_c grafu G^j będzie elementem zbiorów $\bar{q}^p, \bar{q}^r$ oraz niech w grafie G^j do wierzchołka q_c skierowany będzie łuk z wierzchołka q_a opisany literą z_p . Niech wierzchołek q_a grafu G^j samodzielnie wyznacza wierzchołek q_a grafu G^k wyrażającego automat A^k . Jeśli na wejściu automatu A^k znajdującego się w stanie q_a pojawiłaby się litera z_p , to automat ten mógłby osiągnąć stan q^p lub q^r . W celu uniknięcia podobnych niejednoznaczności musi być spełniony warunek zwany pierwszym warunkiem determinizmu dla operacji Δ^{T_i} .

Warunek 2.3.

Pierwszy warunek determinizmu dla operacji Δ^{T_i} :

$$(\forall \bar{q}^p, \bar{q}^r \in T_i) (\bar{q}^p \neq \bar{q}^r \Rightarrow \bar{q}^p \cap \bar{q}^r = \emptyset) \quad (2.67)$$

Z kolei zostanie przedstawiony drugi warunek determinizmu dla operacji Δ^{T_i} .

Niech wierzchołek q^i grafu G^k będzie wyznaczony przez zbiór $\bar{q}^i$ zawierający wierzchołki q_j, q_k . Niech w grafie G^j istnieją dwa łuki opisane jedną i tą samą literą z_p , z których jeden skierowany jest z wierzchołka q_j do wierzchołka q_p , a drugi - z wierzchołka q_k do q_r . Szczególnym przypadkiem łuku jest pętla skierowana z wierzchołka q_j do q_j . Aby był spełniony drugi warunek determinizmu dla operacji Δ^{T_i} wierzchołki q_p, q_r muszą być elementami pewnego zbioru $\bar{q}^s \in T_i$ wyznającego wierzchołek q^s grafu G^k . Gdyby powyższy warunek nie był

spełniony, to z wierzchołka q^i grafu G^k skierowane byłyby łuki do różnych wierzchołków tego grafu opisane literą z_p .

Graf G^j można określić następująco:

$$G^j = \langle Q^j, \Gamma^j, L^j \rangle \quad (2.68)$$

gdzie znaczenie poszczególnych współrzędnych zadane jest za pomocą definicji 2.2.

Warunek 2.4.

Drugi warunek determinizmu dla operacji Δ^{T_i} :

$$(\forall \langle q_j, z_s, q_c \rangle, \langle q_k, z_t, q_r \rangle \in \mathcal{G}^j) ((q_j \in \bar{q}_v^i \wedge q_k \in \bar{q}_v^i \wedge z_s = z_t) \Rightarrow (\exists \bar{q}_v^s \in T_i)(q_c, q_r \in \bar{q}_v^s)) \quad (2.69)$$

Niżej zostanie przedstawiony trzeci warunek determinizmu dla operacji Δ^{T_i} .

Każdy z wierzchołków grafu G^k może być opisany tylko jedną literą alfabetu wyjściowego Y^k automatu wyrażonego grafem G^k . Stąd każdy element zbioru $\bar{q}_v^i \in T_i$ musi być opisany tą samą literą alfabetu wyjściowego Y^j automatu wyrażonego grafem G^j . Warunek ten jest trzecim warunkiem determinizmu dla operacji Δ^{T_i} .

Z wyrażenia (2.25) wynika, że sposób opisanie wierzchołków grafu G^j literami alfabetu Y^j określony jest za pomocą funkcji charakteryzujących obciążenie wierzchołków grafów przynależnych do zbioru $\{G_i\}_{i \in J}$.

Zbiór Q_i jest zbiorem wierzchołków grafu G_i podautomatu A_i , natomiast Ψ_i - funkcją charakteryzującą obciążenie wierzchołków grafu G_i literami alfabetu wyjściowego Y_i tego podautomatu.

Warunek 2.5.

Trzeci warunek determinizmu dla operacji Δ^{T_i} :

$$(\forall \bar{q}_v^i \in T_i)(\forall q_j, q_k \in \bar{q}_v^i)((q_j \in Q_s \wedge q_k \in Q_t) \Rightarrow \Psi_s(q_j) = \Psi_t(q_k)) \quad (2.70)$$

Niżej zostanie przedstawiona definicja operacji Δ^{T_i} .

W celu skrócenia zapisów w tych miejscach, w których nie będzie istotne czy wierzchołek grafu G^k wyznaczony jest przez zbiór $\bar{q}_v^i \in T_i$, czy przez wierzchołek grafu G^j samodzielnie wyznaczający wierzchołek grafu G^k , dla oznaczenia wierzchołków grafu G^k stosowane będą symbole $q', q'', \dots$.

Jeśli wierzchołek q^i grafu G^k wyznaczony jest przez zbiór $\bar{q}_v^i$ zawierający m. in. wierzchołek q_j , to możemy napisać $q_j \in \bar{q}_v^i$. W celu skrócenia zapisów dla tych wierzchołków grafu G^j , które samodzielnie wyznaczają wierzchołki grafu G^k będziemy stosowali oznaczenie określone wyrażeniem (2.66).

Graf G^k można przedstawić w następujący sposób:

$$G^k = \langle \Theta^k, \mathcal{G}^k, L^k \rangle \quad (2.71)$$

gdzie Θ^k - zbiór wierzchołków q' grafu G^k , $\mathcal{G}^k$ - zbiór trójek uporządkowanych $\langle q', z_p, q' \rangle$ przyporządkowanych łukom grafu G^k , L^k - zbiór par uporządkowanych $\langle q', y_s \rangle$ charakteryzujący obciążenie wierzchołków grafu G^k .

Zbiór Θ^k zawiera wierzchołki wyznaczone przez zbiory $\bar{q}_v^i$ wierzchołków grafu G^j będące elementami zbioru T_i oraz wierzchołki przyporządkowane tym wierzchołkom grafu G^j , które samodzielnie wyznaczają wierzchołki grafu G^k .

Zbiór $\mathcal{G}^k$ zawiera trójki uporządkowane $\langle q', z_p, q' \rangle$ przy czym $q', q'' \in \Theta^k$.

Niech prawdziwe będzie zdanie:

$$q_1, q_j \in \bar{q} \wedge \langle q_1, z_s, q_n \rangle \in \mathcal{G}^j \wedge \langle q_j, z_s, q_m \rangle \in \mathcal{G}^j \quad (2.72)$$

Aby był spełniony drugi warunek determinizmu dla operacji $\Delta_i^{T_i}$, musi być prawdziwe zdanie:

$$(\exists \bar{q}'' \in T_i)(q_n, q_m \in \bar{q}'') \quad (2.73)$$

Trójkom $\langle q_i, z_s, q_n \rangle, \langle q_j, z_s, q_m \rangle$ przyporządkowanym dwu łukom grafu G^j przyporządkowana jest trójka $\langle q', z_s, q'' \rangle \in \mathcal{G}^k$.

Zbiór L^k zawiera pary $\langle q', y_s \rangle$. Niech $q_i, q_j \in \bar{q}'$. Z trzeciego warunku determinizmu dla operacji Δ wynika, że wierzchołki q_i, q_j grafu G^j muszą być opisane tą samą literą y_s to znaczy $\langle q_i, y_s \rangle, \langle q_j, y_s \rangle \in L^j$. Parom $\langle q_i, y_s \rangle, \langle q_j, y_s \rangle$ przyporządkowana jest para $\langle q', y_s \rangle \in L^k$.

Definicja 2.10.

Operacją Δ nazywa się taką operację wykonywaną na grafie G^j otrzymanym w wyniku przeprowadzenia operacji $\oplus$ na zbiorze $\{G_i\}_{i \in J}$ grafów, przy spełnieniu trzech warunków determinizmu dla operacji Δ , w której wyniku zostaje uzyskany graf:

$$G^k = \Delta^{\mathcal{T}_i} G^j = \langle \mathcal{Q}^k, \mathcal{G}^k, L^k \rangle \quad (2.74)$$

gdzie

$$\mathcal{Q}^k = \{q^i : \bar{q}^i \in \mathcal{T}_i\} \cup (\mathcal{Q}^j \setminus \bigcup_{\bar{q}^i \in \mathcal{T}_i} \bar{q}^i) \quad (2.75)$$

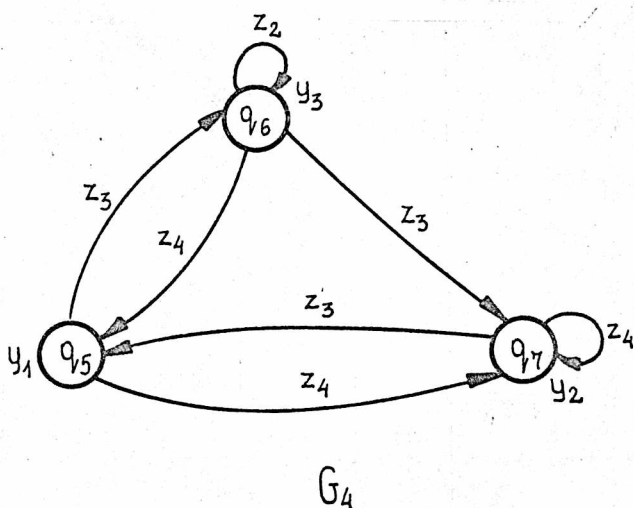
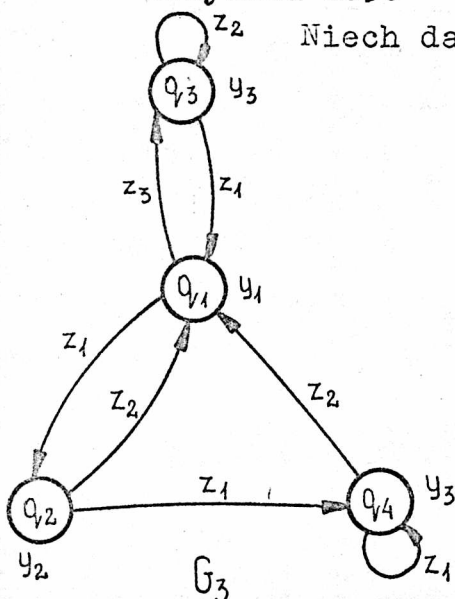
$$\mathcal{G}^k = \{ \langle q', z_p, q'' \rangle : q_i \in \bar{q} \wedge q_n \in \bar{q}'' \wedge q', q'' \in \mathcal{Q}^k \wedge \langle q_i, z_p, q_n \rangle \in \mathcal{G}^j \} \quad (2.76)$$

$$L^k = \{ \langle q', y_s \rangle : q_i \in \bar{q} \wedge q' \in \mathcal{Q}^k \wedge \langle q_i, y_s \rangle \in L^j \} \quad (2.77)$$

Operacja Δ zostanie zilustrowana przykładem.

Przykład 2.5.

Niech dane będą grafy przedstawione na rys.2.12.



Rys. 2.12. Grafy G_3, G_4

Niech zbiór S_r określony będzie w sposób następujący:

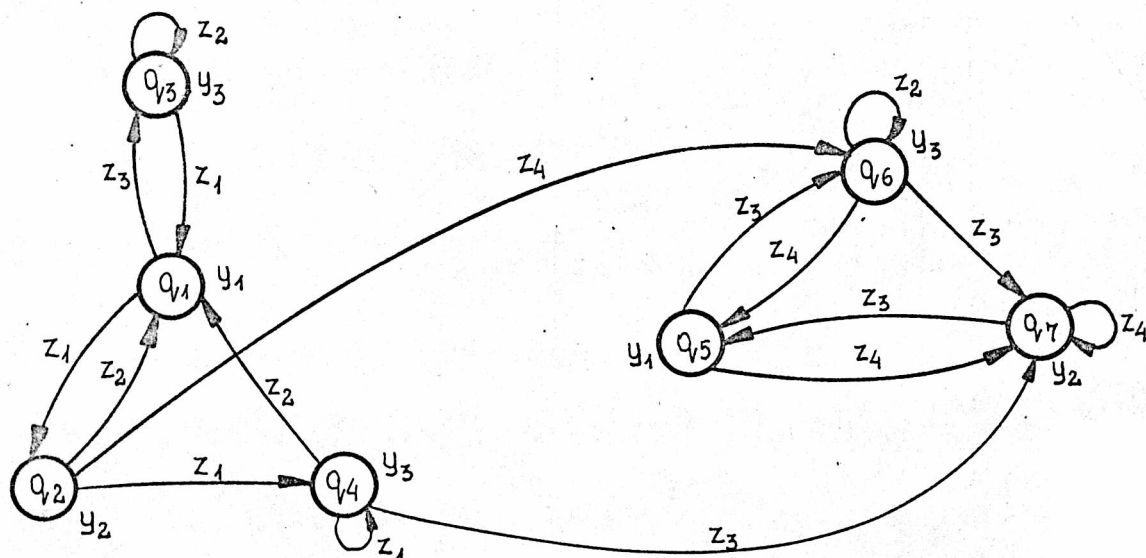
$$S_r = \{ \langle q_2, z_4, q_6 \rangle, \langle q_4, z_3, q_7 \rangle \} \quad (2.78)$$

Z postaci grafów G_3 , G_4 i zbioru S_r wynika, że spełnione są warunki determinizmu dla operacji $\oplus_r$.

Graf G^r określony w następujący sposób:

$$G^r = \bigoplus_{i \in \{3,4\}}^{S_r} G_i \quad (2.79)$$

przedstawiony jest na rys. 2.13.



Rys. 2.13. Graf G^r

Niech zbiór T_r będzie określony w następujący sposób:

$$T_r = \{ \bar{q}^1, \bar{q}^3 \} \quad (2.80)$$

gdzie

$$\bar{q}^1 = \{q_1, q_5\}, \quad \bar{q}^3 = \{q_3, q_6\} \quad (2.81)$$

Ponieważ

$$\bar{q}^1 \cap \bar{q}^3 = \emptyset \quad (2.82)$$

a więc spełniony jest pierwszy warunek determinizmu dla operacji Δ^r .

Z kolei należy sprawdzić, czy spełniony jest drugi warunek determinizmu dla tej operacji.

W pierwszej kolejności sprawdzamy, czy z wierzchołka q^1 grafu G^r nie są skierowane łuki do różnych wierzchołków tego grafu, opisa-

ne tą samą literą z_p .

Niech symbol Zq_i oznacza zbiór liter, którymi opisane są łuki grafu G^r skierowane z wierzchołka q_i .

Zbiory Zq_1, Zq_5 dla wierzchołków $q_1, q_5 \in \bar{q}^1$ mają następujące postaci:

$$\begin{aligned} Zq_1 &= \{z_1, z_3\} \\ Zq_5 &= \{z_3, z_4\} \end{aligned} \quad (2.83)$$

Na podstawie powyższych zbiorów zostaje wyznaczony zbiór $Zq_{1,5}$ określony następująco:

$$Zq_{1,5} = Zq_1 \cap Zq_5 = \{z_3\} \quad (2.84)$$

Łuk grafu G^r skierowany z wierzchołka q_1 a opisany literą z_3 skierowany jest do wierzchołka q_3 . Łuk grafu G^r skierowany z wierzchołka q_5 a opisany literą z_3 skierowany jest do wierzchołka q_6 . Dla spełnienia drugiego warunku determinizmu dla operacji Δ^r wierzchołki q_3, q_6 muszą być elementami tego samego zbioru będącego elementem zbioru T_r . Postać zbioru T_r wskazuje, że warunek ten jest spełniony.

Dla wierzchołka q^3 przeprowadzamy analizę podobną do rozważań przeprowadzonych dla wierzchołka q^1 .

Zbiory Zq_3, Zq_6 mają następujące postaci:

$$\begin{aligned} Zq_3 &= \{z_1, z_2\} \\ Zq_6 &= \{z_2, z_3, z_4\} \end{aligned} \quad (2.85)$$

Zbiór $Zq_{3,6}$ określony jest w sposób następujący:

$$Zq_{3,6} = Zq_3 \cap Zq_6 = \{z_2\} \quad (2.86)$$

Łuk grafu G^r skierowany z wierzchołka q_3 a opisany literą z_2 skierowany jest do wierzchołka q_3 . Łuk grafu G^r skierowany z wierzchołka q_6 a opisany literą z_2 skierowany jest do wierzchołka q_6 .

Z powyższych rozważań wynika, że spełniony jest drugi warunek determinizmu dla operacji Δ^r .

Następnie należy sprawdzić czy spełniony jest trzeci warunek

determinizmu dla tej operacji.

Z postaci grafów G_3 , G_4 wynikają następujące równości:

$$\Psi_3(q_1) = y_1 \quad (2.87)$$

$$\Psi_3(q_3) = y_3$$

$$\Psi_4(q_5) = y_1$$

$$\Psi_4(q_6) = y_3$$

Stąd prawdziwe są równości:

$$\Psi_3(q_1) = \Psi_4(q_5) \quad (2.88)$$

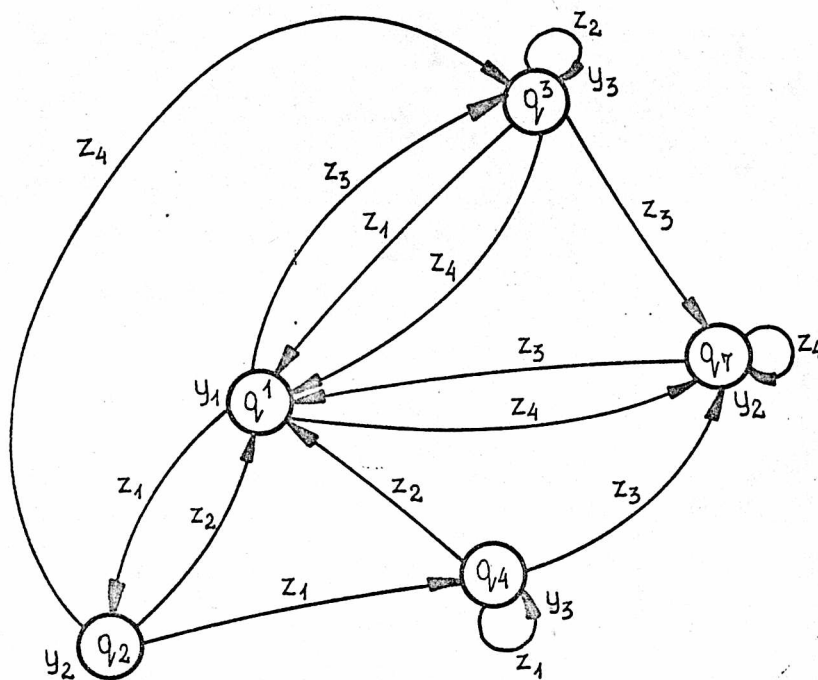
$$\Psi_3(q_3) = \Psi_4(q_6) \quad (2.89)$$

Z wyrażeń (2.88), (2.89) i postaci zbioru T_r wynika, że spełniony jest trzeci warunek determinizmu dla operacji Δ_r .

Graf G^p określony następująco:

$$G^p = \Delta_r G^r \quad (2.90)$$

przedstawiony jest na rys. 2.14.



Rys. 2.14. Graf G^p

Niżej zostanie podana definicja automatu zagregowanego drugiego rodzaju.

Definicja 2.11.

Automatem zagregowanym drugiego rodzaju nazywamy automat z wy-
różnionym stanem zwanym stanem początkowym, wyrażony grafem będącym
wynikiem wykonania operacji Δ^{T_i} na grafie G^j zadany definicją 2.2.,
przy czym $T_i \neq \emptyset$.

Z powyższej definicji wynika, że automat AZII jest systemem pod-
automatów przynależnych do różnych automatów odwzorowywanych działa-
niem automatu AP.

Z kolei zostaną przedstawione twierdzenia charakteryzujące auto-
maty zagregowane drugiego rodzaju.

Automat A^j wyrażony grafem G^j określonym za pomocą definicji
2.2. można przedstawić następująco:

$$A^j = \langle Z^j, Y^j, Q^j, q_0^j, \Phi^j, \Psi^j \rangle \quad (2.91)$$

gdzie poszczególne współrzędne określone są za pomocą twierdzenia 2.1.

Twierdzenie 2.6.

Jeśli graf G^k automatu zagregowanego drugiego rodzaju A^k zadany
jest wyrażeniem:

$$G^k = \bigoplus_{i \in J} \Delta^{T_i} \oplus^{S_j} G_i$$

to automat A^k można określić następująco:

$$A^k = \langle Z^k, Y^k, Q^k, q_0^k, \Phi^k, \Psi^k \rangle$$

gdzie

$$Z^k = Z^j \quad (1^\circ)$$

$$Y^k = Y^j \quad (2^\circ)$$

$$Q^k = \{q^i : \bar{q}^i \in T_i\} \cup (Q^j \setminus \bigcup_{\bar{q}^i \in T_i} \bar{q}^i) \quad (3^\circ)$$

$$\Phi^k = \{ \langle q', z_p, q'' \rangle : q_1 \in \bar{q}' \wedge q_n \in \bar{q}'' \wedge q', q'' \in Q^k \wedge \langle q_1, z_p, q_n \rangle \in \Phi^j \} \quad (4^\circ)$$

$$\Psi^k = \{ \langle q', y_s \rangle : q_1 \in \bar{q}' \wedge q' \in Q^k \wedge \langle q_1, y_s \rangle \in \Psi^j \} \quad (5^\circ)$$

Dowód.

Z definicji operacji Δ^{T_i} wynika, że łuki grafu G^k nie mogą być opisa-

ne takimi literami z_c , które nie są elementami zbioru Z^j , a więc:

$$Z^k \subset Z^j \quad (6^0)$$

Każda litera $z_1 \in Z^j$ opisuje co najmniej jeden łuk grafu G^k , a więc:

$$Z^j \subset Z^k \quad (7^0)$$

Z $(6^0), (7^0)$ otrzymujemy wyrażenie (1^0) tezy twierdzenia.

Z definicji operacji Δ^{τ_i} wnioskujemy, że wierzchołki grafu G^k nie mogą być opisane takimi literami y_c , które nie są elementami zbioru Y^j , a więc:

$$Y^k \subset Y^j \quad (8^0)$$

Każda litera $y_1 \in Y^j$ opisuje co najmniej jeden wierzchołek grafu G^k , czyli:

$$Y^j \subset Y^k \quad (9^0)$$

Z $(8^0), (9^0)$ otrzymujemy wyrażenie (2^0) tezy twierdzenia.

Zbiór Θ^j wierzchołków grafu G^j reprezentuje zbiór Q^j stanów automatu A^j , natomiast zbiór Θ^k wierzchołków grafu G^k - zbiór Q^k stanów automatu A^k . Stąd otrzymujemy wyrażenie (3^0) tezy.

Dla automatu A^j i grafu G^j wyrażającego A^j prawdziwe są równości:

$$\Phi^j = \mathcal{Q}^j$$

$$\Psi^j = L^j$$

Analogiczne równości prawdziwe są dla automatu A^k i grafu G^k .

Z powyższego i z definicji operacji Δ^{τ_i} otrzymujemy wyrażenia $(4^0), (5^0)$ tezy twierdzenia. ■

Z wyrażenia (5^0) tezy powyższego twierdzenia wynika następujący wniosek.

Wniosek 2.2.

Automat zagregowany drugiego rodzaju jest automatem w sensie automatu Moore'a.

Za pomocą definicji 2.1. określono deterministyczny automat Moore'a. W celu wykorzystania tej definicji, w prowadzonych w tym

punkcie rozważaniach, należy w miejsce symboli $q_i, q_a, \dots$ podstawić symbole $q', q'', \dots$.

Twierdzenie 2.7.

Automat zagregowany drugiego rodzaju jest automatem deterministycznym.

Dowód.

Dane są graf G^j określony za pomocą definicji 2.2. i graf G^k zadany definicją 2.10.

W celu uproszczenia zapisów dowodu będziemy stosowali oznaczenie określone wyrażeniem (2.66).

Ponieważ automat wyrażony grafem G^j jest automatem deterministycznym, a więc dla jego relacji Φ^j charakteryzującej przejścia między stanami tego automatu prawdziwe jest zdanie:

$$(\forall \langle q_i, z_p, q_j \rangle, \langle q_k, z_s, q_n \rangle \in \Phi^j) ((q_i = q_k \wedge z_p = z_s) \Rightarrow q_j = q_n) \quad (1^0)$$

Symbol $\exists_1$ oznacza kwantyfikator szczegółowy jednostkowy.

Z pierwszego warunku determinizmu dla operacji Δ^{T_i} wynika, że:

$$(\forall q' \in Q^k) (\forall \langle q_i, z_p, q_j \rangle \in \Phi^j) (q_i \in \bar{q}' \Rightarrow (\exists_1 q'' \in Q^k) (q_j \in \bar{q}'')) \quad (2^0)$$

Z (2^0) i drugiego warunku determinizmu dla operacji Δ^{T_i} wnioskujemy, że:

$$(\forall \langle q_j, z_s, q_c \rangle, \langle q_k, z_t, q_r \rangle \in \Phi^j) ((q_j \in \bar{q}^d \wedge q_k \in \bar{q}^d \wedge z_s = z_t) \Rightarrow (\exists_1 \bar{q}^a \in T_i) (q_c, q_r \in \bar{q}^a)) \quad (3^0)$$

Z (1^0) , (2^0) wynika, że dla wierzchołków q_t grafu G^j samodzielnie wyznaczających wierzchołki grafu G^k prawdziwe jest zdanie:

$$(\forall \langle q_t, z_p, q_j \rangle, \langle q_t, z_s, q_r \rangle \in \Phi^j) (z_p = z_s \Rightarrow (\exists_1 q''' \in Q^k) (q_j, q_r \in \bar{q}''')) \quad (4^0)$$

Na podstawie (3^0) , (4^0) i pierwszego warunku determinizmu dla operacji Δ^{T_i} stwierdzamy, że dla relacji Φ^k charakteryzującej przejścia między stanami automatu A^k prawdziwa jest zależność:

$$(\forall \langle q', z_s, q'' \rangle, \langle q^{IV}, z_p, q^V \rangle \in \Phi^k) ((q' = q^{IV} \wedge z_s = z_p) \Rightarrow q'' = q^V) \quad (5^0)$$

Ponieważ automat A^j jest automatem deterministycznym, a więc prawdziwe jest wyrażenie:

$$(\forall \langle q_i, y_p \rangle, \langle q_c, y_s \rangle \in \Psi^j) (q_i = q_c \Rightarrow y_p = y_s) \quad (6^0)$$

Z (6⁰) i trzeciego warunku determinizmu dla operacji Δ^{T_i} wynika, że relacja Ψ^k charakteryzująca reakcje automatu A^k jest funkcją.

Ponieważ relacje Φ^k , Ψ^k są funkcjami, a więc automat A^k jest automatem deterministycznym.

2.3.5. Własności automatów zagregowanych drugiego rodzaju

Twierdzenia przedstawione w tym punkcie dotyczą relacji między zbiorami $B(G)$ charakteryzującymi zachowanie automatów zagregowanych drugiego rodzaju. Relacje te wyznaczone są dla różnych zależności między zbiorami S_j oraz różnych relacji między zbiorami T_i .

Niech symbol:

$$\text{Det} (S_j, T_i, \{G_i\}_{i \in J}) \quad (2.92)$$

oznacza, że zbiory S_j , T_i , $\{G_i\}_{i \in J}$ spełniają warunki determinizmu dla operacji $\oplus^{S_j}$ i Δ^{T_i} .

Niżej przedstawiony zostanie lemat charakteryzujący warunki determinizmu dla operacji $\oplus^S$ i Δ^T . Lemat ten jest podstawą do formułowania założeń twierdzeń 2.8., 2.9., 2.10. .

Lemat 2.3.

$$Z. S_m \subset S_n \wedge \text{Det}(S_n, T, \{G_i\}_{i \in J})$$

$$T. \text{Det}(S_m, T, \{G_i\}_{i \in J})$$

D. Ponieważ $\text{Det}(S_n, T, \{G_i\}_{i \in J})$, a więc m. in. spełnione są warunki determinizmu dla operacji $\oplus^{S_n}$, co oznacza, że prawdziwe jest wyrażenie:

$$\text{Det}(S_n, \{G_i\}_{i \in J}) \quad (1^0)$$

Z zależności $S_m \subset S_n$, (1⁰) i lematu 2.2. z punktu 2.3.3. wynika, że:

$$\text{Det}(S_m, \{G_i\}_{i \in J}) \quad (2^0)$$

czyli, że zbiory $S_m, \{G_i\}_{i \in J}$ spełniają warunki determinizmu dla operacji $\oplus^m$.

Na mocy założenia spełniony jest pierwszy warunek determinizmu dla operacji Δ^T wykonywanej na grafie:

$$G^n = \bigoplus_{i \in J}^{S_n} G_i \quad (3^\circ)$$

Stąd prawdziwe jest zdanie:

$$(\forall \bar{q}^p, \bar{q}^r \in T) (\bar{q}^p \neq \bar{q}^r \Rightarrow \bar{q}^p \cap \bar{q}^r = \emptyset) \quad (4^\circ)$$

Z powyższego wynika, że również spełniony jest pierwszy warunek determinizmu dla operacji Δ^T wykonywanej na grafie:

$$G^m = \bigoplus_{i \in J}^{S_m} G_i \quad (5^\circ)$$

Na podstawie założenia stwierdzamy, że zbiory $S_m, T, \{G_i\}_{i \in J}$ spełniają drugi warunek determinizmu dla operacji Δ . Stąd prawdziwe jest zdanie:

$$(\forall \langle q_j, z_s, q_c \rangle, \langle q_k, z_t, q_r \rangle \in G^n) ((q_j \in \bar{q}^i \wedge q_k \in \bar{q}^i \wedge z_s = z_t) \Rightarrow (\exists \bar{q}^a \in T) (q_c, q_r \in \bar{q}^a)) \quad (6^\circ)$$

Ponieważ $S_m \subset S_n$, a więc na podstawie sposobu określenia zbiorów G^m, G^n za pomocą definicji 2.2. otrzymujemy:

$$G^m \subset G^n \quad (7^\circ)$$

Z (6°), (7°) wnioskujemy, że zbiory $S_m, T, \{G_i\}_{i \in J}$ spełniają drugi warunek determinizmu dla operacji Δ .

Na mocy założenia zbiory $S_n, T, \{G_i\}_{i \in J}$ spełniają trzeci warunek determinizmu dla operacji Δ , a zatem prawdziwe jest zdanie:

$$(\forall \bar{q}^i \in T) (\forall q_j, q_k \in \bar{q}^i) ((q_j \in Q_s \wedge q_k \in Q_t) \Rightarrow \Psi_s(q_j) = \Psi_t(q_k)) \quad (8^\circ)$$

gdzie Q_s, Q_t - zbiory stanów podautomatów A_s, A_t ;

Ψ_s, Ψ_t - funkcje wyjść tych podautomatów.

Dla zbiorów $S_m, T, \{G_i\}_{i \in J}$ również prawdziwe jest wyrażenie (8°) , ponieważ w obu przypadkach rozważania dotyczą tego samego zbioru T . Stąd zbiory $S_m, T, \{G_i\}_{i \in J}$ spełniają trzeci warunek determinizmu dla operacji Δ^T .

Z przeprowadzonych rozważań wynika, że

$$\text{Det} \left(S_m, T, \{G_i\}_{i \in J} \right)$$



Lemat 2.3. i rozważania przedstawione w punkcie 2.3.3. pracy są podstawą do sformułowania twierdzeń 2.8., 2.9., 2.10.. Twierdzenia te dotyczą relacji między zbiorami $B(G)$ - charakteryzującymi zachowanie automatów zagregowanych drugiego rodzaju - dla różnych zależności między zbiorami S_j .

Twierdzenie 2.8.

$$Z. S_m \subset S_n \wedge T \neq \emptyset \wedge \text{Det} \left(S_n, T, \{G_i\}_{i \in J} \right)$$

$$T. B \left(\Delta_{i \in J}^{\oplus S_m} G_i \right) \subset B \left(\Delta_{i \in J}^{\oplus S_n} G_i \right)$$

D. Z założenia i lematu 2.3. wynika, że:

$$\text{Det} \left(S_m, T, \{G_i\}_{i \in J} \right) \quad (1^\circ)$$

Na mocy założenia i (1°) wnioskujemy, że można rozważać automaty zagregowane drugiego rodzaju opisane grafami:

$$G^j = \Delta_{i \in J}^{\oplus S_m} G_i \quad (2^\circ)$$

$$G^k = \Delta_{i \in J}^{\oplus S_n} G_i \quad (3^\circ)$$

Ponieważ prawdziwe jest zdanie:

$$\text{Det} \left(S_n, T, \{G_i\}_{i \in J} \right) \quad (4^\circ)$$

więc prawdziwe jest również wyrażenie:

$$\text{Det} \left(S_n, \{G_i\}_{i \in J} \right) \quad (5^\circ)$$

Z (5°), zależności $S_m \subset S_n$ i lematu 2.2. z punktu 2.3.3. pracy wynika, że:

$$\text{Det}(S_m, \{G_i\}_{i \in J}) \quad (6^\circ)$$

Z (5°), (6°) wnioskujemy, że można analizować automaty deterministyczne opisane grafami:

$$G^p = \bigoplus_{i \in J}^{S_m} G_i \quad (7^\circ)$$

$$G^r = \bigoplus_{i \in J}^{S_n} G_i \quad (8^\circ)$$

Twierdzenie 2.3. z punktu 2.3.3. może być uogólnione do wszystkich takich sytuacji, w których dopuszcza się, że $S_m = \emptyset$ lub $S_n = \emptyset$. Wzmiankowane uogólnienie jest niezbędne w przeprowadzanym dowodzie, ponieważ dla automatów zagregowanych drugiego rodzaju nie zakłada się, że $S_j \neq \emptyset$.

Ponieważ $S_m \subset S_n$, a więc na podstawie wymienionego uogólnienia twierdzenia 2.3. otrzymujemy:

$$B(G^p) \subset B(G^r) \quad (9^\circ)$$

Graf G^j zostaje uzyskany z grafu G^p po wykonaniu takiej operacji Δ^T nakładania wierzchołków grafu G^p , która jest analogiczna do operacji prowadzącej do otrzymania grafu G^k z grafu G^r . Stąd też:

$$E(G^j) \subset E(G^k) \quad (10^\circ)$$

Z (10°) i lematu 2.1. z punktu 2.3.3. pracy wynika, że

$$B\left(\Delta \bigoplus_{i \in J}^{S_m} G_i\right) \subset B\left(\Delta \bigoplus_{i \in J}^{S_n} G_i\right)$$

Twierdzenie 2.9.

$$Z. S = S_m \cup S_n \wedge T \neq \emptyset \wedge \text{Det}(S, T, \{G_i\}_{i \in J})$$

$$t. B\left(\Delta \bigoplus_{i \in J}^{S_m} G_i\right) \cup B\left(\Delta \bigoplus_{i \in J}^{S_n} G_i\right) \subset B\left(\Delta \bigoplus_{i \in J}^S G_i\right).$$

D. Z założenia i lematu 2.3. wynika, że można rozpatrywać automaty zagregowane drugiego rodzaju opisane grafami:

$$G^j = \bigoplus_{i \in J}^{T S_m} G_i \quad (1^0)$$

$$G^k = \bigoplus_{i \in J}^{T S_n} G_i \quad (2^0)$$

$$G^p = \bigoplus_{i \in J}^{T S} G_i \quad (3^0)$$

Z $S_m \subset S$, $S_n \subset S$ i twierdzenia 2.8. wnioskujemy, że

$$B(G^j) \subset B(G^p) \quad (4^0)$$

$$B(G^k) \subset B(G^p) \quad (5^0)$$

Na podstawie (4^0) , (5^0) i określeń grafów G^j , G^k , G^p stwierdzamy, że:

$$B\left(\bigoplus_{i \in J}^{T S_m} G_i\right) \cup B\left(\bigoplus_{i \in J}^{T S_n} G_i\right) \subset B\left(\bigoplus_{i \in J}^{T S} G_i\right) \quad \square$$

Jeśli założenie twierdzenia 2.9. jest spełnione, to ogólnie nie jest prawdziwa zależność:

$$B\left(\bigoplus_{i \in J}^{T S} G_i\right) \subset B\left(\bigoplus_{i \in J}^{T S_m} G_i\right) \cup B\left(\bigoplus_{i \in J}^{T S_n} G_i\right) \quad (2.93)$$

Twierdzenie 2.10.

$$Z. S = S_m \cap S_n \wedge T \neq \emptyset \wedge \text{Det}(S_m, T, \{G_i\}_{i \in J}) \wedge \text{Det}(S_n, T, \{G_i\}_{i \in J})$$

$$T. B\left(\bigoplus_{i \in J}^{T S} G_i\right) \subset B\left(\bigoplus_{i \in J}^{T S_m} G_i\right) \cap B\left(\bigoplus_{i \in J}^{T S_n} G_i\right)$$

D. Ponieważ $S = S_m \cap S_n$, a więc:

$$S \subset S_m \wedge S \subset S_n \quad (1^0)$$

Z (1^0) oraz z lematu 2.3. wynika, że:

$$\text{Det}(S, T, \{G_i\}_{i \in J}) \quad (2^0)$$

Z założenia i z (2^0) wnioskujemy, że można analizować automaty zagregowane drugiego rodzaju opisane grafami:

$$G^j = \bigoplus_{i \in J}^{TS} G_i \quad (3^0)$$

$$G^k = \bigoplus_{i \in J}^{TS_n} G_i \quad (4^0)$$

$$G^r = \bigoplus_{i \in J}^{TS} G_i \quad (5^0)$$

Na mocy (1^0) i twierdzenia 2.8. stwierdzamy, że:

$$B\left(\bigoplus_{i \in J}^{TS} G_i\right) \subset B\left(\bigoplus_{i \in J}^{TS_m} G_i\right) \quad (6^0)$$

$$B\left(\bigoplus_{i \in J}^{TS} G_i\right) \subset B\left(\bigoplus_{i \in J}^{TS_n} G_i\right) \quad (7^0)$$

Z $(6^0), (7^0)$ wnioskujemy, że:

$$B\left(\bigoplus_{i \in J}^{TS} G_i\right) \subset B\left(\bigoplus_{i \in J}^{TS_m} G_i\right) \cap B\left(\bigoplus_{i \in J}^{TS_n} G_i\right) \quad \square$$

Jeśli założenie twierdzenia 2.10. jest spełnione, to ogólnie nie jest prawdziwa zależność:

$$B\left(\bigoplus_{i \in J}^{TS_m} G_i\right) \cap B\left(\bigoplus_{i \in J}^{TS_n} G_i\right) \subset B\left(\bigoplus_{i \in J}^{TS} G_i\right) \quad (2.94)$$

Niech dane będzie zdanie:

$$S = S_m \setminus S_n \wedge T \neq \emptyset \wedge \text{Det}(S_m, T, \{G_i\}_{i \in J}) \wedge \text{Det}(S_n, T, \{G_i\}_{i \in J}) \quad (2.95)$$

W przypadku, gdy prawdziwe jest powyższe zdanie, ogólnie nie jest prawdziwa żadna z dwu następujących zależności:

$$B\left(\begin{smallmatrix} T & S \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i\right) \subset B\left(\begin{smallmatrix} T & S_m \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i\right) \setminus B\left(\begin{smallmatrix} T & S_n \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i\right) \quad (2.96)$$

$$B\left(\begin{smallmatrix} T & S_m \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i\right) \setminus B\left(\begin{smallmatrix} T & S_n \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i\right) \subset B\left(\begin{smallmatrix} T & S \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i\right) \quad (2.97)$$

W dalszej części tego punktu przedstawione są twierdzenia dotyczące zależności między zbiorami $B(G)$ - charakteryzującymi zachowanie automatów zagregowanych drugiego rodzaju - dla różnych relacji między zbiorami T_i .

Dla relacji $T_m \subset T_n$ nie jest prawdziwy lemat analogiczny do lematu 2.3. Stąd założenia twierdzeń ilustrujących zależności między zbiorami $B(G)$ dla różnych relacji między zbiorami T_i są zwykle bardziej rozbudowane niż założenia przedstawionych wyżej trzech twierdzeń.

Twierdzenie 2.11.

$$Z. T_m \subset T_n \wedge T_m \neq \emptyset \wedge \text{Det}(S, T_m, \{G_i\}_{i \in J}) \wedge \text{Det}(S, T_n, \{G_i\}_{i \in J})$$

$$T. B\left(\begin{smallmatrix} T_m & S \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i\right) \subset B\left(\begin{smallmatrix} T_n & S \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i\right)$$

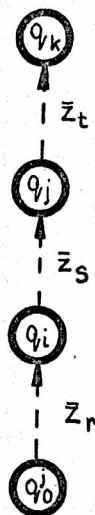
D. Z założenia wynika, że można rozpatrywać automaty zagregowane drugiego rodzaju opisane grafami:

$$G^j = \begin{smallmatrix} T_m & S \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i \quad (1^0)$$

$$G^k = \begin{smallmatrix} T_n & S \\ \Delta \oplus & \\ i \in J & \end{smallmatrix} G_i \quad (2^0)$$

Drogi w grafie G^k są równe drogom grafu G^j lub są drogami będącymi wynikiem nałożenia na siebie pewnych wierzchołków dróg grafu G^j .

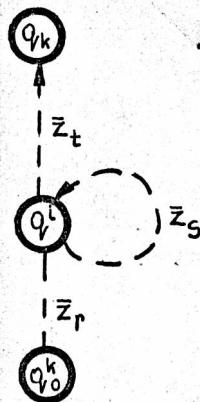
Na przykład niech fragment grafu G^j ma postać przedstawioną na rys. 2.15.

Rys. 2.15. Fragment grafu G^j

Niech zbiór $\bar{q}^i = \{q_i, q_j\}$ spełnia wyrażenie:

$$\bar{q}^i \in T_n \quad (3^0)$$

W wyniku nałożenia na siebie wierzchołków q_i, q_j fragment grafu G^k ma postać przedstawioną na rys. 2.16.

Rys. 2.16. Fragment grafu G^k

Stąd wnioskujemy, że:

$$\langle \bar{z}, \bar{y} \rangle \in B\left(\Delta_{i \in J}^{T_m S} G_i\right) \Rightarrow \langle \bar{z}, \bar{y} \rangle \in B\left(\Delta_{i \in J}^{T_n S} G_i\right) \quad (4^0)$$

a zatem prawdziwe jest zdanie:

$$B\left(\Delta_{i \in J}^{T_m S} G_i\right) \subset B\left(\Delta_{i \in J}^{T_n S} G_i\right)$$



Z powyższego twierdzenia wynika wniosek dotyczący zależności między automatami zagregowanymi pierwszego i drugiego rodzaju.

Wniosek 2.3.

Dla każdego automatu zagregowanego pierwszego rodzaju opisanego grafem $G^c = \bigoplus_{i \in J}^S G_i$ istnieje automat zagregowany drugiego rodzaju o grafie $G^d = \bigoplus_{i \in J}^{TS} G_i (T \neq \emptyset)$, taki że $B(G^c) \subset B(G^d)$.

Dowód tego wniosku jest analogiczny do dowodu twierdzenia 2.11. W założeniu twierdzenia 2.11. zakłada się, że $T_m, T_n \neq \emptyset$, aby to twierdzenie dotyczyło automatów zagregowanych drugiego rodzaju, dla których przyjmuje się, że $T \neq \emptyset$. W dowodzie wniosku jako zbiór T_m traktowany jest zbiór pusty, natomiast jako zbiór T_n - zbiór T .

Twierdzenie 2.12.

$$Z. T = T_m \cup T_n \wedge T_n \neq \emptyset \wedge T_m \neq \emptyset \wedge \text{Det}(S, T_m, \{G_i\}_{i \in J}) \wedge \text{Det}(S, T_n, \{G_i\}_{i \in J}) \wedge \text{Det}(S, T, \{G_i\}_{i \in J})$$

$$T. B\left(\bigoplus_{i \in J}^{T_m S} G_i\right) \cup B\left(\bigoplus_{i \in J}^{T_n S} G_i\right) \subset B\left(\bigoplus_{i \in J}^{TS} G_i\right)$$

D. Na mocy założenia twierdzenia wnioskujemy, że można analizować automaty zagregowane drugiego rodzaju wyrażone grafami G^j, G^k - określonymi tak jak w dowodzie poprzedniego twierdzenia oraz grafem:

$$G^p = \bigoplus_{i \in J}^{TS} G_i \quad (1^0)$$

Ponieważ $T_m \subset T$ i $T_n \subset T$, a więc na mocy twierdzenia 2.11. stwierdzamy, że:

$$B\left(\bigoplus_{i \in J}^{T_m S} G_i\right) \subset B\left(\bigoplus_{i \in J}^{TS} G_i\right) \quad (2^0)$$

$$B\left(\bigoplus_{i \in J}^{T_n S} G_i\right) \subset B\left(\bigoplus_{i \in J}^{TS} G_i\right) \quad (3^0)$$

Z wyrażeń (2^0) , (3^0) wynika, że:

$$B\left(\begin{smallmatrix} T_m S \\ \Delta \oplus \\ i \in J \end{smallmatrix} G_i\right) \cup B\left(\begin{smallmatrix} T_n S \\ \Delta \oplus \\ i \in J \end{smallmatrix} G_i\right) \subset B\left(\begin{smallmatrix} T S \\ \Delta \oplus \\ i \in J \end{smallmatrix} G_i\right)$$

Ogólnie nie jest prawdziwa zależność:

$$B\left(\begin{smallmatrix} T S \\ \Delta \oplus \\ i \in J \end{smallmatrix} G_i\right) \subset B\left(\begin{smallmatrix} T_m S \\ \Delta \oplus \\ i \in J \end{smallmatrix} G_i\right) \cup B\left(\begin{smallmatrix} T_n S \\ \Delta \oplus \\ i \in J \end{smallmatrix} G_i\right) \quad (2.98)$$

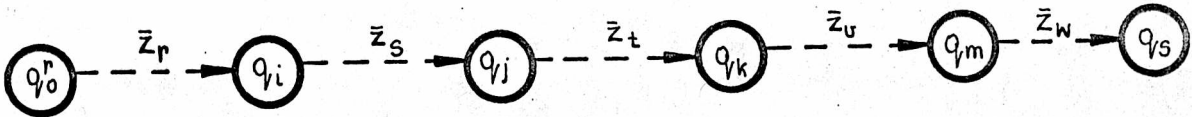
Powyższe stwierdzenie zostanie zilustrowane przykładem.

Przykład 2.6.

Niech fragment grafu:

$$G^r = \bigoplus_{i \in J}^S G_i \quad (2.99)$$

ma postać przedstawioną na rys. 2.17.



Rys. 2.17. Fragment grafu G^r

Niech z wierzchołka q_i nie prowadzi droga, której łuki są opisane literami sekwencji $\bar{z}_t$ oraz niech z wierzchołka q_k nie prowadzi droga o łukach opisanych literami sekwencji $\bar{z}_w$.

Przyjmijmy, że dla zbiorów $\bar{q}^i = \{q_i, q_j\}$, $\bar{q}^k = \{q_k, q_m\}$ prawdziwe są zdania:

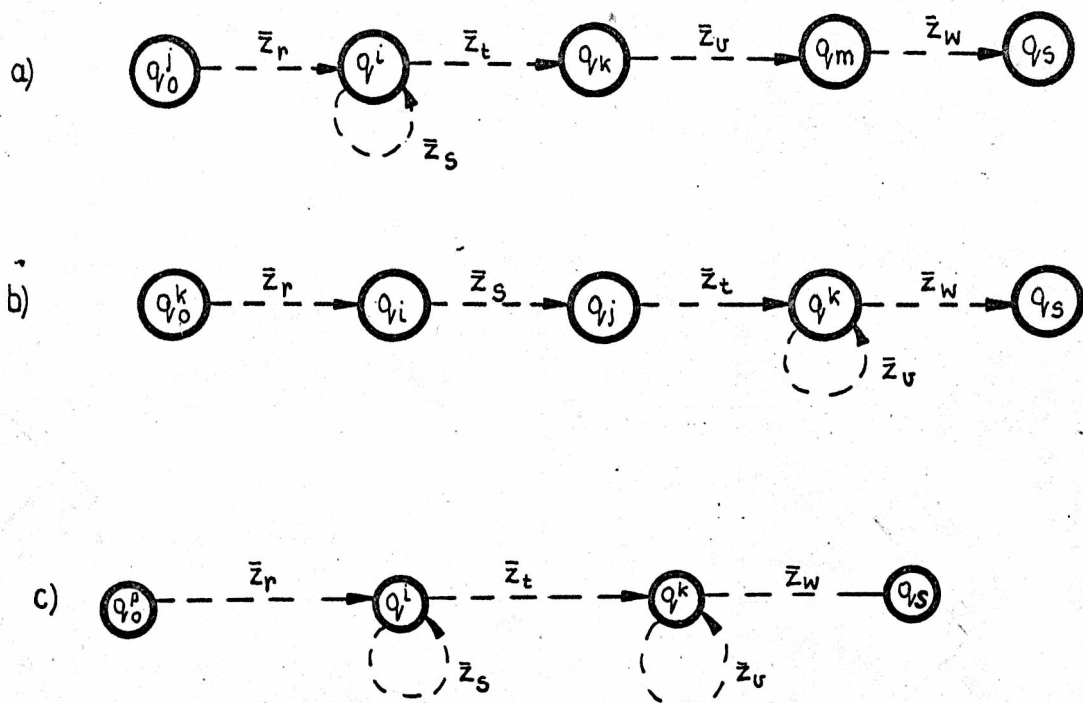
$$\bar{q}^i \in T_m \wedge \bar{q}^k \notin T_m$$

$$\bar{q}^i \notin T_n \wedge \bar{q}^k \in T_n$$

a zatem

$$\bar{q}^i \in T \wedge \bar{q}^k \in T$$

Stąd w grafach G^j , G^k , G^p określonych jak w dowodzie twierdzenia 2.12. znajdują się fragmenty przedstawione na rys. 2.18.



Rys. 2.18. Fragmenty: a) grafu G^j , b) grafu G^k ,
c) grafu G^p

Zbiór $B(G^p)$ zawiera parę uporządkowaną $\langle \bar{z}, \bar{y} \rangle$, której pierwsza składowa ma następującą postać:

$$\bar{z} = \bar{z}_r \bar{z}_t \bar{z}_w \quad (2.100)$$

Zbiór $B(G^j)$ nie zawiera elementu, którego pierwsza składowa ma postać przedstawioną za pomocą wyrażenia (2.100). Podobnie w zbiorze $B(G^k)$ nie ma pary uporządkowanej, której pierwsza składowa jest sekwencją o wzmiankowanej postaci. Stąd zbiór:

$$B(G^j) \cup B(G^k) \quad (2.101)$$

nie zawiera elementu należącego do zbioru $B(G^p)$.

Twierdzenie 2.13.

$$Z. T = T_m \wedge T_n \wedge T \neq \emptyset \wedge \text{Det}(S, T_m, \{G_i\}_{i \in J}) \wedge \text{Det}(S, T_n, \{G_i\}_{i \in J}) \wedge \text{Det}(S, T, \{G_i\}_{i \in J})$$

$$T. B\left(\Delta_{i \in J}^{\top S} G_i\right) \subset B\left(\Delta_{i \in J}^{T_m S} G_i\right) \cap B\left(\Delta_{i \in J}^{T_n S} G_i\right)$$

D. Z założenia wnioskujemy, że można analizować automaty zagregowane drugiego rodzaju opisane grafami:

$$G^J = \Delta_{i \in J}^{T_m S} G_i \quad (1^\circ)$$

$$G^K = \Delta_{i \in J}^{T_n S} G_i \quad (2^\circ)$$

$$G^S = \Delta_{i \in J}^{\top S} G_i \quad (3^\circ)$$

Ponieważ $T \subset T_m$, $T \subset T_n$, a więc z twierdzenia 2.11. wynika, że:

$$B\left(\Delta_{i \in J}^{\top S} G_i\right) \subset B\left(\Delta_{i \in J}^{T_m S} G_i\right) \quad (4^\circ)$$

$$B\left(\Delta_{i \in J}^{\top S} G_i\right) \subset B\left(\Delta_{i \in J}^{T_n S} G_i\right) \quad (5^\circ)$$

Z wyrażeń (4°) , (5°) wnioskujemy, że:

$$B\left(\Delta_{i \in J}^{\top S} G_i\right) \subset B\left(\Delta_{i \in J}^{T_m S} G_i\right) \cap B\left(\Delta_{i \in J}^{T_n S} G_i\right) \quad \blacksquare$$

Ogólnie nie jest prawdziwa zależność:

$$B\left(\Delta_{i \in J}^{T_m S} G_i\right) \cap B\left(\Delta_{i \in J}^{T_n S} G_i\right) \subset B\left(\Delta_{i \in J}^{\top S} G_i\right) \quad (2.102)$$

Jeśli prawdziwe jest zdanie:

$$T = T_m \setminus T_n \wedge T \neq \emptyset \wedge T_n \neq \emptyset \wedge \text{Det}(S, T_m, \{G_i\}_{i \in J}) \wedge \text{Det}(S, T_n, \{G_i\}_{i \in J}) \wedge \text{Det}(S, T, \{G_i\}_{i \in J}) \quad (2.103)$$

to ogólnie nie jest prawdziwa żadna z dwu następujących zależności:

$$B\left(\Delta_{i \in J}^{\oplus} G_i\right) \subset B\left(\Delta_{i \in J}^{T_m S} G_i\right) \setminus B\left(\Delta_{i \in J}^{T_h S} G_i\right) \quad (2.104)$$

$$B\left(\Delta_{i \in J}^{T_m S} G_i\right) \setminus B\left(\Delta_{i \in J}^{T_h S} G_i\right) \subset B\left(\Delta_{i \in J}^{\oplus} G_i\right) \quad (2.105)$$

Niżej zostanie przedstawiony sposób budowy takiego automatu AZI o grafie G^P , który dla zadanego automatu AZII wyrażonego grafem G^J spełnia zależność:

$$B(G^J) \subset B(G^P) \quad (2.106)$$

Graf G^J jest grafem automatu A^J , natomiast graf G^P - automatu A^P . Stąd celem prowadzonych dalej rozważań jest budowa takiego automatu A^P , którego reakcje na sekwencje wejściowe $\bar{z} \in Z(G^J)$, gdzie zbiór $Z(G^J)$ określony jest tak jak w punkcie 2.3.3., są identyczne z reakcjami automatu A^J na te sekwencje.

Niech dany będzie automat AZII o grafie G^J określonym następująco:

$$G^J = \Delta_{i \in J}^{T_j S_j} G_i \quad (2.107)$$

oraz niech:

$$G^k = \Delta_{i \in J}^{S_j} G_i \quad (2.108)$$

Automat A^J ma wyróżniony stan będący stanem początkowym. Stąd wierzchołek grafu G^J przyporządkowany temu stanowi jest nazywany wierzchołkiem początkowym. W celu uniknięcia mało informatywnych szczegółów zostanie wprowadzone założenie dotyczące grafu G^J .

Założenie 2.1.

Wierzchołek początkowy grafu G^J (wyrażenie 2.107) automatu zagregowanego drugiego rodzaju A^J wyznaczony jest przez jeden wierzchołek grafu G^k (wyrażenie 2.108).

W celu uzyskania grafu G^p operacje $\bigoplus_{S_j, T_j}$ wykonywane na zbiorze $\{G_i\}_{i \in J}$ grafów podautomatów należy zastąpić operacją $\bigoplus_{S^p}$, dla której prawdziwa jest zależność:

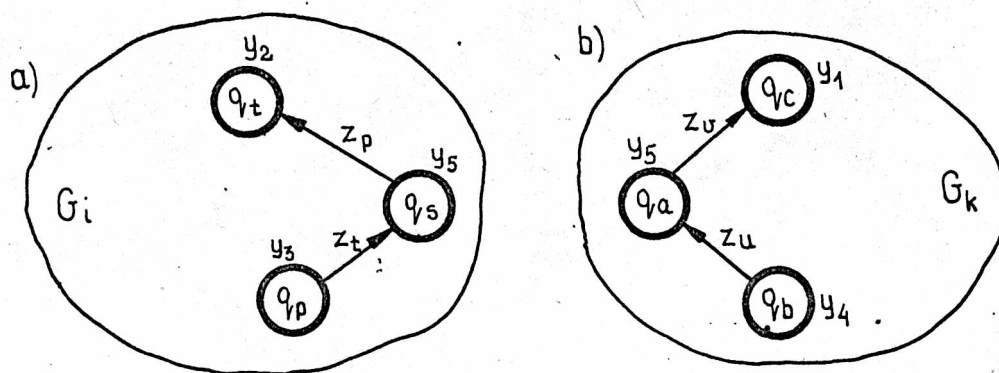
$$G^p = \bigoplus_{i \in J}^{S^p} G_i \quad (2.109)$$

Sposób zastąpienia operacji $\bigoplus_{S_j, T_j}$ za pomocą operacji $\bigoplus_{S^p}$, w pierwszej kolejności będzie zilustrowany prostym przykładem, a następnie - przedstawiony ogólnie.

Przykład 2.7.

Niech dane będą grafy G_i, G_k o fragmentach przedstawionych na rys. 2.19. W wyniku wykonania na tych grafach operacji $\bigoplus_{S_r, T_r}$ otrzymujemy graf:

$$G^r = \bigoplus_{j \in \{i, k\}}^{S_r, T_r} G_j \quad (2.110)$$



Rys. 2.19. Fragmenty: a) grafu G_i , b) grafu G_k

Z rys. 2.19. wynika m. in., że po podaniu litery z_t na wejście podautomatu A_i , wyrażonego grafem G_i , znajdującego się w stanie q_p nastąpi zmiana stanu tego podautomatu na stan q_s oraz na jego wyjściu pojawi się litera y_5 .

Niech q_i będzie wierzchołkiem grafu G_j podautomatu tzn. $q_i \in Q_j$. Symbol $Z_{q_i}^j$ jest symbolem zbioru tych liter $z_c \in Z_j$, którymi opisane

są łuki grafu G_j wychodzące z wierzchołka q_i . Dla grafu G_i przedstawionego na rys. 2.19. niech $z_v \notin Z_{q_s}^i$, natomiast dla grafu G_k niech $z_p \notin Z_{q_a}^k$.

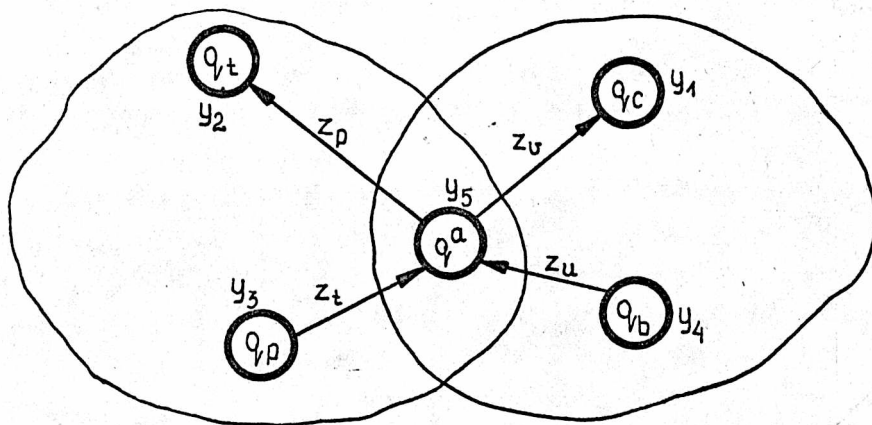
Niech elementem rodziny T_r będzie zbiór:

$$\bar{q}^a = \{q_a, q_s\} \quad (2.111)$$

Stąd w wyniku wykonania operacji Δ^{T_r} na grafie:

$$G^n = \bigoplus_{j \in \{i, k\}}^{S^r} G_j \quad (2.112)$$

zostaną nałożone na siebie wierzchołki q_s grafu G_i oraz wierzchołki q_a grafu G^k . Wierzchołek grafu G^r uzyskany w wyniku nałożenia na siebie elementów zbioru $\bar{q}^a$ oznaczony jest symbolem q^a . Fragment grafu G^r zadanego wyrażeniem (2.110) przedstawiony jest na rys. 2.20.



Rys. 2.20. Fragment grafu G^r automatu zagregowanego drugiego rodzaju A^r

Celem prowadzonych rozważań jest rezygnacja z nakładania na siebie wierzchołków grafów G_i, G_k . Stąd na tych grafach należy wykonać taką operację $\bigoplus^{S_s}$, której wynikiem będzie graf:

$$G^s = \bigoplus_{j \in \{i, k\}}^{S_s} G_j \quad (2.113)$$

automatu zagregowanego pierwszego rodzaju spełniającego zależność:

$$B(G^r) \subset B(G^s) \quad (2.114)$$

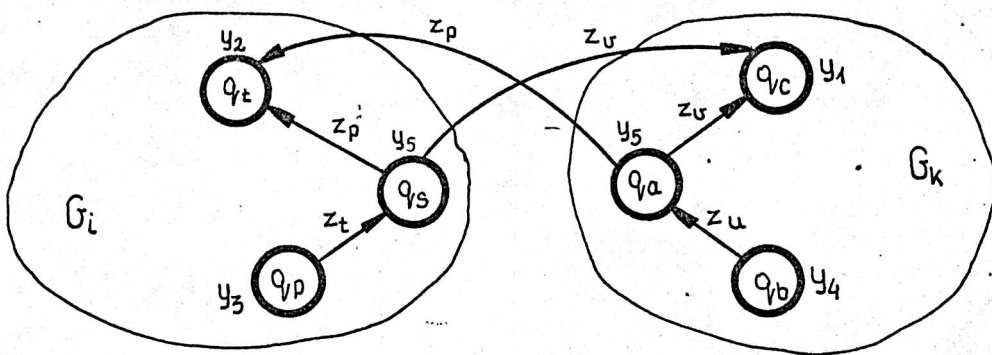
Niżej zostanie przedstawiona część operacji $\oplus_{S_s}$ przeprowadzanej na grafach G_i , G_k podautomatów.

W stanie q^a automatu A^r , wyrażonego grafem G^r , na wejściu tego automatu może pojawić się jedna z liter ze zbioru:

$$Z_{q^a} = Z_{q_a}^k \cup Z_{q_s}^i \quad (2.115)$$

W stanie q_a podautomatu A_k , na wejściu tego podautomatu może pojawić się dowolny element zbioru $Z_{q_a}^k$, natomiast w stanie q_s podautomatu A_i - $Z_{q_s}^i$. Z założenia, że $z_v \notin Z_{q_s}^i$ i z rys. 2.19. wynika, że w stanie q_s podautomatu A_i na wejściu tego podautomatu może pojawić się litera z_p , natomiast nie może pojawić się litera z_v . Podobnie w stanie q_a podautomatu A_k może pojawić się na jego wejściu litera z_v , a nie może - litera z_p .

W wyniku wykonania operacji wyrażonej za pomocą napisu (2.113) w grafie G^s będą m. in. dwa łuki przedstawione na rys. 2.21.



Rys. 2.21. Fragment grafu G^s automatu zagregowanego pierwszego rodzaju A^s

Z rys. 2.21. wynika, że w stanach q_s , q_a automatu A^s mogą pojawić się litery z_v , $z_p \in Z_{q_a}$. Z rys. 2.20. wnioskujemy, że po podaniu litery z_v na wejście automatu A^r znajdującego się w stanie q^a na

wyjściu tego automatu zostanie wygenerowana litera $y_1 \in Y^r$. Natomiast z rys. 2.21. otrzymujemy, że po podaniu litery z_v na wejście automatu A^S znajdującego się w stanie q_s lub q_a , na wyjściu tego automatu pojawi się litera y_1 . Podobne rozważania można przeprowadzić dla litery z_p .

Zbiór S_S spełnia zależność:

$$S_r \subset S_S$$

gdzie S_r jest zbiorem charakteryzującym operację ilustrowaną wyrażeniem (2.112). Z rys. 2.21. wnioskujemy, że elementami zbioru S_S są m. in. następujące trójki uporządkowane:

$$\langle q_a, z_p, q_t \rangle, \langle q_s, z_v, q_c \rangle$$

Niżej zostaną przedstawione ogólne zasady zastąpienia operacji $T_{\Delta}^j, S_j^{\oplus}$ - odpowiednią operacją $S_p^{\oplus}$.

Niech dane będą grafy G^j, G^k, G^p opisane wyrażeniami (2.107), (2.108), (2.109). Celem prowadzonych rozważań jest synteza zbioru S_p , którego symbol występuje w wyrażeniu (2.109).

Operacja $S_p^{\oplus}$ musi zastąpić operację $S_j^{\oplus}$. Stąd elementami zbioru S_p są elementy zbioru S_j , to znaczy:

$$S_j \subset S_p$$

Trójki uporządkowane:

$$\langle q_1, z_s, q_j \rangle \in S_p \setminus S_j \quad (2.116)$$

umożliwiają zastąpienie operacji T_{Δ}^j za pomocą operacji $S_p^{\oplus}$.

Graf G^j (wyrażenie 2.107) można określić jako $G^j = \langle Q^j, G^j, L^j \rangle$ gdzie znaczenie poszczególnych współrzędnych zadane jest definicją 2.10.

Elementami zbioru Q^j są wierzchołki wyznaczone przez jeden wierzcho-

łuk q_c grafu G^k (wyrażenie 2.108) oraz wierzchołki wyznaczone przez zbiory $\bar{q}^a \in T_j$. W celu skrócenia zapisów stosowane będą określenia: wierzchołek typu q_c dla wierzchołka grafu G^j wyznaczonego przez jeden wierzchołek grafu G^k , wierzchołek typu q^a dla wierzchołka grafu G^j wyznaczonego przez zbiór rodziny T_j . W podobnym celu będzie stosowany symbol q lub q' dla oznaczenia wierzchołka grafu G^j , w przypadku gdy ten wierzchołek może być wierzchołkiem każdego z dwu wyżej wymienionych typów.

Niech symbol Z_{q_c} , gdzie q_c jest wierzchołkiem grafu G^k , oznacza zbiór tych liter $z_p \in Z^k$, którymi opisane są łuki G^k wychodzące z wierzchołka q_c . Symbol Z_{q^a} , gdzie q^a jest wierzchołkiem grafu G^j wyznaczonym przez zbiór $\bar{q}^a \in T_j$, niech oznacza zbiór tych liter $z_s \in Z^j$ którymi opisane są łuki grafu G^j wychodzące z wierzchołka q^a .

Lemat 2.4.

$$Z_{q^a} = \bigcup_{q_c \in \bar{q}^a} Z_{q_c}$$

Dowód wynika ze sposobu nakładania na siebie wierzchołków $q_c \in \bar{q}^a$. ■

Dla wierzchołka $q \in Q^j$ można wyznaczyć zbiór zwany zbiorem jego następców, a oznaczony symbolem $N(q)$. Zbiór $N(q)$ jest zbiorem tych wierzchołków grafu G^j , do których skierowane są łuki tego grafu z wierzchołka q .

Definicja 2.12.

$$N(q) = \{q' : (\exists z_s \in Z^j)(\langle q, z_s, q' \rangle \in \Theta^j)\} \quad (2.117)$$

gdzie Z^j - alfabet wejściowy automatu A^j .

Dla wierzchołka $q^a \in Q^j$ wyznaczonego przez zbiór $\bar{q}^a \in T_j$, należy określić zbiór $N(q^a)$ jego następców. Następnie dla wierzchołka $q_c \in \bar{q}^a$ należy wyznaczyć zbiór tych następców q^a , do których nie prowadzą łuki opisane literami $z_s \in Z_{q_c}$. Wzmiankowany zbiór oznaczony

jest symbolem $\bar{N}(q_c)$.

Definicja 2.13.

$$\bar{N}(q_c) = \{q: q \in N(q^a) \wedge q_c \in \bar{q}^a \wedge \bar{q}^a \in T_j \wedge (\exists z_s \in Z_{q^a} \setminus Z_{q_c}) (\langle q^a, z_s, q \rangle \in \mathcal{C}^j)\} \quad (2.118)$$

W przykładzie 2.7. $q_t, q_c \in N(q^a)$, gdzie q_t, q_c, q^a są wierzchołkami grafu G^T przedstawionego na rys. 2.20. Ponieważ w grafie G_k , ilustrowanym za pomocą rys. 2.19., nie ma łuku skierowanego z wierzchołka q_a i opisanego literą z_p , a więc $q_t \in \bar{N}(q_a)$.

W prowadzonych rozważaniach ogólnych, niech $q \in \bar{N}(q_c)$ oraz niech łuk skierowany z wierzchołka $q^a \in \mathcal{Q}^j$ do wierzchołka $q \in \mathcal{Q}^j$ będzie opisany literą $z_i \in Z_{q^a} \setminus Z_{q_c}$. W celu uzyskania grafu G^P (wyrażenie 2.109) z grafu G^k , należy do G^k dobudować łuk opisany literą z_i skierowany z wierzchołka q_c do jednego z wierzchołków wyznaczających wierzchołek q . Niech wybranym wierzchołkiem będzie q_n . Stąd otrzymujemy, że:

$$\langle q_c, z_i, q_n \rangle \in S_p$$

W szczególnym przypadku, jeśli wierzchołek q jest wyznaczony przez jeden wierzchołek q_s , to łuk dobudowany do grafu G^k , opisany literą z_i ma być skierowany z wierzchołka q_c do q_s . Podobną analizę należy przeprowadzić dla innych liter $z_j \in Z_{q^a} \setminus Z_{q_c}$. Aby było możliwe dobudowanie do grafu G^k określonych w przedstawiony wyżej sposób łuków wychodzących z wierzchołka q_c , zbiór S_p musi zawierać trójki $\langle q_c, z_i, q_n \rangle$ wyznaczające te łuki. Zbiór tych trójek dla wierzchołka $q_c \in \bar{q}^a$ zostaje oznaczony symbolem $S(q_c | \bar{q}^a)$.

W podobny sposób należy dobudować do grafu G^k łuki wychodzące z innych wierzchołków $q_b \in \bar{q}^a$. Stąd można wyznaczyć zbiór:

$$S(\bar{q}^a) = \bigcup_{q_b \in \bar{q}^a} S(q_b | \bar{q}^a) \quad (2.119)$$

Rozważania podobne do rozważań dotyczących zbioru $\bar{q}^a$ należy przeprowadzić dla innych zbiorów $\bar{q}^m \in T_j$. Stąd otrzymamy zbiór:

$$S_j^+ = \bigcup_{\bar{q}^m \in T_j} s(\bar{q}^m) \quad (2.120)$$

Zbiór S_p określony jest następująco:

$$S_p = S_j \cup S_j^+ \quad (2.121)$$

Kolejność dobudowywania do grafów ze zbioru $\{G_i\}_{i \in J}$ łuków wyznaczonych przez elementy zbioru S_p nie jest dowolna. W pierwszej kolejności dobudowywane są łuki określone przez elementy zbioru S_j , a następnie - przez elementy zbioru S_j^+ . Stąd łuki wyznaczone przez elementy zbioru S_j^+ dobudowywane są do grafu G^k przedstawionego za pomocą wyrażenia (2.108).

Wierzchołek początkowy grafu G^p wyznaczony jest przez ten wierzchołek grafu G^k , przez który wyznaczony jest wierzchołek początkowy grafu G^j . Wierzchołek początkowy grafu G^p może być określony w podany wyżej sposób dzięki założeniu 2.1.

Rezygnacja z założenia 2.1. spowodowałaby konieczność uwzględnienia szczegółów zmniejszających stopień czytelności prowadzonych rozważań.

W końcowej części tego punktu zostanie udowodnione twierdzenie, z którego wynika, że wyrażenie (2.106) jest prawdziwe. W dowodzie tego twierdzenia zastosowany jest przedstawiony niżej lemat.

Dla elementu q_b zbioru $\bar{q}^a \in T_j$ można stosować zapis $q_b \in \bar{q}^a$. W celu uproszczenia dowodu lematu dla wierzchołka typu q_c grafu G^j stosowane będzie oznaczenie określone wyrażeniem (2.66) czyli $q_c \in \bar{q}_c$.

W dowodzie tego lematu, w tych przypadkach, w których typ wierzchołka grafu G^j jest nieistotny, dla określenia wierzchołków G^j stosowane będą symbole $q, q', q'', \dots$.

Lemat 2.5.

Z. Dany jest automat zagregowany drugiego rodzaju A^j . Wierzchołek początkowy grafu G^j (wyrażenie 2.107) tego automatu wyznaczony jest przez jeden wierzchołek grafu G^k (wyrażenie 2.108).

Niech po podaniu sekwencji $\bar{z} \in Z(G^j)$ na wejście automatu A^j , znajdującego się w stanie początkowym q_v , automat ten osiągnie stan q wyznaczony przez zbiór $\bar{q}$.

T. Po podaniu sekwencji $\bar{z}$ na wejście automatu zagregowanego pierwszego rodzaju A^p , zadanego grafem G_p (wyrażenie 2.109), znajdującego się w stanie początkowym q_v , automat ten osiągnie stan $q_m \in \bar{q}$.

Dowód.

Niech długość sekwencji $\bar{z}$ wynosi $l(\bar{z}) = n$ oraz niech symbol $w_k(\bar{z})$ oznacza k pierwszych liter tej sekwencji. Niech po podaniu litery $w_1(\bar{z})$ na wejście automatu A^j znajdującego się w stanie początkowym q_v , automat ten osiągnie stan q' .

Ze sposobu konstrukcji grafu G^p wynika, że z wierzchołka q_v tego grafu skierowany jest łuk opisany literą $w_1(\bar{z})$. Niech ten łuk będzie skierowany do wierzchołka q_w . Stąd automat A^p znajdujący się w stanie początkowym q_v , po podaniu na jego wejście litery $w_1(\bar{z})$ osiągnie stan q_w .

W celu udowodnienia, że $q_w \in \bar{q}'$ przeprowadzony zostanie dowód nie wprost.

Założenie L/1.

Niech $q_w \notin \bar{q}'$.

Z założenia L/1 wynika, że w grafie G^j jest łuk opisany literą $w_1(z)$, skierowany z wierzchołka q_v do takiego wierzchołka q'' , dla którego prawdziwa jest zależność $q_w \in \bar{q}''$. W grafie G^j jest również łuk opisany literą $w_1(z)$, skierowany z wierzchołka q_v do $q' \neq q''$. Stąd w tym grafie są dwa łuki opisane literą $w_1(z)$, skierowane z wierzchołka q_v do dwóch wierzchołków. Wobec tego automat A^j jest automatem

niedeterministycznym. Ponieważ automaty zagregowane drugiego rodzaju są automatami deterministycznymi, a więc $q_w \in \bar{q}'$.

Niech automat A^j , znajdujący się w stanie początkowym q_v , po podaniu na jego wejście sekwencji $w_k(\bar{z})$, gdzie $1 \leq k < n$, osiągnie stan q''' .

Dla potrzeb dowodu zostaje przyjęte następujące założenie.

Założenie L/2.

Niech po podaniu sekwencji $w_k(\bar{z})$, gdzie $1 \leq k < n$, na wejście automatu A^p , znajdującego się w stanie początkowym q_v , automat ten osiągnie stan $q_p \in \bar{q}'''$.

Niech $(k+1)$ -szą literą sekwencji $\bar{z}$ będzie z_i , a więc $w_{k+1}(\bar{z}) = w_k(\bar{z})z_i$. Niech automat A^j , znajdujący się w stanie początkowym, po podaniu na jego wejście sekwencji $w_{k+1}(\bar{z})$ osiągnie stan q^{IV} .

Ponieważ $z_i \in Z_q'''$, a więc ze sposobu konstrukcji zbioru S_p i założenia L/2 wnioskujemy, że z wierzchołka q_p grafu G^p skierowany jest łuk opisany literą z_i . Niech ten łuk będzie skierowany do wierzchołka q_r . Stąd automat A^p , znajdujący się w stanie początkowym, po podaniu na jego wejście sekwencji $w_{k+1}(\bar{z})$ osiągnie stan q_r .

W celu udowodnienia, że $q_r \in \bar{q}^{IV}$ przeprowadzony zostanie dowód nie wprost.

Założenie L/3.

Niech $q_r \notin \bar{q}^{IV}$.

Z założenia L/3 wynika, że w grafie G^j jest łuk opisany literą z_i , skierowany z wierzchołka q''' do wierzchołka q^V - spełniającego zależność $q_r \in \bar{q}^V$. W grafie G^j jest również łuk opisany literą z_i , skierowany z wierzchołka q''' do wierzchołka q^{IV} . Stąd w tym grafie są dwa łuki opisane literą z_i , skierowane z wierzchołka q''' do dwóch wierzchołków. Z rozumowania podobnego do rozumowania przeprowadzonego dla wierzchołka q_w wynika, że $q_r \in \bar{q}^{IV}$.

Zatem udowodniono, że jeśli automat A^P znajdujący się w stanie początkowym, po podaniu na jego wejście sekwencji $w_k(\bar{z})$ osiągnie stan $q_p \in \bar{q}'''$, to automat ten znajdujący się w stanie początkowym, po podaniu na jego wejście sekwencji $w_{k+1}(\bar{z})$ osiągnie stan $q_r \in \bar{q}^{IV}$. Poprzednio udowodniono, że $q_w \in \bar{q}'$. Stosując zasadę indukcji, z dwóch ostatnich zdań i założenia otrzymujemy tezę. ■

Twierdzenie 2.14.

Z. Dany jest automat zagregowany drugiego rodzaju A^J wyrażony grafem:

$$G_J = \begin{matrix} T_j & S_j \\ \Delta & \oplus_j \\ i \in J \end{matrix} G_i$$

Wierzchołek początkowy grafu G^J tego automatu wyznaczony jest przez jeden wierzchołek grafu:

$$G^k = \begin{matrix} S_j \\ \oplus_j \\ i \in J \end{matrix} G_i$$

T. Dla automatu A^J istnieje taki automat zagregowany pierwszego rodzaju A^P o grafie:

$$G^P = \begin{matrix} S \\ \oplus^P \\ i \in J \end{matrix} G_i$$

dla którego prawdziwa jest zależność:

$$B(G^J) \subset B(G^P)$$

Dowód.

Niech $\langle \bar{z}, \bar{y} \rangle \in B(G^J)$, gdzie $l(\bar{z}) = l(\bar{y}) = n$.

Wierzchołek $q_c \in \bar{q}$ grafu G^P , na mocy trzeciego warunku determinizmu dla operacji $\begin{matrix} T_j \\ \Delta \end{matrix}$, opisany jest tą literą alfabetu wyjściowego Y^P , którą opisany jest wierzchołek q grafu G^J . Stąd na mocy lematu 2.5., reakcja automatu A^P , znajdującego się w stanie początkowym, na k -tą literę sekwencji $\bar{z} \in Z(G^J)$, gdzie $1 \leq k \leq n$, jest taka sama jak reakcja automatu A^J , znajdującego się w stanie początkowym, na k -tą li-

terę tej sekwencji. Z treści powyższego zdania wynika, że $\langle \bar{z}, \bar{y} \rangle \in B(G^P)$
 Ponieważ:

$$(\forall \bar{z} \in Z(G^J))(\forall \bar{y} \in Y^{J*})(\langle \bar{z}, \bar{y} \rangle \in B(G^J) \Rightarrow \langle \bar{z}, \bar{y} \rangle \in B(G^P))$$

a więc otrzymujemy tezę. ■

2.3.6. Proces syntezy formalnej automatu zagregowanego zdolnego do realizacji zadania zgodnie z nowym wariantem jego rozwiązania

Dany jest graf wariantu rozwiązania zadania (WRZ). Należy sprawdzić, czy na bazie zbioru L podautomatów odwzorowywanych działaniem automatu AP można zbudować automat zagregowany o grafie izomorficznym z grafem WRZ zdolny do realizacji zadania zgodnie z wymienionym wariantem jego rozwiązania.

Rozważania prowadzone w tym punkcie będą dotyczyły automatów zagregowanych pierwszego rodzaju.

Dla automatów zagregowanych drugiego rodzaju proces syntezy odpowiedniego automatu jest podobny.

Jeśli okaże się, że na bazie zbioru L można zbudować automat AZI o grafie izomorficznym z grafem WRZ, to uczący się automat parametryczny będzie zdolny do działania zgodnie z rozważanym wariantem.

Automat UAP może odwzorowywać bardzo wiele automatów zagregowanych pierwszego rodzaju. Podstawą do budowy tych automatów są podautomaty odwzorowywane działaniem AP . Przyjmuje się, że łuki łączące wierzchołki grafów podautomatów otrzymane w wyniku wykonania operacji $\oplus$ mogą być oznaczone dowolnymi literami. Stąd zbiór Z' liter, którymi opisane są wzmiankowane łuki spełnia zależność:

$$\bigcup_{i \in I} Z_i \subset Z' \quad (2.122)$$

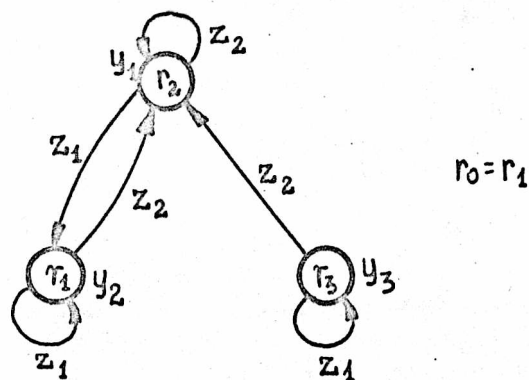
gdzie Z_i - alfabet wejściowy podautomatu A_i .

Przyjęte zostają następujące założenia dotyczące WRZ.

Założenie 2.2.

Graf wariantu rozwiązania zadania jest grafem, w którym do każdego wierzchołka prowadzi droga z wierzchołka początkowego.

Przykład grafu nie spełniającego tego warunku jest przedstawiony na rys. 2.22.



Rys. 2.22. Graf WRZ, w którym nie istnieje droga prowadząca z wierzchołka początkowego r_1 do wierzchołka r_3

Litera y_3 jest nieistotna dla realizacji zadania zgodnie z WRZ o przedstawionym wyżej grafie. Ogólnie przyjęte założenie nie wprowadza istotnych ograniczeń z punktu widzenia sposobu realizacji zadania.

Założenie 2.3.

W grafie wariantu rozwiązania zadania graf wyrażający dane podzadanie występuje nie więcej, niż jeden raz.

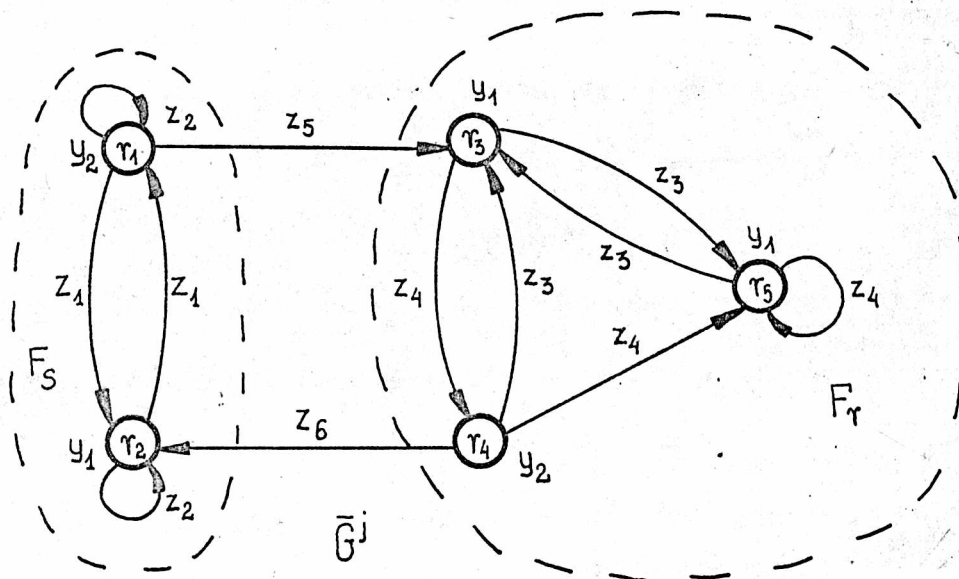
Mimo, że graf opisujący zadane podzadanie występuje tylko jeden raz w grafie WRZ, to jednak wzmiankowane podzadanie dzięki odpowiedniej budowie grafu WRZ, może być wielokrotnie wykonywane w procesie realizacji zadania zgodnie z tym WRZ. Rezygnacja z tego założenia powodowałaby konieczność wprowadzenia wielu mało istotnych szczegółów.

Proces uczenia się polega na budowie takiego automatu AZI, którego graf jest izomorficzny z grafem WRZ. W wyniku analizy WRZ należy zatem dokonać wyboru ze zbioru L odpowiednich podautomatów oraz uzyskanym podzbiorem zbioru L wykonać odpowiednią operację $\oplus$, w celu otrzymania wzmiankowanego wyżej AZI.

Niech zamiast grafu WRZ dany będzie graf pewnego automatu. W wyniku analizy grafu tego automatu podobnej do analizy grafu WRZ można zrealizować dekompozycję tego automatu na zbiór współpracujących ze sobą podautomatów ze zbioru L .

Graf WRZ oznaczony jest symbolem $\bar{G}^i$. Graf ten ma wierzchołek początkowy r_0^i . Podautomaty będące elementami zbioru L wyrażone są grafami ze zbioru G^+ zadanego wyrażeniem (2.5). Analiza grafu $\bar{G}^i$ WRZ polega na rozbiciu tego grafu na podgrafy opisujące podzadania, do których realizacji zdolne są podautomaty ze zbioru L . Jeśli WRZ zawiera co najmniej jedno podzadanie, którego graf nie jest izomorficzny z żadnym z elementów zbioru G^+ , to rozważany WRZ nie zostanie opanowany przez automat UAP.

Niech graf $\bar{G}^j$ WRZ ma postać przedstawioną na rys. 2.23.



Rys. 2.23. Graf $\bar{G}^j$ WRZ

Niech podgrafy F_S , F_r grafu $\bar{G}^j$ będą izomorficzne z grafami pewnych podautomatów ze zbioru L . W procesie analizy grafu $\bar{G}^j$ należy w tym grafie wyodrębnić podgrafy F_S , F_r oraz określić, między którymi wierzchołkami tych podgrafów są łuki łączące te podgrafy i jakimi literami alfabetu Z te łuki są opisane.

Problem syntezy automatu AZI o grafie izomorficznym z grafem WRZ jest problemem kombinatorycznym.

Niżej zostaną przedstawione rozważania istotne dla procesu syntezy. Niech a_i będzie liczbą wierzchołków grafu $G_i \in G^+$, tzn. $a_i = \overline{Q}_i$. Wprowadzona zostaje liczba n określona w następujący sposób:

$$n = \sum_{i \in I} a_i \quad (2.123)$$

Ponieważ przyjęto założenie 2.3., wg którego dane podzadanie występuje nie więcej niż jeden raz w WRZ, a więc po ocenie liczby wierzchołków grafu $\bar{G}^i$ WRZ - dalszej analizie będzie poddany graf $\bar{G}^i$, jeśli liczba jego wierzchołków jest nie większa niż n .

Dany automat AZI może zawierać różną liczbę podautomatów.

Definicja 2.14.

Zbiorem podautomatów w danym automacie AZI oznaczonym symbolem Z_i jest zbiór tych podautomatów ze zbioru L , na których grafach została przeprowadzona operacja $\oplus$ dająca w wyniku rozważany AZI. Rodzina wszystkich zbiorów podautomatów w automatach zagregowanych pierwszego rodzaju możliwych do uzyskania na bazie zbioru L oznaczona jest symbolem Z .

Lemat 2.6.

$$\overline{Z} = 2^k - (k+1), \text{ gdzie } k = \overline{G}^+$$

Dowód.

Automat AZI może składać się z l podautomatów, gdzie $2 \leq l \leq k$. Liczba l spełnia zależność $l \neq 1$ ze względu na ograniczenie $S_j \neq \emptyset$ i wyrażenie (2.16). Liczba l elementowych zbiorów $Z_i \in Z$ wynosi $\binom{k}{l}$.

Z powyższego wynika, że

$$\overline{Z} = \binom{k}{2} + \dots + \binom{k}{k} = 2^k - (k+1) \quad \blacksquare$$

Każdemu zbiorowi Z_i zostaje przyporządkowana liczba h_i będąca liczbą wierzchołków grafów podautomatów przynależnych do zbioru Z_i .

Definicja 2.15.

$$h_i = \sum_{G_s \in Z_i} \overline{Q}_s \quad (2.124)$$

gdzie Q_s - zbiór wierzchołków grafu G_s .

Rodzina Z zostaje podzielona na klasy abstrakcji zawierające te zbiory $Z_i \in Z$, którym przyporządkowana jest ta sama liczba h_i . Klasa abstrakcji wyznaczona przez liczbę r oznaczona jest symbolem Z^r .

Definicja 2.16.

$$Z^r = \{Z_i : Z_i \in Z \wedge h_i = r\} \quad (2.125)$$

Jeśli moc zbioru wierzchołków grafu WRZ równa jest r , to w procesie analizy tego grafu uwzględniane są tylko elementy rodziny Z^r . Wprowadzony wyżej podział rodziny Z umożliwia w wielu przypadkach uzyskanie automatu AZI o grafie izomorficznym z grafem WRZ, po wykonaniu mniejszej liczby operacji niż liczba wymagana w sytuacji, w której ten podział nie zostałby wprowadzony.

Jeśli prawdziwe jest zdanie:

$$(\forall i, j \in I)(\overline{Q}_i = \overline{Q}_j) \quad (2.126)$$

tn. liczby wierzchołków wszystkich grafów ze zbioru G^+ są sobie równe, to wzmiankowany podział nie daje zmniejszenia liczby operacji wymaganych dla realizacji procesu syntezy.

Niżej zostanie określona operacja wykonywana na grafie G^j WRZ i grafie G_i podautomatu. Operacja ta jest istotna dla przedstawionego procesu syntezy. Jej symbolem jest symbol $\odot$.

W celu określenia wyniku rozważanej operacji, dla grafu G_i podautomatu z wyróżnionym wierzchołkiem q_s - interpretowanego jako para uporządkowana $\langle G_i, q_s \rangle$ - zostanie zdefiniowana tzw. część dostępna pary $\langle G_i, q_s \rangle$ oznaczona symbolem $D(\langle G_i, q_s \rangle)$. Definicja $D(\langle G_i, q_s \rangle)$ wymaga wprowadzenia definicji pomocniczych.

Definicja 2.17.

Wierzchołkiem dostępnym z wierzchołka q_s grafu G_i nazywamy wierzchołek q_t lub każdy taki wierzchołek tego grafu, do którego prowadzi droga z wierzchołka q_s .

Jeśli wierzchołek q_t jest wierzchołkiem dostępnym z wierzchołka q_s grafu G_i , to stosujemy następujący zapis:

$$q_t = d(<G_i, q_s>)$$

Zbiór oznaczony symbolem $U(<G_i, q_s>)$ jest zbiorem tych wierzchołków grafu G_i , które są dostępne z wierzchołka q_s .

Definicja 2.18.

$$U(<G_i, q_s>) = \{q_t : q_t = d(<G_i, q_s>)\} \quad (2.127)$$

Definicja 2.19.

Częścią dostępną pary $<G_i, q_s>$ oznaczoną symbolem $D(<G_i, q_s>)$ nazywamy graf, którego zbiór wierzchołków jest zbiorem $U(<G_i, q_s>)$, natomiast zbiór łuków zawiera te łuki grafu G_i , które łączą wierzchołki zbioru $U(<G_i, q_s>)$.

Wniosek 2.4.

Graf $D(<G_i, q_s>)$ jest podgrafem grafu G_i .

Symboliczny zapis operacji $\odot$ wykonanej na grafie $\tilde{G}^j$ z wyróżnionym wierzchołkiem r_k i grafie G_i o wyróżnionym wierzchołku q_s jest następujący:

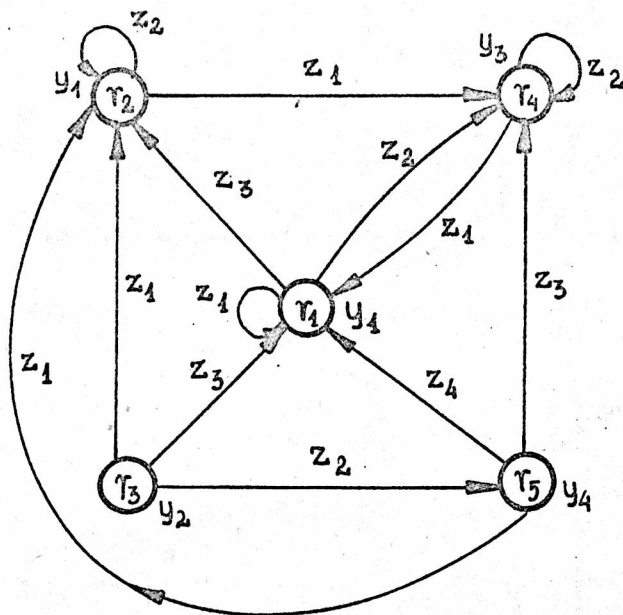
$$<\tilde{G}^j, r_k> \odot <G_i, q_s> = e, \quad \text{gdzie } e \in \{0, 1\} \quad (2.128)$$

Liczba e przyjmuje wartość 1, jeśli w grafie $\tilde{G}^j$ istnieje taki jego podgraf zawierający wierzchołek r_k , który jest izomorficzny z grafem $D(<G_i, q_s>)$, przy czym wierzchołkowi r_k tego podgrafu przyporządkowany jest wierzchołek q_s .

W celu zilustrowania operacji $\odot$ przedstawiony zostanie przykład.

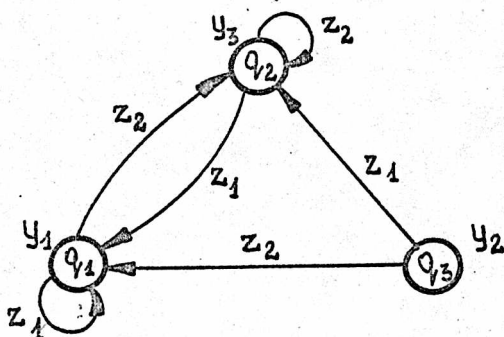
Przykład 2.8.

Niech graf $\bar{G}^j$ wariantu rozwiązania zadania ma postać przedstawioną na rys. 2.24.



Rys. 2.24. Graf $\bar{G}^j$ wariantu rozwiązania zadania

Niech graf G_i podautomatu ma postać przedstawioną na rys. 2.25.



Rys. 2.25. Graf G_i podautomatu A_i

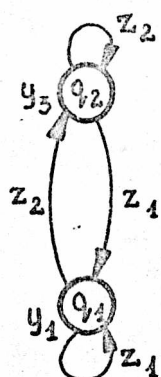
Dążymy do wyznaczenia liczby e_p określonej w następujący sposób:

$$\langle \bar{G}^j, r_1 \rangle \odot \langle G_i, q_1 \rangle = e_p \quad (2.129)$$

Liczba e_p przyjmuje wartość 1, jeśli w grafie $\bar{G}^j$ istnieje taki podgraf, który jest izomorficzny z grafem $D(\langle G_i, q_1 \rangle)$, przy czym wierzchołkowi r_1 przyporządkowany jest wierzchołek q_1 . W przeciwnym przypadku $e_p = 0$.

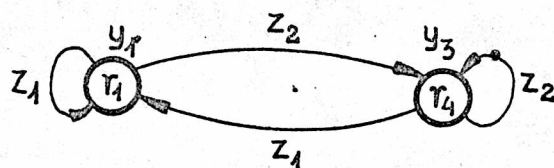
W celu wyznaczenia liczby e_p należy określić graf $D(\langle G_i, q_1 \rangle)$.

Graf ten jest przedstawiony na rys. 2.26.



Rys. 2.26. Graf $D(<G_i, q_1>)$

Liczba $e_p=1$, ponieważ w grafie $\bar{G}^j$ istnieje podgraf $\mathcal{G}_a$, przedstawiony na rys. 2.27., izomorficzny z grafem $D(<G_i, q_1>)$, przy czym wierzchołkowi r_1 przyporządkowany jest wierzchołek q_1 .



Rys. 2.27. Podgraf $\mathcal{G}_a$ grafu $\bar{G}^j$ z rys. 2.24.

Podgrafy grafu $\bar{G}^j$ izomorficzne z grafami typu grafu $D(<G_i, q_s>)$ oznaczane będą symbolami $\mathcal{G}_c$. Dany graf $\mathcal{G}_c$ ma wyróżniony wierzchołek. Wierzchołkiem tym jest wierzchołek, któremu jest przyporządkowany wierzchołek q_s grafu $D(<G_i, q_s>)$.

Przyjęte zostaje następujące założenie dotyczące grafów wariantów rozwiązania zadań.

Założenie 2.4.

Dla każdego wierzchołka r_k grafu $\bar{G}^i$ wariantu rozwiązania zadania istnieje najwyżej jeden graf $D(<G_i, q_j>)$, gdzie $G_i \in G^+$, dla którego istnieje podgraf $\mathcal{G}_c$ grafu $\bar{G}^i$ izomorficzny z grafem $D(<G_i, q_j>)$, pod warunkiem, że wierzchołkowi r_k grafu $\mathcal{G}_c$ ma być przyporządkowany q_j .

Niżej zostanie przedstawione twierdzenie charakteryzujące operację $\odot$ - istotne dla procesu syntezy.

Twierdzenie 2.15.

Niech graf G^k automatu AZI będzie izomorficzny z grafem $\bar{G}^j$ WRZ.

Jeśli

$$\langle \bar{G}^j, r_k \rangle \odot \langle G_i, q_s \rangle = 0$$

to wierzchołkiem grafu G^k przyporządkowanym wierzchołkowi r_k grafu $\bar{G}^j$ nie może być wierzchołek q_s grafu G_i .

Dowód.

Dowód wynika z warunków jakie muszą spełniać izomorficzne grafy G^k , $\bar{G}^j$, określenia operacji $\odot$ i wniosku 2.4. ■

Dla grafu $\bar{G}^i$ i jego podgrafu $\mathcal{G}_c$ zostaje określona operacja oznaczona symbolem $\square$: Przyjęty zostaje następujący zapis:

$$\bar{G}^i \square \mathcal{G}_c = V_a \quad (2.130)$$

Wynikiem operacji zadanej wyrażeniem (2.130) jest zbiór tych wierzchołków grafu $\bar{G}^i$ nie będących wierzchołkami grafu $\mathcal{G}_c$, do których skierowane są łuki z grafu $\mathcal{G}_c$.

Niech trójka uporządkowana $\langle r_i, z_s, r_j \rangle$ będzie przyporządkowana łukowi grafu $\bar{G}^i$ skierowanemu z wierzchołka r_i do r_j , a opisanemu literą $z_s \in Z'$.

Graf $\bar{G}^i$ można określić w następujący sposób:

$$\bar{G}^i = \langle R^i, \mathcal{S}^i, \Pi^i \rangle \quad (2.131)$$

gdzie R^i - zbiór wierzchołków grafu $\bar{G}^i$, $\mathcal{S}^i$ - zbiór trójek uporządkowanych postaci $\langle r_i, z_s, r_j \rangle$ przyporządkowanych łukom $\bar{G}^i$, Π^i - funkcja charakteryzująca obciążenie wierzchołków $\bar{G}^i$ literami y_s .

Graf $\mathcal{G}_c$ można określić następująco:

$$\mathcal{G}_c = \langle R_c, \mathcal{S}_c, \Pi_c \rangle \quad (2.132)$$

gdzie zbiory R_c , $\mathcal{S}_c$, Π_c określone są podobnie jak zbiory $R^i, \mathcal{S}^i, \Pi^i$.

Stąd zbiór V_a zadany wyrażeniem (2.130) można zdefiniować w następujący sposób:

$$V_a = \{ r_j : r_j \in R^i \setminus R_c \wedge (\exists \langle r_k, z_s, r_j \rangle \in \delta^i) (r_k \in R_c) \} \quad (2.133)$$

Dla potrzeb procesu syntezy określony zostanie zbiór oznaczony symbolem $\wp(T, F)$. Niech grafy T, F będą grafami izomorficznymi oraz niech wierzchołkiem T będzie r_i , natomiast wierzchołkiem F - q_j .

Definicja 2.20.

Zbiór $\wp(T, F)$ jest zbiorem par uporządkowanych $\langle r_i, q_j \rangle$, gdzie r_i, q_j są wzajemnie jednoznacznie przyporządkowanymi sobie wierzchołkami izomorficznych grafów T, F .

Dla potrzeb prezentacji procesu syntezy zostaną przedstawione definicje zbiorów oznaczonych symbolami $Q^j, Q^{j,s}$. W celu określenia tych zbiorów zostanie sformułowany lemat 2.7.

Niech Y_i będzie alfabetem wyjściowym podautomatu A_i oraz niech $Q = \bigcup_{i \in I} Q_i$, gdzie Q_i jest zbiorem wierzchołków grafu G_i .

Lemat 2.7.

Alfabet wyjściowy dowolnego automatu AZI, uzyskanego w wyniku wykonania operacji $\oplus^S$ na wybranych grafach ze zbioru G^+ , jest zawarty w zbiorze Y określonym następująco:

$$Y = \bigcup_{i \in I} Y_i$$

Dowód.

Ponieważ operacja $\oplus^S$ nie powoduje wbudowania w graf automatu AZI takiego wierzchołka, który nie należy do zbioru Q , a więc wierzchołki grafu AZI opisane są elementami $y_c \in Y$. Stąd alfabet wyjściowy tego automatu zawarty jest w zbiorze Y . $\square$

Niech γ będzie zbiorem grafów:

$$\gamma = \{ G_1, \dots, G_j, \dots, G_k \} \quad (2.134)$$

Symbol $W(\gamma)$ oznacza rodzinę zbiorów wierzchołków grafów $G_j \in \gamma$

zn.

$$W(\gamma) = \{Q_1, \dots, Q_j, \dots, Q_k\} \quad (2.135)$$

Symbolem Q^j zostaje oznaczony zbiór tych wierzchołków grafów ze zbioru G^+ , które opisane są literą y_j alfabetu Y .

Definicja 2.21.

$$Q^j = \{q_i : q_i \in Q_V \wedge \Psi_V(q_i) = y_j \wedge Q_V \in W(G^+)\} \quad (2.136)$$

gdzie Ψ_V jest funkcją wyjść podautomatu A_V .

Niech zbiór wierzchołków grafu $\bar{G}^i$ WRZ ma moc $\bar{R}^i = r$. Niech zbiór Z_S zawierający pewne grafy ze zbioru G^+ będzie przynależnym do rodziny Z^n określonej za pomocą definicji 2.16. Zbiór $Q^{j,s}$, gdzie s jest indeksem symbolu Z_S , jest zbiorem wierzchołków $q_k \in Q$ opisanych literą $y_j \in Y$ będących wierzchołkami grafów $G_V \in Z_S$.

Definicja 2.22.

$$Q^{j,s} = \{q_i : q_i \in Q_V \wedge \Psi_V(q_i) = y_j \wedge Q_V \in W(Z_S)\} \quad (2.137)$$

Stosowanie w procesie syntezy zbioru $Q^{j,s} \subset Q^j$ zamiast zbioru Q^j , często umożliwia zmniejszenie liczby niezbędnych operacji w tym procesie.

Lemat 2.8.

$$\bar{R}^i = \bar{Q} \Rightarrow (\forall y_j \in Y)(\forall Z_S \in Z^n)(Q^{j,s} = Q^j)$$

Dowód.

Jeśli prawdziwe jest założenie, to istnieje tylko jeden $Z_S \in Z^n$ określony następująco $Z_S = G^+$. Stąd z definicji 2.21 i 2.22 otrzymujemy tezę. $\square$

Z lematu 2.8. wynika, że istnieje taki graf $\bar{G}^i$ WRZ, dla którego stosowanie w procesie syntezy zbioru $Q^{j,s}$ zamiast Q^j nie powoduje

zmniejszenia liczby niezbędnych operacji w tym procesie.

Grafowi G_V wraz z jego wyróżnionym wierzchołkiem q_1 można przyporządkować parę uporządkowaną $\langle G_V, q_1 \rangle$. Zbiorowi Q zawierającemu n wierzchołków można przyporządkować zbiór oznaczony symbolem g - zawierający n takich par.

Definicja 2.23.

$$g = \{ \langle G_V, q_i \rangle : q_i \in Q_V \wedge G_V \in G^+ \} \quad (2.138)$$

gdzie Q_V - zbiór wierzchołków grafu G_V .

Niżej zostanie określony zbiór $g^{j,s}$ - istotny dla sformułowania algorytmu syntezy. Zbiór $g^{j,s}$ zawiera pary postaci $\langle G_V, q_i \rangle$, w których q_i jest elementem zbioru $Q^{j,s}$ i jednocześnie wierzchołkiem G_V .

Definicja 2.24.

$$g^{j,s} = \{ \langle G_V, q_i \rangle : q_i \in Q_V \cap Q^{j,s} \wedge G_V \in Z_s \} \quad (2.139)$$

Niech zbiór φ będzie zbiorem par uporządkowanych $\langle v_i, v_j \rangle$ oraz symbol $\text{ord}^k \langle v_i, v_j \rangle$ ($k \in \{1, 2\}$) niech oznacza k -tą współrzędną pary $\langle v_i, v_j \rangle$. Zbiór $\text{ord}^k \varphi$ jest zbiorem k -tych współrzędnych par należących do zbioru φ .

Definicja 2.25.

$$\text{ord}^k \varphi = \{ v_c : v_c = \text{ord}^k \langle v_i, v_j \rangle \wedge \langle v_i, v_j \rangle \in \varphi \} \quad (2.140)$$

gdzie $k \in \{1, 2\}$.

Automaty zagregowane pierwszego rodzaju zdolne do realizacji konkretnych zadań charakteryzują się cechami określanymi za pomocą wariantów rozwiązania tych zadań. Na przykład tylko do pewnych wierzchołków grafów podautomatów mogą być skierowane łuki z grafów innych podautomatów lub tylko z pewnych wierzchołków grafów podautomatów mogą być skierowane łuki do wierzchołków grafów innych podautomatów. Często poszczególne ograniczenia mogą powodować zmniejszenie liczby operacji niezbędnych w procesie syntezy odpowiedniego AZI, dzięki

możliwości pominięcia pewnych przypadków. W prezentowanym procesie syntezy nie uwzględniono ograniczeń charakterystycznych dla konkretnych zadań. Określenie wszystkich ewentualnych ograniczeń jest niemożliwe.

Niżej zostanie przedstawiony w sposób szkicowy proces syntezy automatu zagregowanego pierwszego rodzaju o grafie izomorficznym z grafem $\bar{G}^i$ wariantu rozwiązania zadania. Oprócz grafu $\bar{G}^i$ podstawową daną jest zbiór G^+ grafów podautomatów.

Część pierwsza.

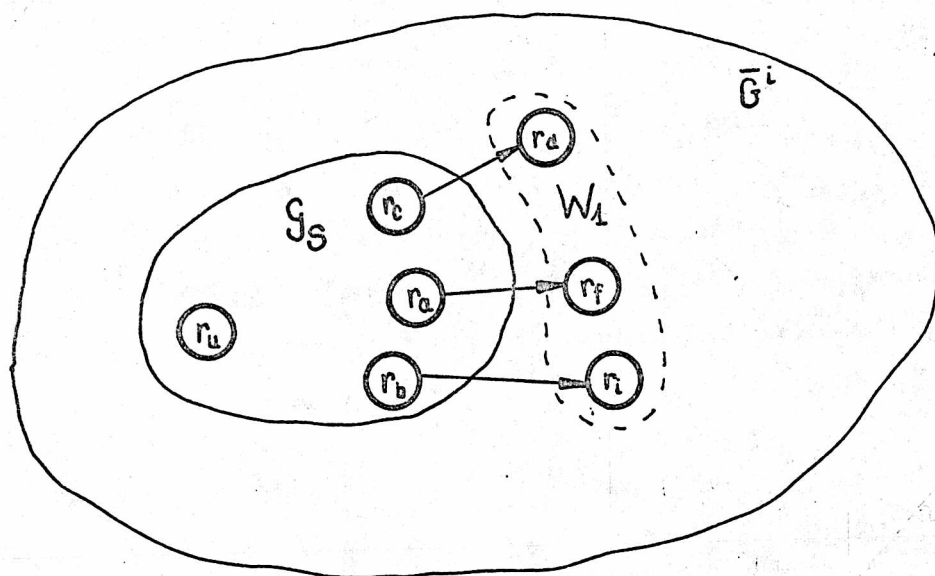
Przyjmujemy, że moc zbioru R^i wierzchołków grafu $\bar{G}^i$ równa się r . Stąd analizie będą poddane zbiory $Z_c \in Z^r$, gdzie rodzina Z^r zadana jest definicją 2.16. Niech badany na początku zbiorem rodziny Z^r będzie Z_0 . Zakładamy, że wierzchołek początkowy r_u grafu $\bar{G}^i$ obciążony jest literą $y_c \in Y$. Wyznaczamy taki graf $D(<G_i, q_r>)$, gdzie $G_i \in Z_0$ i q_r opisany jest literą y_c , tzn. $<G_i, q_r> \in g^{c,s}$, dla którego istnieje podgraf grafu $\bar{G}^i$ izomorficzny z $D(<G_i, q_r>)$, pod warunkiem, że wierzchołkowi r_u ma być przyporządkowany wierzchołek q_r . Na mocy założenia 2.4. może istnieć najwyżej jeden graf $D(<G_i, q_r>)$ spełniający wymagane warunki. Niech szukany grafem będzie $D(<G_0, q_b>)$, a podgrafem grafu $\bar{G}^i$ izomorficznym z tym grafem - graf G_s . Stąd mamy wyznaczone wstępne przyporządkowanie wierzchołkom grafu G_s wierzchołków grafu $D(<G_0, q_b>)$. Przyporządkowanie to nazywane jest wstępnym, ponieważ na dalszych tzw. etapach procesu syntezy może być sformułowany wniosek, że jest ono niewłaściwe tzn. nie prowadzi do znalezienia automatu AZI o grafie izomorficznym z grafem $\bar{G}^i$. Niżej zostanie podana definicja etapu procesu syntezy.

Definicja 2.26.

Etapem procesu syntezy nazywamy część tego procesu związaną z operacjami zorientowanymi na wyznaczenie dla wierzchołka r_u grafu $\bar{G}^i$ WRZ takiego grafu $D(<G_i, q_r>)$, dla którego istnieje podgraf grafu

$\bar{G}^i$ zawierający wierzchołek r_a izomorficzny z $D(\langle G_i, q_r \rangle)$, przy czym wierzchołkowi r_a ma być przyporządkowany wierzchołek q_r .

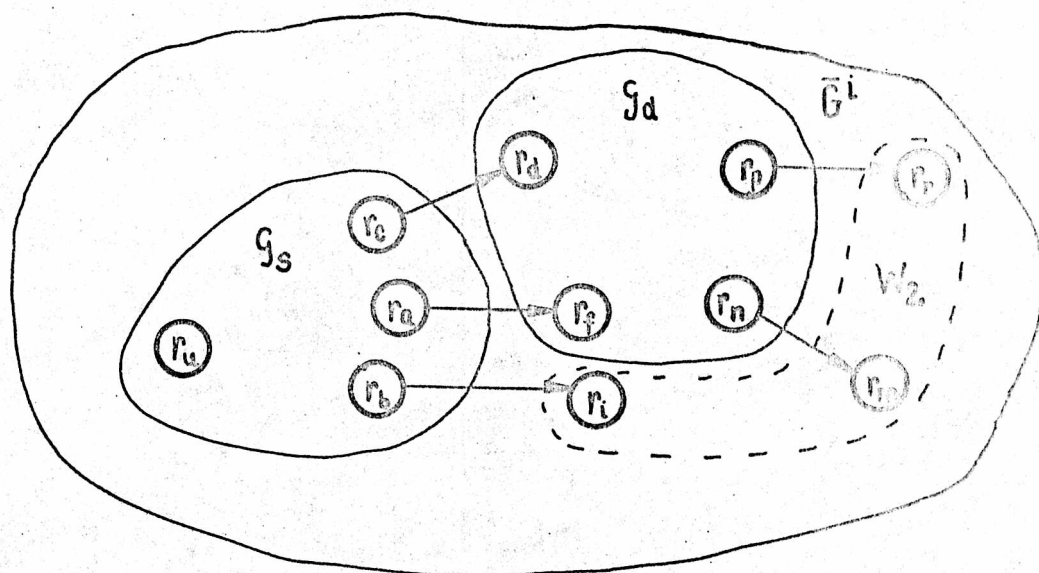
Dla grafu G_S zostaje określony zbiór tych wierzchołków grafu $\bar{G}^i$ nie będących wierzchołkami G_S , do których skierowane są łuki z wierzchołków grafu G_S . Zbiór ten oznaczony jest symbolem W_1 . Niech ilustracją wyników pierwszego etapu procesu syntezy będzie rys. 2.28.



Rys. 2.28. Ilustracja rezultatów pierwszego etapu procesu syntezy

Elementom zbioru W_1 należy wstępnie przyporządkować wierzchołki grafów ze zbioru G^+ . Podobne operacje jak poprzednio dla wierzchołka r_u przeprowadzane są na drugim etapie dla wierzchołka $r_d \in W_1$. Niech grafem izomorficznym z $D(\langle G_v, q_s \rangle)$, gdzie $G_v \in Z_S$, będzie podgraf

G_d grafu $\bar{G}^i$ pod warunkiem, że wierzchołkowi r_d tego podgrafu jest wstępnie przyporządkowany wierzchołek q_s . Po drugim etapie wyznaczone jest wstępne przyporządkowanie wierzchołkom grafów G_S , G_d - elementów zbioru Q . Niech rezultaty uzyskane po drugim etapie mają postać przedstawioną na rys. 2.29.



Rys. 2.29. Ilustracja rezultatów uzyskanych po drugim etapie procesu syntezy

Zbiór W_2 jest zbiorem tych wierzchołków grafu $\bar{G}^1$ nie będących wierzchołkami grafów G_s , G_d , do których skierowane są łuki z G_s , G_d . Należy określić te wierzchołki $q_s \in Q$, które można przyporządkować elementom zbioru W_2 .

Ogólnie na $(j+1)$ -szym etapie procesu syntezy wykonywane są operacje dotyczące wstępnego przyporządkowania jednemu elementowi zbioru W_j - odpowiedniego wierzchołka $q_t \in Q$. W zbiorze tych wierzchołków przyporządkowanych do Q , które poddane są analizie na $(j+1)$ -szym etapie, w celu wybrania jednego z nich jako wierzchołka wstępnie przyporządkowanego elementowi $r_s \in W_j$, nie ma elementów zbioru Q przyporządkowanych wierzchołkom grafu $\bar{G}^1$ w wyniku j etapów procesu syntezy.

W podobny sposób na dalszych etapach procesu syntezy realizowana jest rozpoznawanie podgrafów grafu $\bar{G}^1$ izomorficznych z grafami typu grafu $D(\langle G_i, q_r \rangle)$, gdzie $G_i \in {}^*Z_S$.

Jeśli na $(j+1)$ -szym etapie dla wierzchołka $r_s \in W_j$ nie zostanie znaleziony odpowiedni element zbioru Q , to dla zbioru *Z_S grafów podautomatów nie zostanie określony automat AZI o grafie izomorficznym z grafem $\bar{G}^1$ WRZ. Stąd uzyskano po j etapach wstępne przyporządkowanie

elementom zbioru R^1 wierzchołków grafów ze zbioru G^+ jest niewłaściwe. W tym przypadku podobna analiza jest prowadzona dla innego zbioru rodziny Z^r . Jeśli dla żadnego zbioru $Z_t \in Z^r$ nie zostanie znalezione wstępne przyporządkowanie, w którego wyniku wszystkich wierzchołkom grafu $\bar{G}^1$ będą przyporządkowane wierzchołki grafów ze zbioru Z_t , to dla grafu $\bar{G}^1$ wariantu rozwiązania zadania nie zostanie zbudowany automat AZI o grafie izomorficznym z $\bar{G}^1$.

Jeśli dla pewnego $Z_d \in Z^r$ zostanie znalezione wstępne przyporządkowanie wierzchołków grafów ze zbioru Z_d dla wszystkich wierzchołków grafu $\bar{G}^1$, to następuje przejście do drugiej części procesu syntezy.

Druga część procesu syntezy dotyczy tych łuków grafu $\bar{G}^1$, którym nie są przyporządkowane łuki grafów podautomatów $G_c \in Z_d$ izomorficznych z podgrafami grafu $\bar{G}^1$. Niech zbiór wzmiankowanych łuków grafu $\bar{G}^1$ będzie oznaczony symbolem δ^1 . Na podstawie zbioru δ^1 zostaje określony zbiór S_k . Zbiór ten jest podstawą do wykonania operacji $\oplus_{S_k}$ na grafach $G_c \in Z_d$, w której wyniku zostaje uzyskany graf:

$$G^k = \bigoplus_{G_c \in Z_d}^{S_k} G_c \quad (2.141)$$

automatu zagregowanego pierwszego rodzaju izomorficzny z grafem $\bar{G}^1$ wariantu rozwiązania zadania.

Niżej zostanie przedstawiony proces syntezy automatu zagregowanego pierwszego rodzaju o grafie izomorficznym z grafem $\bar{G}^1$ WRZ.

Dany jest graf $\bar{G}^1$ WRZ z wierzchołkiem początkowym $r_0^1 = r_u$ i zbiór G^+ grafów podautomatów ze zbioru L .

Część pierwsza.

Część ta dotyczy wyznaczania przyporządkowania wierzchołkom grafu $\bar{G}^1$ - wierzchołków grafów ze zbioru G^+ . W celu określenia tego przyporządkowania, wyznaczane jest tzw. wstępne przyporządkowanie

wierzchołkom podzbioru zbioru R^i - wierzchołków grafów ze zbioru G^i oznaczone symbolem φ_j .

Definicja 2.27.

Wstępnym przyporządkowaniem φ_j nazywamy zbiór par postaci $\langle r_k, q_j \rangle$, gdzie $r_k \in R^i$, $q_j \in Q$, określający elementy zbioru Q przyporządkowane wierzchołkom grafu $\bar{G}^i$ po j etapach procesu syntezy.

Niech moc zbioru wierzchołków grafu $\bar{G}^i$ $\bar{R}^i = r$. Stąd dalszym rozważaniami będą poddane zbiory rodziny Z^i określonej za pomocą definicji 2.16.

Niech elementem rodziny Z^i będzie zbiór Z_g oraz niech

$$\pi^i(r_u) = y_c \quad (2.142)$$

gdzie π^i - funkcja charakteryzująca obciążenie wierzchołków $\bar{G}^i$. Stąd wierzchołkowi r_u może być przyporządkowany wierzchołek $q_i \in \text{ord}^2 G^{c,s}$. Dla pary $\langle G_v, q_i \rangle \in G^{c,s}$ zostaje wykonana następująca operacja:

$$\langle \bar{G}_i, r_u \rangle \odot \langle G_v, q_i \rangle = e_p \quad (2.143)$$

Jeśli $e_p = 0$, to na bazie zbioru Z_g nie można zbudować automatu AZI o grafie izomorficznym z grafem $\bar{G}^i$.

Niech $e_p = 1$ oraz niech graf G_t będzie podgrafem grafu $\bar{G}^i$ izomorficznym z grafem $D(\langle G_v, q_i \rangle)$ pod warunkiem, że wierzchołkowi r_u grafu G_t przyporządkowany jest wierzchołek q_i .

Przez przyporządkowanie grafowi G_t - grafu $D(\langle G_v, q_i \rangle)$, zostaje określone wstępne przyporządkowanie φ_1 zgodnie z wyrażeniem:

$$\varphi_1 = \wp(G_t, D(\langle G_v, q_i \rangle)) \quad (2.144)$$

gdzie $\wp(G_t, D(\langle G_v, q_i \rangle))$ jest zbiorem określonym za pomocą definicji 2.20. Stąd φ_1 jest zbiorem par $\langle r_i, q_o \rangle$, których współrzędne są wzajemnie jednoznacznie przyporządkowanymi sobie wierzchołkami obu wyżej rozważanych grafów.

Następnie zostaje wykonana operacja:

$$W_1 = \bar{G}^1 \sqcup \mathcal{G}_t \quad (2.145)$$

Prowadzone rozważania wymagają wprowadzenia pewnych zbiorów i operacji.

Zbiór $g_j^{c,s} \subset g^{c,s}$ zawiera te elementy zbioru $g^{c,s}$ określonego za pomocą definicji 2.24., dla których po j etapach procesu syntezy nie wyznaczono jeszcze wstępnego przyporządkowania φ_j .

Niech k_s będzie podzbiorem zbioru wierzchołków grafów ze zbioru G^+ tzn. $k_s \subset Q$. Niżej zostanie przedstawiona definicja zbioru oznaczonego symbolem $p(k_s)$. Zbiór ten zawiera te pary $\langle G_v, q_i \rangle \in g$ (zbiór g określony jest za pomocą definicji 2.23.), których druga współrzędna $q_i \in k_s$.

Definicja 2.28.

$$p(k_s) = \{ \langle G_v, q_i \rangle : \langle G_v, q_i \rangle \in g \wedge q_i \in k_s \} \quad (2.146)$$

Zbiór $W_j \subset R^1$ określony jest następująco.

Definicja 2.29.

Zbiorem W_j nazywamy zbiór tych wierzchołków grafu $\bar{G}^1$, do których skierowane są łuki z wierzchołków podgrafów $\mathcal{G}_u$ grafu $\bar{G}^1$ wyznaczonych w wyniku j etapów procesu syntezy, przy czym do W_j należą tylko te wierzchołki grafu $\bar{G}^1$, dla których jeszcze nie wyznaczono wstępnego przyporządkowania φ_j .

Na $(j+1)$ -szym etapie dla jednego z elementów $r_s \in W_j$ podjęte zostają działania, których celem jest przyporządkowanie wierzchołkowi r_s - wierzchołka jednego z grafów ze zbioru G^+ .

Niech po j etapach procesu syntezy określone będą następujące zbiory:

$$\varphi_j, W_j, g_j^{c,s} \quad (2.147)$$

Jeśli zbiór pierwszych współrzędnych wstępnego przyporządkowania φ_j jest równy zbiorowi wierzchołków grafu $\bar{G}^1$ tzn.

$$\text{crd}^1 \varphi_j = R^1 \quad (2.148)$$

to część pierwsza procesu syntezy jest zakończona.

Jeśli w wyniku j etapów procesu syntezy nie wszystkim wierzchołkom grafu $\bar{G}^i$ zostały przyporządkowane wierzchołki grafów ze zbioru G^+ tzn.

$$\text{ord}^1 \varphi_j \not\subseteq R^i \quad (2.149)$$

to przeprowadzone zostaną przedstawione niżej operacje.

Niech prawdziwa będzie zależność przedstawiona za pomocą wyrażenia (2.149) oraz niech $r_t \in W_j$ i $\Pi^i(r_t) = y_d$. Na $(j+1)$ -szym etapie wierzchołkowi r_t ma być wstępnie przyporządkowany wierzchołek jednego z grafów ze zbioru G^+ .

Dla pary uporządkowanej $\langle G_c, q_p \rangle \in g_j^{d,s}$, gdzie indeks d jest indeksem symbolu y_d , wykonana zostaje następująca operacja:

$$\langle \bar{G}^i, r_t \rangle \odot \langle G_c, q_p \rangle = e_s \quad (2.150)$$

Jeśli $e_s = 0$, to podobna operacja wykonywana jest dla innej pary

$\langle G_t, q_r \rangle \in g_j^{d,s}$. Niech $e_s = 0$ oraz niech prawdziwa będzie zależność:

$$\langle \bar{G}^i, r_t \rangle \odot \langle G_t, q_r \rangle = 1 \quad (2.151)$$

Niech podgrafem grafu $\bar{G}^i$ izomorficznym z grafem $D(\langle G_t, q_r \rangle)$ będzie graf oznaczony symbolem $\mathcal{G}_s$.

Graf $\mathcal{G}_s$ poddany jest analizie, której podstawą jest przedstawione niżej twierdzenie.

Twierdzenie 2.16.

Z. Niech dane będzie wstępne przyporządkowanie φ_j uzyskane po j etapach procesu syntezy. Niech na $(j+1)$ -szym etapie zostanie stwierdzone, że grafem izomorficznym z grafem $D(\langle G_s, q_m \rangle)$ jest $\mathcal{G}_c$. Stąd na tym etapie zostanie wyznaczony zbiór:

$$\mathcal{S}(\mathcal{G}_c, D(\langle G_s, q_m \rangle)) \quad (1^0)$$

Niech prawdziwe będzie zdanie:

$$(\exists \langle r_j, q_d \rangle \in \wp(\wp_c, D(\langle G_s, q_m \rangle))) (\exists \langle r_j, q_c \rangle \in \varphi_j) (q_d \neq q_c) \quad (2^0)$$

(tzn. zbiór $\wp(\wp_c, D(\langle G_s, q_m \rangle))$ zawiera parę o pierwszej współrzędnej identycznej z pierwszą współrzędną pewnej pary zbioru φ_j , przy czym drugie współrzędne tych dwu par są różne).

T. Wstępne przyporządkowanie φ_{j+1} nie może być określone w następujący sposób:

$$\varphi_{j+1} = \varphi_j \cup \wp(\wp_c, D(\langle G_s, q_m \rangle)) \quad (3^0)$$

Dowód.

Jeśli prawdziwe jest zdanie (2^0) , to zbiór φ_{j+1} określony za pomocą wyrażenia (3^0) zawierałby m. in. pary $\langle r_j, q_d \rangle$ i $\langle r_j, q_c \rangle$. Stąd wierzchołkowi r_j grafu $\bar{G}^i$ byłyby przyporządkowane dwa różne wierzchołki q_d, q_c , a zatem zbiór φ_{j+1} określony za pomocą wyrażenia (3^0) nie może być podstawą do wyznaczenia przyporządkowania wierzchołkom grafu $\bar{G}^i$ - wierzchołków grafu automatu AZI izomorficznego z grafem $\bar{G}^i$. $\square$

Dla grafu G_s izomorficznego z grafem $D(\langle G_t, q_r \rangle)$ niech prawdziwe będzie zdanie:

$$\neg (\exists \langle r_i, q_f \rangle \in \wp(\wp_s, D(\langle G_t, q_r \rangle))) (\exists \langle r_i, q_h \rangle \in \varphi_j) (q_f \neq q_h) \quad (2.152)$$

W tym przypadku zbiór φ_{j+1} określony jest następująco:

$$\varphi_{j+1} = \varphi_j \cup \wp(\wp_s, D(\langle G_t, q_r \rangle)) \quad (2.153)$$

Z wyrażenia (2.153) wynika, że po $(j+1)$ etapach wstępne przyporządkowanie φ_{j+1} zawiera oprócz par $\langle r_c, q_b \rangle \in \varphi_j$ również pary, których współrzędne są wzajemnie jednoznacznie przyporządkowanymi sobie wierzchołkami izomorficznych grafów $G_s, D(\langle G_t, q_r \rangle)$.

Gdyby zdanie przedstawione za pomocą wyrażenia (2.152) nie było prawdziwe, to wstępne przyporządkowanie φ_j byłoby niewłaściwe.

Stąd należałoby poddać analizie inny zbiór grafów $Z_u \in Z^r$.

W celu wyznaczenia zbioru W_{j+1} wykonana zostaje operacja:

$$V_d = \bar{G}^i \boxminus G_s \quad (2.154)$$

Zbiór V_d jest zbiorem tych wierzchołków grafu $\bar{G}^i$, nie będących wierzchołkami grafu G_s , do których skierowane są łuki z wierzchołków G_s . Podstawą do określenia zbioru W_{j+1} jest następujący lemat.

Lemat 2.9.

Niech na $(j+1)$ -szym etapie procesu syntezy zostanie wyznaczony zbiór V_d określony wyrażeniem (2.154).

Zbiór W_{j+1} zadany jest zależnością:

$$W_{j+1} = (W_j \cup V_d) \setminus \text{crd}^1 \varphi_{j+1}$$

Dowód.

Jest możliwe, że dla pewnych wierzchołków zbiorów W_j , V_d zostało już wyznaczone wstępne przyporządkowanie w wyniku $j+1$ etapów procesu syntezy. Elementami zbioru W_{j+1} mogą być tylko te wierzchołki grafu $\bar{G}^i$, dla których wstępne przyporządkowanie φ_{j+1} nie zostało jeszcze wyznaczone. Z powyższego otrzymujemy wyrażenie określające zbiór W_{j+1} .

Sposób wyznaczenia zbiorów $g_{j+1}^{c,s}$, gdzie $z_s \in Z^r$ i $y_c \in Y$, zadany jest za pomocą lematu 2.10. Zbiór wierzchołków grafu G_b dostępnych z wierzchołka q_c oznaczony jest symbolem $U(<G_b, q_c>)$.

Lemat 2.10.

Zakładamy, że na $(j+1)$ -szym etapie procesu syntezy dla grafu $D(<G_b, q_c>)$ znaleziono izomorficzny z nim podgraf grafu $\bar{G}^i$.

Zbiór $g_{j+1}^{c,s}$ zadany jest wyrażeniem:

$$g_{j+1}^{c,s} = g_j^{c,s} \setminus p(U(<G_b, q_c>))$$

Dowód.

Zbiór $g_{j+1}^{c,s}$ powinien zawierać te pary $<G_p, q_t> \in g_j^{c,s}$, których drugie współrzędne q_t na $(j+1)$ -szym etapie nie zostały wstępnie

przyporządkowane wierzchołkom grafu $\bar{G}^i$.

Na $(j+1)$ -szym etapie wyznaczono wstępne przyporządkowanie dla wierzchołków grafu $D(<G_b, q_c>)$.

Elementami zbioru $p(U(<G_b, q_c>))$, na mocy definicji 2.28., są pary $<G_v, q_i>$, których drugie współrzędne $q_i \in U(<G_b, q_c>)$.

Z powyższego otrzymujemy zależność określającą zbiór $g_{j+1}^{c,s}$. $\square$

Jeśli na pierwszym etapie dla grafu $D(<G_v, q_i>)$ został określony izomorficzny z nim podgraf grafu $\bar{G}^i$, to zbiór $g_1^{c,s}$ zadany jest wyrażeniem

$$g_1^{c,s} = g^{c,s} \setminus p(U(<G_v, q_i>)) \quad (2.155)$$

Jeśli dla zbioru $^*Z_s \in ^*Z^i$ nie zostanie znalezione wstępne przyporządkowanie φ_k spełniające zależność:

$$\text{crd}^1 \varphi_k = R^i \quad (2.156)$$

(tzn. jeśli istnieją takie wierzchołki grafu $\bar{G}^i$, dla których nie znaleziono odpowiednich elementów $q_t \in Q$), to podobne operacje przeprowadzone są dla innego zbioru $^*Z_t \in ^*Z^i$.

Jeśli dla żadnego ze zbiorów $^*Z_u \in ^*Z^i$ nie zostanie określone odpowiednie wstępne przyporządkowanie, to dla grafu $\bar{G}^i$ WRZ nie zostanie zbudowany, na bazie zbioru L podautomatów, automat AZI o grafie izomorficznym z grafem $\bar{G}^i$.

Niech dla wstępnego przyporządkowania φ_w uzyskanego dla zbioru $^*Z_v \in ^*Z^i$ prawdziwa będzie zależność:

$$\text{crd}^1 \varphi_w = R^i \quad (2.157)$$

Stąd pierwsza część procesu syntezy jest zakończona.

Druga część.

Przyporządkowanie φ_w jest podstawą do drugiej części procesu syntezy. W wyniku pierwszej części został określony zbiór *Z_v zawierający te grafy podautomatów, które są izomorficzne z podgrafami grafu $\bar{G}^i$ WRZ. Dzięki temu zostały wyróżnione te łuki grafu $\bar{G}^i$, którym są

przyporządkowane łuki grafów ze zbioru Z_V . Należy określić zbiór S_k będący podstawą do wykonania operacji:

$$G^k = \bigoplus_{G_j \in Z_V}^{S_k} G_j \quad (2.158)$$

w której wyniku zostanie uzyskany graf G^k - izomorficzny z grafem $\bar{G}^i$. W tym celu zostaje wyznaczony zbiór:

$$\mathcal{G}_V^i = \bigcup_{G_j \in Z_V} \bar{\Phi}_j \quad (2.159)$$

gdzie $\bar{\Phi}_j$ - zbiór trójek postaci $\langle q_i, z_s, q_j \rangle$ przyporządkowanych łukom grafu G_j podautomatu. Zbiór $\mathcal{G}_V^i$ jest zbiorem trójek postaci $\langle q_i, z_s, q_j \rangle$ przyporządkowanych łukom grafów $G_j \in Z_V$. Ponieważ zbiór φ_w na mocy twierdzenia 2.16. jest funkcją różnowartościową, a więc istnieje funkcja odwrotna względem φ_w .

Dla trójki $\langle q_i, z_s, q_j \rangle \in \mathcal{G}_V^i$ przyporządkowanej łukowi grafu podautomatu można określić następującą trójkę:

$$\langle r_p, z_s, r_t \rangle = \langle \varphi_w^{-1}(q_i), z_s, \varphi_w^{-1}(q_j) \rangle \quad (2.160)$$

gdzie φ_w^{-1} - funkcja odwrotna względem funkcji φ_w . Trójka $\langle r_p, z_s, r_t \rangle$ jest przyporządkowana temu łukowi grafu $\bar{G}^i$, któremu przyporządkowany jest łuk grafu pewnego podautomatu związany z trójką $\langle q_i, z_s, q_j \rangle$. Dla zbioru $\mathcal{G}_V^i$ zawierającego trójki postaci $\langle q_i, z_s, q_j \rangle$ zostaje wyznaczony zbiór $\mathcal{G}_V^i$, którego elementami są trójki określone za pomocą wyrażenia (2.160). Zbiór $\mathcal{G}_V^i$ określony jest następująco:

$$\mathcal{G}_V^i = \{ \langle r_p, z_s, r_t \rangle : \varphi_w^{-1}(q_i) = r_p \wedge \varphi_w^{-1}(q_j) = r_t \wedge \langle q_i, z_s, q_j \rangle \in \mathcal{G}_V^i \} \quad (2.161)$$

Z kolei zostaje wyznaczony następujący zbiór:

$$\mathcal{G}^i = \mathcal{G}^i \setminus \mathcal{G}_V^i \quad (2.162)$$

gdzie $\mathcal{G}^i$ - zbiór trójek $\langle r_k, z_s, r_c \rangle$ przyporządkowanych łukom grafu

$\bar{G}^i$. Zbiór δ^i zawiera trójki $\langle r_d, z_p, r_s \rangle$ przyporządkowane tym łukom grafu $\bar{G}^i$, którym nie są przyporządkowane łuki grafów podautomatów przynależnych do Z_v . Trójkom $\langle r_d, z_p, r_s \rangle \in \delta^i$ należy przyporządkować trójki $\langle q_r, z_s, q_t \rangle$ będące elementami zbioru S_k . Zbiór ten określa operację $\oplus_{S_k}$, w której wyniku zostaje uzyskany graf G^k zadany wyrażeniem (2.158), izomorficzny z grafem $\bar{G}^i$. Dzięki określonymu w pierwszej części przyporządkowaniu φ_w , zbiór S_k zostaje określony następująco:

$$S_k = \{ \langle q_r, z_s, q_t \rangle : q_r = \varphi_w(r_i) \wedge q_t = \varphi_w(r_j) \wedge \langle r_i, z_s, r_j \rangle \in \delta^i \} \quad (2.163)$$

Z kolei dla grafu G^k zostaje określona jego postać zagregowana.

Niżej zostanie przedstawiony algorytm syntezy odpowiedniego automatu AZI a następnie - prosty przykład zastosowania tego algorytmu.

Algorytm syntezy formalnej automatu AZI zdolnego do realizacji zadania zgodnie z nowym wariantem jego rozwiązania.

1. Wyznaczyć liczbę wierzchołków grafu $\bar{G}^i$ wariantu rozwiązania zadania:

$$r = \overline{R^i}$$

2. Jeśli nie istnieje zbiór Z^r , to skoczyć do 25.

3. Ze zbioru Z^r wybrać element Z_s .

4. Określić literę:

$$y_c = \Pi^i(r_u)$$

którą opisany jest wierzchołek początkowy r_u grafu $\bar{G}^i$.

5. Jeśli nie istnieje para $\langle G_z, q_j \rangle \in g^{c,s}$, dla której prawdziwa jest zależność:

$$\langle \bar{G}^i, r_u \rangle \odot \langle G_z, q_j \rangle = 1$$

to skoczyć do 19.

6. Określić podgraf G_t grafu $\bar{G}^i$ izomorficzny z grafem $D(\langle G_z, q_j \rangle)$ pod warunkiem, że wierzchołkowi r_u jest przyporządkowany wierzchołek q_j .

7. $\varphi_1 = \rho(G_t, D(\langle G_z, q_j \rangle))$

8.
$$W_1 = \bar{G}^i \boxminus G_t$$

9. Dla zbiorów $g^{a,s}$ wykonać operacje:

$$g_1^{a,s} = g^{a,s} \setminus p(U(\langle G_z, q_j \rangle))$$

i skoczyć do 11.

10. Jeśli:

$$\text{crd}^1 \quad \varphi_j = R^i$$

to skoczyć do 20.

11. Dla wybranego ze zbioru W_j wierzchołka r_t określić literę:

$$y_d = \prod^i(r_t)$$

12. Jeśli nie istnieje para $\langle G_b, q_c \rangle \in g_j^{d,s}$ spełniająca zależność:

$$\langle \bar{G}^i, r_t \rangle \odot \langle G_b, q_c \rangle = 1$$

to skoczyć do 19.

13. Określić podgraf G_u grafu $\bar{G}^i$ izomorficzny z grafem $D(\langle G_b, q_c \rangle)$ pod warunkiem, że wierzchołkowi r_t jest przyporządkowany wierzchołek q_c .

14. Jeśli jest prawdziwe zdanie:

$$(\exists \langle r_k, q_z \rangle \in p(G_u, D(\langle G_b, q_c \rangle))) (\exists \langle r_k, q_v \rangle \in \varphi_j) (q_z \neq q_v)$$

to skoczyć do 19.

15.

$$\varphi_{j+1} = \varphi_j \cup p(G_u, D(\langle G_b, q_c \rangle))$$

16.

$$V_d = \bar{G}^i \boxminus G_u$$

17.

$$W_{j+1} = (W_j \cup V_d) \setminus \text{crd}^1 \varphi_{j+1}$$

18. Dla zbiorów $g_j^{a,s}$ wykonać operacje:

$$g_{j+1}^{a,s} = g_j^{a,s} \setminus p(U(\langle G_b, q_c \rangle))$$

i skoczyć do 10, zwiększając indeks j o 1.

19. Jeśli w zbiorze Z^r jest element dotąd nie analizowany, to skoczyć do 5. W przeciwnym przypadku skoczyć do 25.

20. Dla zbioru Z_S grafów podautomatów wykonać operację:

$$\mathcal{G}_S^i = \bigcup_{G_j \in Z_S} \Phi_j$$

21. Na podstawie zbioru $\mathcal{G}_S^i$ wyznaczyć zbiór $\mathcal{G}_S^i$ zgodnie z wyrażeniem (2.161).

22.

$$\mathcal{G}^i = \mathcal{G}^i \setminus \mathcal{G}_S^i$$

23. Na podstawie zbioru $\mathcal{G}^i$ wyznaczyć zbiór S_k zgodnie z wyrażeniem (2.163).

24.

$$G^k = \bigoplus_{G_j \in Z_S} G_j$$

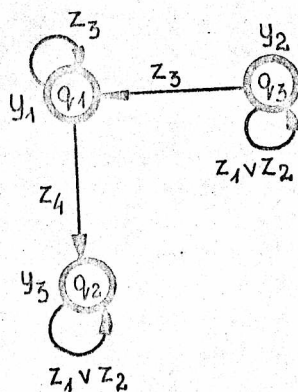
i skoczyć do 26.

25. Sygnalizować, że nie został zbudowany automat AZI zdolny do realizacji zadania zgodnie z wariantem jego rozwiązania wyrażonym grafem $\bar{G}^i$.

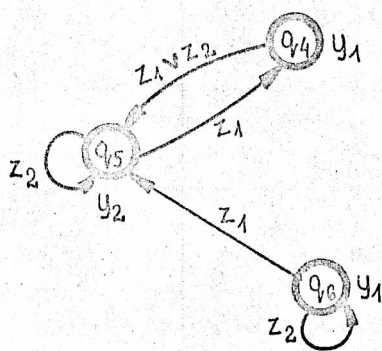
26. Stop.

Przykład 2.9.

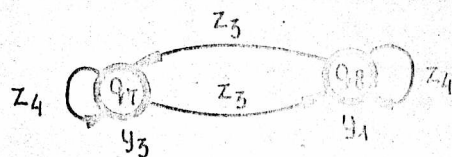
Zbiór Z_S zawiera grafy G_1, G_2, G_3 podautomatów przedstawione na rys. 2.30.



G_1



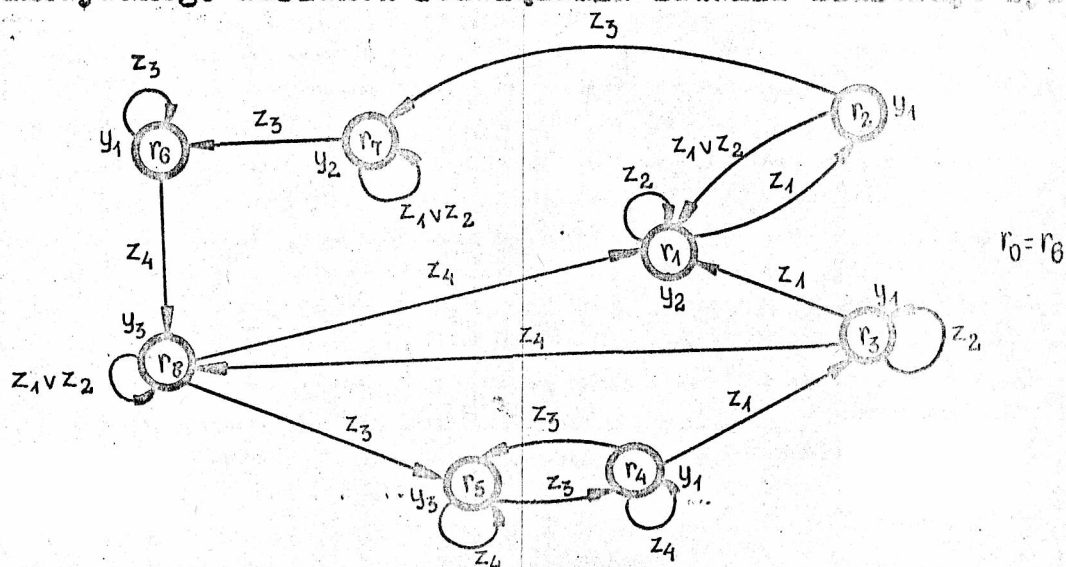
G_2



G_3

Rys. 2.30. Grafy G_1, G_2, G_3 podautomatów

Graf $\bar{G}^1$ opanowywanego wariantu rozwiązania zadania ilustruje rys. 2.31.



Rys. 2.31. Graf $\bar{G}^1$ opanowywanego WRZ

Zbiory $q^{j,s}$ zadane definicją 2.22. określone są następująco:

$$q^{1,s} = \{q_1, q_4, q_6, q_8\} \quad q^{2,s} = \{q_3, q_5\} \quad q^{3,s} = \{q_2, q_7\} \quad (2.164)$$

Zbiory $g^{j,s}$ określone zgodnie z definicją 2.24. mają następującą postać:

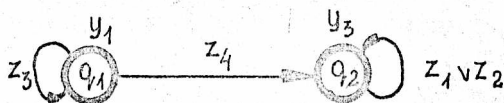
$$\begin{aligned} g^{1,s} &= \{ \langle G_1, q_1 \rangle, \langle G_2, q_4 \rangle, \langle G_2, q_6 \rangle, \langle G_3, q_8 \rangle \} \\ g^{2,s} &= \{ \langle G_1, q_3 \rangle, \langle G_2, q_5 \rangle \} \\ g^{3,s} &= \{ \langle G_1, q_2 \rangle, \langle G_3, q_7 \rangle \} \end{aligned} \quad (2.165)$$

Wierzchołek początkowy r_6 grafu $\bar{G}^1$ obciążony jest literą y_1 .

Istnieje para $\langle G_1, q_1 \rangle \in g^{1,s}$, dla której prawdziwa jest zależność:

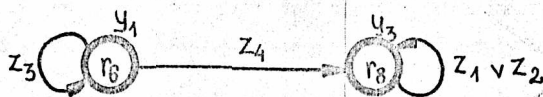
$$\langle \bar{G}^1, r_6 \rangle \odot \langle G_1, q_1 \rangle = 1 \quad (2.166)$$

Dla grafu G_1 zostaje wyznaczony graf $D(\langle G_1, q_1 \rangle)$ - przedstawiony na rys. 2.32.



Rys. 2.32. Graf $D(\langle G_1, q_1 \rangle)$

Podgraf G_t grafu G^1 izomorficzny z grafem $D(<G_1, q_1>)$ ilustruje rys. 2.33.



Rys. 2.33. Graf G_t

Przyporządkowanie wierzchołkom grafu G_t wierzchołków grafu $D(<G_1, q_1>)$ zadane jest wyrażeniem:

$$\rho(G_t, D(<G_1, q_1>)) = \{<r_6, q_1>, <r_8, q_2>\} \quad (2.167)$$

Stąd wstępne przyporządkowanie φ_1 po pierwszym etapie procesu sym- tozy określone jest następująco:

$$\varphi_1 = \{<r_6, q_1>, <r_8, q_2>\} \quad (2.168)$$

Zbiór W_1 zawierający te wierzchołki grafu G^1 nie będące wierzchołkami grafu G_t , do których skierowane są łuki z wierzchołków grafu G_t określony jest wyrażeniem:

$$W_1 = G^1 \square G_t = \{r_1, r_5\} \quad (2.169)$$

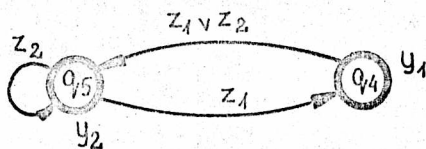
W celu uproszczenia rozważań zostaną pominięte operacje wykonywane na zbiorach $G_j^{a,s}$.

Dalszej analizie będzie poddany wierzchołek $r_1 \in W_1$. Wierzchołek ten obciążony jest literą y_2 .

Istnieje para $<G_2, q_5> \in G^{2,s}$, dla której prawdziwa jest zależność:

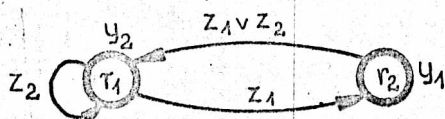
$$<G^1, r_1> \odot <G_2, q_5> = 1 \quad (2.170)$$

Graf $D(<G_2, q_5>)$ przedstawiony jest na rys. 2.34.



Rys. 2.34. Graf $D(<G_2, q_5>)$

Podgraf $\mathcal{G}_u$ grafu $\mathcal{G}^1$ izomorficzny z grafem $D(\langle G_2, q_5 \rangle)$ ilustruje rys. 2.35.



Rys. 2.35. Graf $\mathcal{G}_u$

Przyporządkowanie wierzchołkom grafu $\mathcal{G}_u$ wierzchołków grafu $D(\langle G_2, q_5 \rangle)$ określone jest wyrażeniem:

$$\rho(\mathcal{G}_u, D(\langle G_2, q_5 \rangle)) = \{ \langle r_2, q_4 \rangle, \langle r_1, q_5 \rangle \} \quad (2.171)$$

Ponieważ nie jest prawdziwe zdanie:

$$(\exists \langle r_k, q_z \rangle \in \rho(\mathcal{G}_u, D(\langle G_2, q_5 \rangle))) (\exists \langle r_k, q_v \rangle \in \varphi_1) (q_z \neq q_v) \quad (2.172)$$

a więc wstępne przyporządkowanie φ_2 po dwu etapach procesu syntezy, zadane jest wyrażeniem:

$$\varphi_2 = \varphi_1 \cup \rho(\mathcal{G}_u, D(\langle G_2, q_5 \rangle)) = \{ \langle r_6, q_1 \rangle, \langle r_8, q_2 \rangle, \langle r_2, q_4 \rangle, \langle r_1, q_5 \rangle \} \quad (2.173)$$

Z kolei zostaje wyznaczony zbiór V_d określony następująco:

$$V_d = \mathcal{G}^1 \boxminus \mathcal{G}_u = \{ r_7 \} \quad (2.174)$$

Zbiór W_2 ma następującą postać:

$$W_2 = (W_1 \cup V_d) \setminus \text{crd}^1 \varphi_2 = \{ r_5, r_7 \} \quad (2.175)$$

Dla zbioru $\text{crd}^1 \varphi_2$ prawdziwa jest zależność:

$$\text{crd}^1 \varphi_2 \not\subseteq R^i \quad (2.176)$$

Stąd kontynuowany jest proces wyznaczania wstępnego przyporządkowania.

Dalszym rozważaniom będzie poddany wierzchołek $r_5 \in W_2$.

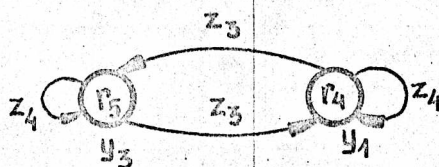
Wierzchołek ten obciążony jest literą y_3 .

Istnieje para $\langle G_3, q_7 \rangle \in g^{3,s}$, dla której prawdziwe jest wyrażenie:

$$\langle \bar{G}^i, r_5 \rangle \odot \langle G_3, q_7 \rangle = 1 \quad (2.177)$$

Graf $D(\langle G_3, q_7 \rangle)$ jest identyczny z grafem G_3 z rys. 2.30.

Podgraf $\mathcal{G}_c$ grafu $\bar{G}^i$ izomorficzny z grafem $D(\langle G_3, q_7 \rangle)$ przedstawiony jest na rys. 2.36.



Rys. 2.36. Graf $\mathcal{G}_c$

Przyporządkowanie wierzchołkom grafu $\mathcal{G}_c$ wierzchołków grafu $D(\langle G_3, q_7 \rangle)$ określone jest następująco:

$$\mathcal{P}(\mathcal{G}_c, D(\langle G_3, q_7 \rangle)) = \{ \langle r_5, q_7 \rangle, \langle r_4, q_8 \rangle \} \quad (2.178)$$

Wstępne przyporządkowanie Ψ_3 ma postać:

$$\Psi_3 = \{ \langle r_6, q_1 \rangle, \langle r_8, q_2 \rangle, \langle r_2, q_4 \rangle, \langle r_1, q_5 \rangle, \langle r_5, q_7 \rangle, \langle r_4, q_8 \rangle \} \quad (2.179)$$

Następnie zostaje wyznaczony zbiór:

$$V_F = \bar{G}^i \boxminus \mathcal{G}_c = \{ r_3 \} \quad (2.180)$$

Zbiór W_3 zadany jest wyrażeniem:

$$W_3 = (W_2 \cup V_F) \setminus \text{crd}^1 \Psi_3 = \{ r_7, r_3 \} \quad (2.181)$$

Zbiór $\text{crd}^1 \Psi_3$ spełnia zależność:

$$\text{crd}^1 \Psi_3 \not\subseteq R^i \quad (2.182)$$

Dalszym rozważaniom poddany będzie wierzchołek $r_7 \in W_3$ obciążo-

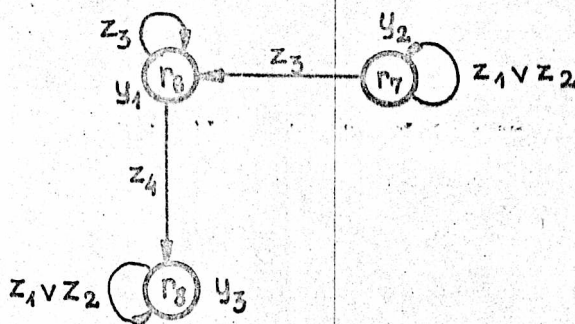
ny literą y_2 . Dla pary $\langle G_1, q_3 \rangle \in g^{2,s}$ prawdziwe jest wyrażenie:

$$\langle \bar{G}^i, r_7 \rangle \odot \langle G_1, q_3 \rangle = 1 \quad (2.183)$$

Graf $D(\langle G_1, q_3 \rangle)$ jest identyczny z grafem G_1 .

Podgraf $\mathcal{G}_p$ grafu $\bar{G}^i$ izomorficzny z grafem G_1 ilustruje rys.

2.37.



Rys. 2.37. Graf $\mathcal{G}_p$

Przyporządkowanie wierzchołkom grafu $\mathcal{G}_p$ wierzchołków grafu $D(\langle G_1, q_3 \rangle)$ określone jest za pomocą wyrażenia:

$$\wp(\mathcal{G}_p, D(\langle G_1, q_3 \rangle)) = \{ \langle r_6, q_1 \rangle, \langle r_7, q_3 \rangle, \langle r_8, q_2 \rangle \} \quad (2.184)$$

Zbiór określony za pomocą wyrażenia (2.184) zawiera m. in. pary $\langle r_6, q_1 \rangle, \langle r_8, q_2 \rangle$, których pierwszymi współrzędnymi są wierzchołki r_6, r_8 . Wierzchołki te są pierwszymi współrzędnymi par $\langle r_6, q_1 \rangle, \langle r_8, q_2 \rangle \in \wp_3$. Z powyższego wynika, że prawdziwe jest zdanie:

$$\neg (\exists \langle r_k, q_z \rangle \in \wp(\mathcal{G}_p, D(\langle G_1, q_3 \rangle))) (\exists \langle r_k, q_r \rangle \in \wp_3) (q_z \neq q_r) \quad (2.185)$$

Zatem wstępne przyporządkowanie $\wp_4$ ma postać:

$$\wp_4 = \{ \langle r_6, q_1 \rangle, \langle r_8, q_2 \rangle, \langle r_2, q_4 \rangle, \langle r_1, q_5 \rangle, \langle r_5, q_7 \rangle, \langle r_4, q_8 \rangle, \langle r_7, q_3 \rangle \} \quad (2.186)$$

Następnie zostaje wyznaczony zbiór:

$$V_n = \bar{G}^i \boxminus \mathcal{G}_p = \{ r_1, r_5 \} \quad (2.187)$$

Zbiór W_4 zadany jest wyrażeniem:

$$W_4 = (W_3 \cup V_n) \setminus \text{crd}^1 \Psi_4 = \{r_3\} \quad (2.188)$$

Ponieważ prawdziwe jest zdanie:

$$\text{crd}^1 \Psi_4 \not\subseteq R^i \quad (2.189)$$

a więc kontynuowany jest proces wyznaczania wstępnego przyporządkowania.

Analiza podobna do rozważań przeprowadzonych na czwartym etapie procesu syntezy jest podstawą do przyporządkowania wierzchołkowi r_3 grafu $\bar{G}^i$ wierzchołka q_6 .

Przyporządkowanie Ψ_5 określone jest następująco:

$$\Psi_5 = \{ \langle r_6, q_1 \rangle, \langle r_8, q_2 \rangle, \langle r_2, q_4 \rangle, \langle r_1, q_5 \rangle, \langle r_5, q_7 \rangle, \langle r_4, q_8 \rangle, \langle r_7, q_3 \rangle, \langle r_3, q_6 \rangle \} \quad (2.190)$$

Zbiór $\text{crd}^1 \Psi_5$ spełnia zależność:

$$\text{crd}^1 \Psi_5 = R^i \quad (2.191)$$

Dla zbioru $Z_S = \{G_1, G_2, G_3\}$ zostaje wyznaczony zbiór:

$$\sigma_S^i = \bigcup_{G_j \in Z_S} \Phi_j \quad (2.192)$$

Na podstawie zbioru σ_S^i zostaje określony zbiór δ_S^i .

Zbiór δ^i zadany jest wyrażeniem:

$$\delta^i = \delta^i \setminus \delta_S^i = \{ \langle r_2, z_3, r_7 \rangle, \langle r_8, z_4, r_1 \rangle, \langle r_3, z_4, r_8 \rangle, \langle r_8, z_3, r_5 \rangle, \langle r_4, z_1, r_3 \rangle \} \quad (2.193)$$

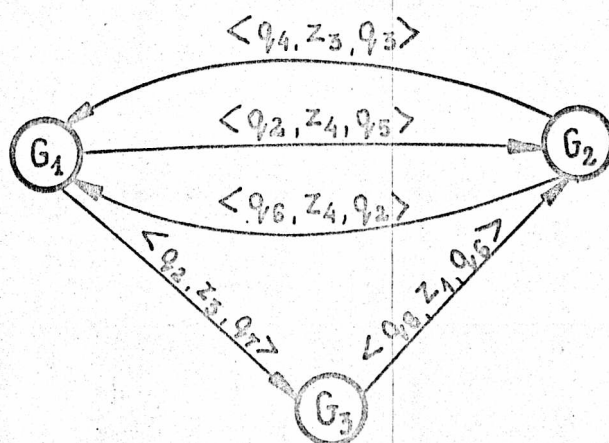
Na podstawie zbiorów δ^i , Ψ_5 zostaje wyznaczony zbiór:

$$S_C = \{ \langle q_4, z_3, q_3 \rangle, \langle q_2, z_4, q_5 \rangle, \langle q_6, z_4, q_2 \rangle, \langle q_2, z_3, q_7 \rangle, \langle q_8, z_1, q_6 \rangle \} \quad (2.194)$$

Graf G^C automatu AZI określony jest zależnością:

$$G^C = \bigoplus_{G_j \in Z_S}^{S_C} G_j \quad (2.195)$$

Postać zagregowaną grafu G^c ilustruje rys. 2.38.



Rys. 2.38. Postać zagregowana grafu G^c

3. REALIZACJA ODWZOROWYWANIA AUTOMATÓW ZAGREGOWANYCH W STRUKTURZE UCZĄCEGO SIĘ AUTOMATU PARAMETRYCZNEGO

Automaty zagregowane odwzorowywane są w uczącym się automacie parametrycznym. W tym rozdziale będzie rozpatrywany sposób odwzorowywania automatów zagregowanych drugiego rodzaju za pomocą uczącego się automatu parametrycznego. W realizacji odwzorowywania automatu zagregowanego drugiego rodzaju należy uwzględnić łuki grafu tego automatu będące wynikiem wykonania operacji $\oplus^{S_j}$ na grafach podautomatów oraz rezultaty nakładania na siebie wierzchołków grafu uzyskanego w wyniku przeprowadzenia wzmiankowanej operacji $\oplus^{S_j}$. W przypadku odwzorowywania automatów zagregowanych pierwszego rodzaju należy rozpatrywać tylko łuki uzyskane w wyniku wykonania operacji $\oplus^{S_j}$ na grafach podautomatów. Stąd problem analizowany w tym punkcie zawiera przypadki występujące w procesie odwzorowywania automatów zagregowanych pierwszego rodzaju.

W początkowej części tego rozdziału rozpatrywany będzie sposób odwzorowywania podautomatów za pomocą automatu AP.

Graf $\bar{G}$ automatu bez wyjścia automatu AP jest multigrafem. Graf ten może być określony w następujący sposób:

$$\bar{G} = \langle B, \Gamma \rangle \quad (3.1)$$

gdzie B - zbiór wierzchołków grafu $\bar{G}$, Γ - zbiór trójek uporządkowanych $\langle b_i, s_p, b_k \rangle$ przyporządkowanych łukom grafu $\bar{G}$. W wymienionej trójce $b_i, b_k \in B$, natomiast s_p jest literą, którą opisany jest dany łuk skierowany z wierzchołka b_i do b_k .

W czasie realizacji przez automat AP zadania $K_j (1 \leq j \leq n)$ zgodnie z przyjętym przed procesem uczenia się wariantem jego rozwiązania, w grafie $\bar{G}$ wyodrębniony zostaje podgraf cząstkowy $\bar{H}^j$. Odpowiednie opisanie grafu $\bar{H}^j$ cechami funkcjonalnymi umożliwia realizację tego zadania. Podgraf cząstkowy $\bar{H}^j$ może być traktowany jako graf automatu $\bar{A}^j$. Automat $\bar{A}^j$ jest przyporządkowany automатовi $\bar{A}^j$ odwzorowywanemu na pomocą automatu AP. Niech symbol $\Gamma' \mid \bar{D}^j$,

gdzie $\bar{B}^j \subset B$, oznacza zbiór tych trójek $\langle b_i, s_p, b_k \rangle \in \bar{\Gamma}^j$, dla których $b_i, b_k \in \bar{B}^j$.

Graf $\bar{H}^j$ ($1 \leq j \leq n$) można określić następująco:

$$\bar{H}^j = \langle \bar{B}^j, \bar{\Gamma}^j \rangle \quad (3.2)$$

gdzie $\bar{B}^j \subset B$ jest zbiorem wierzchołków grafu $\bar{H}^j$, $\bar{\Gamma}^j$ jest zbiorem trójek uporządkowanych $\langle b_i, s_p, b_k \rangle$ przyporządkowanych łukom grafu $\bar{H}^j$. Zbiór $\bar{\Gamma}^j$ spełnia zależność:

$$\bar{\Gamma}^j \subset \bar{\Gamma} \setminus \bar{B}^j \quad (3.3)$$

Opisanie grafu $\bar{H}^j$ odpowiednimi cechami funkcjonalnymi prowadzi do odwzorowania grafu ${}^+G^j$. Graf ${}^+G^j$ jest grafem automatu ${}^+A^j$ odwzorowywanego działaniem AP. Sposób opisanie grafu $\bar{H}^j$, w celu odwzorowania grafu ${}^+G^j$, określony jest za pomocą wyrażeń:

$$z(t) = f[s(t), b(t-1), p_j] \quad (3.4)$$

$$y(t) = h[b(t), p_j] \quad (3.5)$$

Funkcja f jednoznacznie określa sposób opisanie łuków grafu $\bar{H}^j$ literami alfabetu wejściowego automatu ${}^+A^j$, natomiast funkcja h określa sposób opisanie wierzchołków tego grafu literami alfabetu wyjściowego wzmiankowanego automatu.

W grafie ${}^+G^j$ automatu ${}^+A^j$ można wyróżnić podgrafy cząstkowe opisujące podautomaty wchodzące w skład ${}^+A^j$. Podautomat A_i wyrażony jest grafem G_i . W grafie $\bar{H}^j$ można wyodrębnić podgraf cząstkowy H_i , którego opisanie odpowiednimi cechami funkcjonalnymi umożliwia odwzorowanie grafu G_i . Niech symbol $\bar{\Gamma}^j \setminus B_i$, gdzie $B_i \subset \bar{B}^j$, oznacza zbiór tych trójek $\langle b_i, s_k, b_j \rangle \in \bar{\Gamma}^j$, dla których $b_i, b_k \in B_i$. Graf H_i można określić następująco:

$$H_i = \langle B_i, \bar{\Gamma}_i \rangle \quad (3.6)$$

gdzie $B_i \subset \bar{B}^j$ jest zbiorem wierzchołków grafu H_i , $\bar{\Gamma}_i$ jest zbiorem trójek uporządkowanych $\langle b_i, s_p, b_k \rangle$ przyporządkowanych łukom grafu

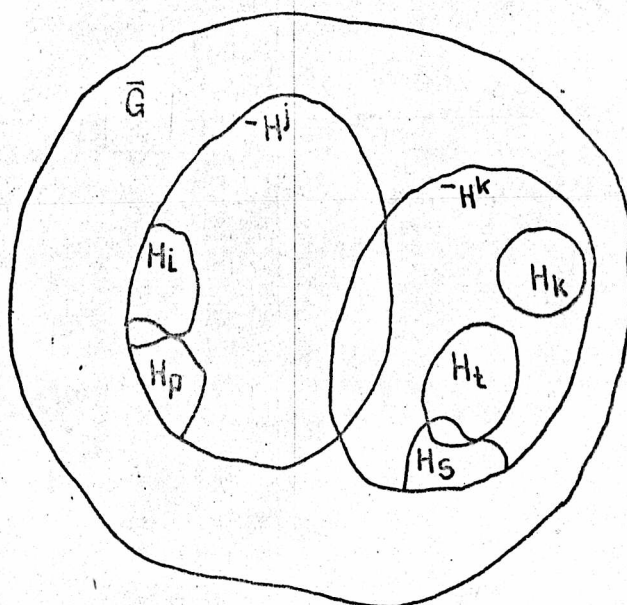
H_i . Dla zbioru Γ_i prawdziwa jest zależność:

$$\Gamma_i \subset \neg \Gamma^j |_{B_i} \quad (3.7)$$

W grafie $\bar{G}$ automatu bez wyjścia automatu AP można wyróżnić zbiór takich podgrafów cząstkowych typu grafu H_i , z których każdy związany jest z pewnym podautomatem A_i ze zbioru L opisanego wyrażeniem (2.4). Zbiór rozważanych podgrafów cząstkowych określony jest w następujący sposób:

$$H^+ = \{H_i\}_{i \in I} \quad (3.8)$$

Szkicowe wyróżnienie grafów $\neg H^j$ i grafów H_i w grafie $\bar{G}$ przedstawia rys. 3.1.



Rys. 3.1. Szkicowe wyróżnienie grafów typu $\neg H^j$ i grafów typu H_i w grafie $\bar{G}$ automatu bez wyjścia automatu AP

Opisanie grafu H_i odpowiednimi cechami funkcjonalnymi prowadzi do odwzorowania grafu G_i podautomatu A_i . Funkcjami opisującymi graf H_i są odpowiednie obcięcia funkcji f, h przedstawionych wyrażeniami (3.4), (3.5). Niech symbol $f|_{H_i}$ będzie symbolem obcięcia funkcji f do grafu H_i . Podobnie niech będzie interpretowany symbol $h|_{H_i}$.

Sposób opisanego grafu H_i , w celu odwzorowania grafu G_i , określony jest za pomocą wyrażeń:

$$z(t) = f|H_i[s(t), b(t-1), p_j] \quad (3.9)$$

$$y(t) = h|H_i[b(t), p_j] \quad (3.10)$$

Wyrażenie (3.9) opisuje obcięcie funkcji przedstawionej za pomocą wyrażenia (3.4) do zbioru łuków grafu H_i . Wyrażenie (3.10) wyznacza obcięcie funkcji opisanej wyrażeniem (3.5) do zbioru wierzchołków grafu H_i .

Kolejna część tego rozdziału będzie dotyczyła odwzorowywania wyników operacji $\oplus$ i Δ , prowadzących do uzyskania automatów zagregowanych drugiego rodzaju z podautomatów ze zbioru L , w uczącym się automacie parametrycznym.

Niech graf G^j automatu zagregowanego drugiego rodzaju określony będzie w następujący sposób:

$$G^j = \begin{matrix} T_j & S_j \\ \Delta & \oplus_{i \in J} \end{matrix} G_i \quad (3.11)$$

Elementami zbioru $\{G_i\}_{i \in J}$ grafów podautomatów poddanych operacjom $\oplus$, Δ niech będą m. in. grafy G_i, G_k . Niech wierzchołek q_1 będzie wierzchołkiem grafu G_i , natomiast wierzchołek q_j - grafu G_k .

Zbiór S_j niech zawiera trójkę uporządkowaną $\langle q_1, z_p, q_j \rangle$. W wyniku wykonania operacji $\oplus$ na zbiorze $\{G_i\}_{i \in J}$ otrzymamy graf:

$$G^k = \begin{matrix} S_j \\ \oplus_{i \in J} \end{matrix} G_i \quad 3.12$$

w którym z wierzchołka q_1 skierowany jest do wierzchołka q_j łuk opisany literą z_p .

Wierzchołki grafów $G_i (i \in J)$ oznaczane są literami q_k , przy czym wierzchołki różnych grafów oznaczone są różnymi literami q_k . Trójki uporządkowane będące elementami zbioru S_j mają postać $\langle q_1, z_p, q_j \rangle$. Symbol q_1 jednoznacznie wyznacza graf, z którego wierzchołek prowadzi

łuk opisany literą z_p . Symbol q_j również jednoznacznie określa ten wierzchołek, do którego prowadzi łuk opisany literą z_p .

Operacji $\oplus_j^{S_j}$ wykonywanej na zbiorze $\{G_i\}_{i \in J}$, w której wyniku zostaje uzyskany graf G^k - wyrażony napisem (3.12), odpowiada operacja oznaczona symbolem $\bar{\oplus}_j^{S_j}$ - przeprowadzona na zbiorze $\{H_i\}_{i \in J}$ dająca w wyniku graf oznaczony symbolem H^k . Odpowiednie opisanie grafu H^k cechami funkcjonalnymi umożliwia odwzorowanie grafu G^k w uczącym się automacie parametrycznym.

Trójkom $\langle q_1, z_p, q_j \rangle \in S_j$ przyporządkowane zostaną trójki tworzące zbiór oznaczony symbolem $\bar{S}_j$, który określa operację $\bar{\oplus}_j^{S_j}$ przeprowadzoną na zbiorze $\{H_i\}_{i \in J}$.

Grafom G_i, G_p przyporządkowane są grafy H_i, H_p . Ponieważ H_i, H_p są podgrafami cząstkowymi grafu $\bar{G}$ automatu bez wyjścia AP, więc może być prawdziwa zależność:

$$B_i \cap B_p \neq \emptyset \quad (3.13)$$

tzn., że pewne wierzchołki grafu H_i mogą być wierzchołkami H_p . Podobnie pewne łuki grafu H_i mogą być łukami grafu H_p .

Niech wierzchołkowi q_1 grafu G_i będzie przyporządkowany wierzchołek b_j grafu H_i oraz wierzchołek b_j niech będzie wierzchołkiem grafu H_p . Przyporządkowanie pierwszej składowej trójki $\langle q_1, z_p, q_j \rangle$ litery b_j powoduje niejednoznaczność, ponieważ nie jest określone z wierzchołka b_j , którego z grafów H_i, H_p skierowany jest do budowywany łuk. Stąd pierwszej składowej rozważanej trójki przyporządkowany zostaje symbol b_j^i , gdzie górny indeks jest indeksem symbolu H_i grafu, z którego skierowany jest wzmiankowany łuk. W podobny sposób określony jest symbol przyporządkowany trzeciej składowej trójki $\langle q_1, z_p, q_j \rangle$. Niech wierzchołkowi q_j grafu G_k przyporządkowany będzie wierzchołek b_c grafu H_k . Stąd trzecia składowa trójki przyporządkowanej trójce $\bar{\oplus}_j^{S_j} \langle q_1, z_p, q_j \rangle$ ma postać b_c^k . Łuk będący wynikiem wykonania operacji $\bar{\oplus}_j^{S_j}$ na grafach $H_i (i \in J)$ i skierowany z wierzchołka b_j grafu

H_i do wierzchołka b_c grafu H_k zostaje opisany literą $\bar{s}_t$. Trójce $\langle q_1, z_p, q_j \rangle$ zostaje wzajemnie jednoznacznie przyporządkowana trójka $\langle b_j^i, \bar{s}_t, b_c^k \rangle \in \bar{S}_j$. Wzmiankowane przyporządkowanie oznaczone jest następująco:

$$\gamma(\langle q_1, z_p, q_j \rangle) = \langle b_j^i, \bar{s}_t, b_c^k \rangle \quad (3.14)$$

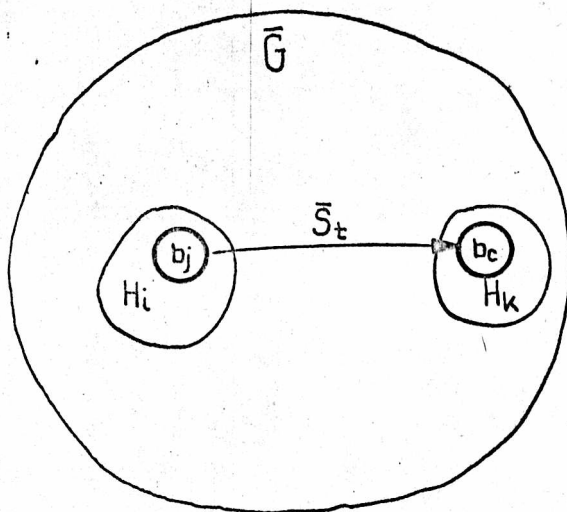
Podobne rozważania przeprowadzone dla każdego elementu zbioru S_j są podstawą do konstrukcji zbioru $\bar{S}_j$ określonego w następujący sposób:

$$\bar{S}_j = \{ \langle b_j^i, \bar{s}_t, b_c^k \rangle : \langle b_j^i, \bar{s}_t, b_c^k \rangle = \gamma(\langle q_1, z_p, q_j \rangle) \wedge \langle q_1, z_p, q_j \rangle \in S_j \} \quad (3.15)$$

Graf H^k określony jest następująco:

$$H^k = \bigoplus_{i \in J}^{\bar{S}_j} H_i \quad (3.16)$$

Łuki wyznaczone przez zbiór $\bar{S}_j$ zostają dobudowane do grafu $\bar{G}$. Na rys. 3.2. przedstawiono łuk wyznaczony przez trójkę $\langle b_j^i, \bar{s}_t, b_c^k \rangle$, dobudowany do grafu $\bar{G}$.



Rys. 3.2. Łuk wyznaczony przez trójkę $\langle b_j^i, \bar{s}_t, b_c^k \rangle$, dobudowany do grafu $\bar{G}$.

Niżej zostanie określona operacja przeprowadzona na grafie H^k , której wynik odpowiada wynikowi operacji Δ_j wykonywanej na grafie G^k . Operacja $\bigoplus_j^{\bar{S}_j}$ wykonywana na zbiorze $\{H_i\}_{i \in J}$ grafów jest podobna do

operacji $\oplus^{S_j}$ przeprowadzonej na zbiorze $\{G_i\}_{i \in J}$. Operacja wykonywana na grafie H^k różni się istotnie od operacji Δ^{T_j} przeprowadzonej na grafie G^k , natomiast operacja ta charakteryzuje się pewnymi cechami wspólnymi z operacją $\oplus^{S_j}$.

Operacja Δ^{T_j} polega na nakładaniu na siebie wierzchołków grafów ze zbioru $\{G_i\}_{i \in J}$. W uczącym się automacie parametrycznym wzmiankowanego nakładania nie można zrealizować. Stwierdzenie to zostanie niżej uzasadnione.

Niech grafy H_i, H_k , będące podstawą do odwzorowania grafów $G_i, G_k \in \{G_i\}_{i \in J}$ będą usytuowane w grafie $\bar{G}$ automatu bez wyjścia automatu AP w sposób przedstawiony na rys. 3.1. Niech jednym z wierzchołków grafu G_i będzie wierzchołek q_a , natomiast jednym z wierzchołków grafu G_k niech będzie q_s oraz niech $q_a, q_s \in \bar{q}^0 \in T_j$. W wyniku wykonania operacji Δ^{T_j} na grafie G^k , wierzchołki q_a, q_s zostaną nałożone na siebie.

Niech wierzchołkowi q_a będzie przyporządkowany wierzchołek b_d grafu H_i , a wierzchołkowi q_s - wierzchołek b_t grafu H_k . Wierzchołków b_d, b_t grafu $\bar{G}$ nie można na siebie nałożyć, ponieważ wierzchołki te są przyporządkowane różnym stanom automatu bez wyjścia automatu AP. Stąd w celu odwzorowania wyniku operacji Δ^{T_j} w uczącym się automacie parametrycznym, należy wykonać pewną operację dotyczącą wierzchołków b_d, b_t - inną niż operacja nakładania wierzchołków na siebie.

Z twierdzenia 2.14. znajdującego się w punkcie 2.3.5. wynika, że dla automatu AZII o grafie G^j istnieje automat AZI o grafie G^p , odwzorowujący zachowanie automatu A^j , tzn. prawdziwe jest wyrażenie:

$$B(G^j) \subset B(G^p) \quad (3.17)$$

Stąd w celu odwzorowania zachowania automatu A^j można zastosować odpowiedni automat AZI. Niech dla automatu A^j dany będzie automat A^p o grafie $G^p = \bigoplus_{i \in J}^{S_p} G_i$. Zbiór S_p jest podstawą do wyznaczenia zbioru oznaczonego symbolem $\bar{S}_p$. Zbiór $\bar{S}_p$ określony jest na bazie

zbioru S_p w taki sposób jak zbiór $\bar{S}_j$ na podstawie zbioru S_j . W wyniku wykonania operacji $\bar{S}_p$ na zbiorze $\{H_i\}_{i \in J}$ grafów zostaje uzyskany graf:

$$H^p = \bigoplus_{i \in J}^{\bar{S}_p} H_i \quad (3.18)$$

Każdy z wierzchołków grafu $\bar{G}$ może w grafie $\bar{H}^j$, przyporządkowanemu grafowi ${}^+G^j$ automatu odwzorowywanego za pomocą AP, występować co najwyżej jeden raz. Wynika to z konieczności odwzorowania w grafie $\bar{G}$ grafu $\bar{H}^j$, który jest podgrafem cząstkowym $\bar{G}$.

Zasady budowy automatów zagregowanych dopuszczają możliwość występowania każdej pary różnych podautomatów A_i, A_j w jednym automacie zagregowanym. Niech pewien wierzchołek b_k grafu $\bar{G}$ będzie wierzchołkiem grafów H_i, H_j . Niech grafy H_i, H_j wejdą w skład grafu H^p przyporządkowanego automatu zagregowanemu A^p . Stąd w grafie H^p co najmniej dwa wierzchołki będą opisane literą b_k . Zatem istnieje możliwość budowy automatów zagregowanych, dla których grafy typu grafu H^p nie są podgrafami cząstkowymi grafu uzyskanego w wyniku uzupełnienia grafu $\bar{G}$ o dodatkowe łuki. W celu dokładnego opisanie wierzchołków grafu H^p należałoby każdy wierzchołek spośród wierzchołków opisanych jedną i tą samą literą b_k opisać innym symbolem. W prowadzonych rozważaniach stosowany jest opis wierzchołków grafów typu grafu H^p literami $b_i \in B$, ponieważ opis ten umożliwia czytelniejsze przedstawienie sposobu odwzorowywania automatów zagregowanych w uczącym się automacie parametrycznym.

Niżej w sposób szkicowy zostanie zilustrowany sposób wykorzystania grafu $\bar{G}$ automatu bez wyjścia automatu AP do budowy grafu H^p - nie będącego podgrafem cząstkowym grafu uzyskanego w wyniku uzupełnienia $\bar{G}$ o dodatkowe łuki.

Niech w grafie $\bar{G}$ będą m. in. podgrafy cząstkowe H_i, H_j, H_k .

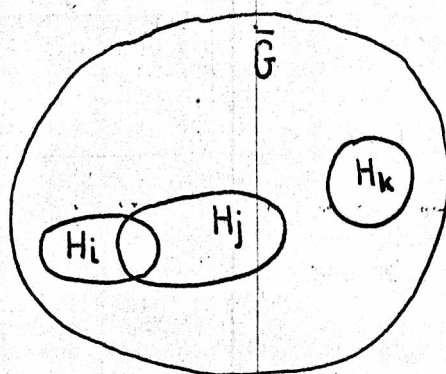
Definicja 3.1.

Grafy H_i, H_j nazywamy częściowo pokrywającymi się, jeśli praw-

dziwe jest zdanie:

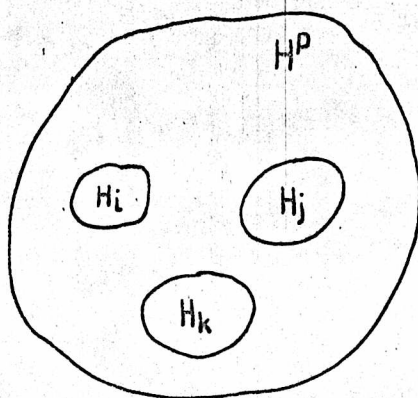
$$B_i \cap B_j \neq \emptyset \quad (3.19)$$

Niech podgrafy cząstkowe H_i , H_j grafu $\bar{G}$ częściowo pokrywają się. Rozważany przypadek ilustruje rys. 3.3.



Rys. 3.3. Szkicowe wyróżnienie podgrafów cząstkowych H_i , H_j , H_k w grafie $\bar{G}$ automatu bez wyjścia automatu AP

Graf H^P może mieć postać, której wyróżnione fragmenty przedstawione są na rys. 3.4.



Rys. 3.4. Szkicowe wyróżnienie podgrafów cząstkowych H_i , H_j , H_k w grafie H^P

Z rysunków 3.3., 3.4. wynika, że w grafie H^P dwukrotnie występuje wspólna część grafów H_i , H_j , a więc graf H^P nie jest podgrafem cząstkowym grafu uzyskanego w wyniku uzupełnienia grafu $\bar{G}$ o dodatkowe łuki. Jeśli grafy H_i , H_j wchodzą w skład dwu różnych grafów typu grafu $\bar{H}^j$, to ich wspólna część może być różnie opisana cechami funkcjonalnymi.

Zatem wspólna część grafów H_i, H_j w grafie H^P może być różnie opisana cechami funkcjonalnymi w zależności od tego, w skład którego z grafów H_i, H_j ona wchodzi.

Sposób opisu grafu H_i cechami funkcjonalnymi w celu odwzorowania grafu G_i określony jest za pomocą wyrażeń (3.9), (3.10). W celu odwzorowania grafu G^P za pomocą grafu H^P , oprócz przyporządkowania cech funkcjonalnych grafom ze zbioru $\{H_i\}_{i \in J}$, należy łuki wyznaczone przez elementy zbioru $\bar{S}_p$ opisać odpowiednimi literami alfabetu Z^P .

Niech trójce uporządkowanej $\langle q_j, z_k, q_c \rangle \in S_p$ będzie wzajemnie jednoznacznie przyporządkowana trójka $\langle b_k^s, \bar{s}_r, b_n^t \rangle \in \bar{S}_p$. W celu odwzorowania grafu G^P za pomocą grafu H^P ten łuk grafu H^P , który wyznaczony jest przez trójkę $\langle b_k^s, \bar{s}_r, b_n^t \rangle$ zostanie opisany literą z_k , będącą drugą współrzędną trójki $\langle q_j, z_k, q_c \rangle$.

Niżej przedstawione rozważania są podstawą do określenia funkcji zwanej funkcją zmiany podautomatów.

Określenia zbiorów $Z_{q_j}^P, Z_{q_j}^{P,-}$ przedstawione są w punkcie 2.3.2. pracy.

Ponieważ automat A^P o grafie G^P jest automatem AZI, a więc zbiory $\{G_i\}_{i \in J}$ oraz S_p spełniają m. in. drugi warunek determinizmu dla operacji $\oplus$. Niech q_s będzie wierzchołkiem grafu G_i . Z powyższego wynika, że:

$$Z_{q_s}^i \cap Z_{q_s}^{i,-} = \emptyset \quad (3.20)$$

Jeśli na wejściu automatu A^P o grafie G^P znajdującego się w stanie $q_s \in Q_i$, gdzie Q_i - zbiór wierzchołków grafu G_i , pojawi się litera $z_k \in Z_{q_s}^i$, to automat A^P osiągnie stan $q_c \in Q_i$. Jeśli w podobnym przypadku na wejściu A^P pojawi się litera $z_t \in Z_{q_s}^{i,-}$ i istnieje trójka

$\langle q_s, z_t, q_r \rangle \in S_p$, to automat A^P osiągnie stan $q_r \in Q_k$.

Dla automatu A^P zostaje wprowadzona tzw. funkcja zmiany podautomatów automatu A^P oznaczona symbolem τ_p . We wzmiarkowanym symbolu dolny indeks jest indeksem symbolu A^P . Funkcja τ_p zostanie wyz-

naczona na podstawie zbioru S_p . Jest to funkcja o trzech argumentach. Jeśli w stanie $q_j \in Q_r$ automatu A^P pojawi się na wejściu A^P litera $z_a \in Z_{q_j}^r$ i $\langle q_j, z_a, q_m \rangle \in S_p$, gdzie $q_m \in Q_k$, to automat A^P osiągnie stan q_m podautomatu A_k . Stan q_j jest stanem podautomatu A_r . W tym przypadku dla argumentów A_r, q_j, z_a wartość funkcji τ_p jest parą uporządkowaną $\langle A_k, q_m \rangle$. Symboliczny zapis rozważanej zmiany podautomatów automatu A^P jest następujący:

$$\langle A_k, q_m \rangle = \tau_p(A_r, q_j, z_a) \quad (3.21)$$

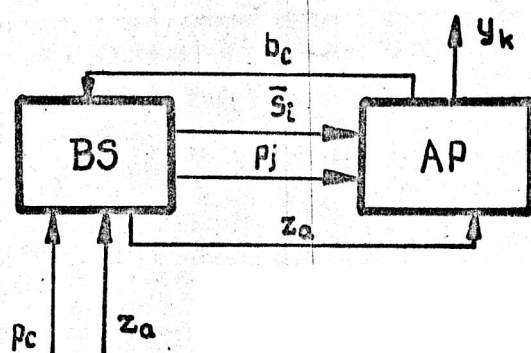
W celu odwzorowania automatu A^P w automacie UAP, należy na podstawie funkcji τ_p określić funkcję oznaczoną symbolem $\bar{\tau}_p$. Niech dana będzie funkcja τ_p określona za pomocą wyrażenia (3.21). Niech podautomat A_r będzie podautomatem automatu ${}^+A^t$ odwzorowywanego działaniem automatu AP, natomiast A_k - automatu ${}^+A^n$. Automatom ${}^+A^t$, ${}^+A^n$ przyporządkowane są wartości p_t , p_n parametru p . Niech stanowi q_j będzie przyporządkowany stan b_c automatu bez wyjścia automatu AP, natomiast stanowi q_m - b_a . W tym przypadku funkcja $\bar{\tau}_p$ określona jest następująco:

$$\langle p_n, b_a \rangle = \bar{\tau}_p(p_t, b_c, z_a) \quad (3.22)$$

Wyrażenie (3.22) interpretowane jest w następujący sposób. Jeśli w stanie b_c automatu bez wyjścia automatu AP dla wartości p_t parametru p , pojawi się na wejściu automatu UAP litera z_a , to nastąpi zmiana wartości parametru p na p_n , oraz zmiana stanu automatu bez wyjścia na stan b_a . Rozważana zmiana podautomatów, związana jest ze zmianą podzadania zadania aktualnie realizowanego przez automat UAP.

Niżej zostanie przedstawiona struktura tej części automatu UAP, która odwzorowuje automaty zagregowane.

Układ odwzorowujący automaty zagregowane ilustruje rys. 3.5.



Rys. 3.5. Schemat blokowy układu odwzorowującego automaty zagregowane

W podrozdziale 1.4. pracy został przedstawiony podział zbioru P^i na dwa rozłączne podzbiory P_1^i , P_2^i . Jeśli na wejściu bloku sterującego (BS) pojawi się wartość $p_c \in P_1^i$ parametru p , to blok BS prześle tę wartość na wejście automatu AP. Litery z_a - kolejno pojawiające się na wejściu bloku BS będą przesyłane na wejście automatu AP. W wyniku powyższego zadanie K_c będzie realizowane przez automat ${}^{+A^c}$ odwzorowywany działaniem automatu AP. Jeśli wartość p_c parametru p podana na wejście bloku BS spełnia zależność $p_c \in P_2^i$, to zadanie K_c będzie realizowane przez automat zagregowany A^p .

W celu odwzorowywania automatu zagregowanego A^p wykorzystywana jest funkcja $\bar{\tau}_p$ zmiany podautomatów dla automatu A^p . Niech w danej chwili będzie realizowane to podzadanie zadania K_c , które realizuje podautomat A_r . Zgodnie z wyrażeniami (3.21), (3.22) wartość p_j podawana na wejście automatu AP spełnia zależność $p_j = p_t$. Jeśli w stanie b_c automatu bez wyjścia automatu AP na wejściu bloku BS pojawi się litera z_a , to zostaną podjęte przez blok BS następujące działania. Jeśli $z_a \in Z_{q_j}^r$, to na wejście automatu AP zostanie podana wartość p_t parametru p oraz litera z_a . Jeśli $z_a \in Z_{q_j}^{r,-}$, to na wejście automatu AP zostanie podana wartość p_n , zgodnie z wyrażeniem (3.22). Poza tym blok BS wygeneruje literę $\bar{s}_1$ powodującą zmianę stanu automatu bez wyjścia automatu AP na stan b_a . W stanie b_a dla wartości p_n automat AP

wygeneruje literę y_k alfabetu wyjściowego określoną następująco:

$$y_k = h(b_a, p_n) \quad (3.23)$$

gdzie h - funkcja określająca działanie bloku wyjściowego automatu AP.

Zmiany wartości parametru p i stanu automatu bez wyjścia automatu AP realizowane są z wykorzystaniem zbioru rejestrów. W rejestrach tych na poszczególnych pozycjach zapisane są następujące informacje:

p_n	b_a	p_t	b_c	z_a
-------	-------	-------	-------	-------

określone wyrażeniem (3.22). Zapis do tych rejestrów dokonywany jest po opanowaniu przez automatu UAP nowego wariantu rozwiązania zadania. Każdemu zadaniu realizowanemu za pomocą jednego z automatów zagregowanych przyporządkowany jest zbiór rejestrów odwzorowujący funkcję zmiany podautomatów dla tego automatu.

ZAKOŃCZENIE

W pracy zaproponowano koncepcję uczenia się automatu deterministycznego. Przedstawiono proces syntezy takiego automatu odwzorowywanego działaniem uczącego się automatu parametrycznego, który zdolny jest do realizacji nowego zadania lub zadania poprzednio realizowanego przez automat UAP ale zgodnie z nowym wariantem jego rozwiązania. Sformułowano również podstawy do syntezy strukturalnej automatu UAP.

Ponieważ praca ta jest pierwszą z zakresu proponowanej koncepcji uczenia się, a więc należy określić problemy dalszych badań.

Analizowana metoda uczenia się polega na opanowywaniu takich wariantów rozwiązania zadań, których graf jest izomorficzny z grafem pewnego automatu zagregowanego, możliwego do uzyskania na bazie zbioru podautomatów odwzorowywanych działaniem automatu AP. Warunek ten powoduje, że klasa wariantów rozwiązania zadań opanowywanych przez automat UAP nie jest odpowiednio duża.

Aby klasa wariantów rozwiązania zadań, do których opanowania zdolny jest automat UAP była większa, należy zrezygnować z tego ograniczenia.

Niżej zostanie przedstawiona propozycja budowy grafu automatu z grafów, które nie muszą wyrażać podautomatów ze zbioru L podautomatów odwzorowywanych działaniem automatu AP .

Niech dane będą:

- 1) graf $\bar{G}^1$ opanowywanego wariantu rozwiązania zadania,
- 2) zbiór $\{G_i\}_{i \in I}$ grafów podautomatów ze zbioru L ,
- 3) liczba rzeczywista t .

Niech symbol F_w oznacza funkcję wielkości grafu, która grafowi przyporządkowuje liczbę rzeczywistą. Przy określaniu wartości tej funkcji mogą być uwzględnione takie wielkości jak np. liczba łuków, liczba wierzchołków grafu.

Oznaczmy symbolem $\mathcal{X}_c$ taki podgraf cząstkowy grafu $\bar{G}^1$, że:

- 1) spełniona jest zależność:

$$F_w(\mathcal{X}_c) \geq t \quad (Z.1)$$

- 2) istnieje izomorficzny z $\mathcal{X}_c$ fragment pewnego grafu ze zbioru $\{G_i\}_{i \in I}$.

Warunek określony wyrażeniem (Z.1) uzasadniony jest tym, że tylko odpowiednio duże fragmenty zadania warto realizować za pomocą urządzenia będącego uczącym się automatem parametrycznym.

Definicja Z.1.

Częściowym pokryciem grafu $\bar{G}^1$ oznaczonym symbolem C_j nazywamy zbiór takich grafów $\mathcal{X}_c$, które nie mają wspólnych wierzchołków.

Niech T będzie zbiorem indeksów symboli częściowych pokryć grafu $\bar{G}^1$.

Należy wyznaczyć częściowe pokrycie C_1 spełniające zależność:

$$F_W(C_i) = \max \{ F_W(C_k) : k \in T \} \quad (Z.2)$$

Stąd zostaje określone pokrycie, które pokrywa największą część grafu $\bar{G}^i$.

Ogólnie pokrycie C_i pokrywa tylko część grafu $\bar{G}^i$. Stąd należy określić sposób realizacji tej części zadania, która nie będzie realizowana przez żaden z podautomatów ze zbioru L .

Elementami zbioru L są podautomaty wchodzące w skład automatów odwzorowywanych za pomocą automatu AP . Automaty te realizują zadania ze zbioru K zgodnie z przyjętymi przed procesem uczenia się wariantami ich rozwiązania. Istotnym problemem jest określenie takiego zbioru podautomatów odwzorowywanych przez automat AP , który umożliwiłby budowę automatów zdolnych do realizacji możliwie wielu zadań.

W celu zwiększenia liczby wariantów rozwiązania zadań, które może opanować uczący się automat parametryczny, należy dla nowych podzadań rozpoznanych w zadaniach określić sposób budowy podautomatów zdolnych do realizacji tych podzadań.

Szczególnie ważnym jest problem budowy grafu wariantu rozwiązania zadania. Ponieważ nie znane są odpowiednio ogólne zasady heurystyki, a więc nie można sformułować algorytmu umożliwiającego syntezę tych grafów wariantów rozwiązania zadań, które potrafi zbudować człowiek. Zwiększenie klasy problemów, które mogą być rozwiązywane za pomocą algorytmu heurystycznego powoduje zmniejszenie efektywności tego algorytmu. Stąd dla interesujących nas klas zadań należy określać algorytmy konstrukcji grafów wariantów ich rozwiązania.

W podrozdziale 1.4. pracy przedstawiono podstawy do budowy algorytmu, wg którego przeprowadzana jest synteza grafu wariantu rozwiązania zadania dla szukania informacji użytkowej rekordu o zadanym kluczu.

Wprowadzenie zasad ograniczających różnorodność grafów wariantów rozwiązania zadań umożliwia zwiększenie liczby tych wariantów,

które może opanować uczący się automat parametryczny. Grafy skończone wyrażające warianty rozwiązania zadań mogą być bardzo zróżnicowane. Stąd należy określić zasady ograniczające różnorodność tych grafów. Przykładem takiej zasady jest budowa sieci działań zgodnie z założeniami programowania strukturalnego. W równoważnych sieciach działań pewne bloki mogą być w różny sposób uporządkowane. W celu zmniejszenia liczby wariantów sieci działań dla danego zadania, należy wprowadzić zasady określające porządkowanie bloków.

Celem odwzorowywania pewnych algorytmów systemu operacyjnego w uczącym się automacie parametrycznym jest zwiększenie efektywności systemu cyfrowego. Stąd należy określić, jakie algorytmy warto odwzorowywać w tym automacie zamiast odwzorowywania ich w oprogramowaniu.

Wymienione wyżej problemy uzasadniają potrzebę dalszych badań w celu zwiększenia możliwości uczącego się automatu parametrycznego w zakresie opanowywania nowych wariantów rozwiązania zadań.

LITERATURA

- [1] Aleksandrov E.A., Osnovy teorji evrističeskich rešenij, Sovetskoe Radio, Moskva 1975
- [2] Apter J.A., Komputery a psychika - Symulacija zachowania, PWN, Warszawa 1973
- [3] Arbib M.A., The metaphorical brain - An introduction to cybernetics as artificial intelligence and brain theory, Wiley - Interscience, New York 1972
- [4] Berlyne E. D., Struktura i kierunek myślenia, PWN, Warszawa 1969
- [5] Blikle A., Automaty i gramatyki, Wstęp do lingwistyki matematycznej, PWN, Warszawa 1971
- [6] Bogomolov, Grunskij, Speranskij, Kontrol i preobrazovanija diskretnych avtomatov, Naukova Dumka, Kiev 1975
- [7] Boute R.J., Distinguishing sets for optimal state identification in checking experiments, IEEE Transactions on Computers, August 1974, 874-877
- [8] Bromirski J., Teoria automatów, WNT, Warszawa 1969
- [9] Cetlin M.L., O povedenji konečnych avtomatov v slučajnych sredach, Avtomatika i telemekhanika, Tom XXII, Nr 10, 1961, 1345-1354
- [10] Chu Y., Computer organization and microprogramming, Prentice-Hall Inc., New York 1972
- [11] Colin A.J.T., Introduction to operating systems, Macdonald: London and American Elsevier Inc., New York 1971
- [12] Cuttle G., Robinson P.B., Executive programs and operating systems, Macdonald: London and American Elsevier Inc., New York 1970
- [13] Donovan J.J., Systems programming, Mc Graw-Hill Book Company, New York 1972

- [14] Eadie D., Nowoczesne maszyny i systemy cyfrowe, WNT, Warszawa 1975
- [15] Feigenbaum E.A., Feldman J., Maszyny matematyczne i myślenie, PWN, Warszawa 1972
- [16] Fryźlewicz Z., Badania symulacyjne algorytmów adresacji w nieuporządkowanych strukturach, Instytut Cybernetyki Technicznej Politechniki Wrocławskiej, Komunikat 293, Wrocław 1975
- [17] Gawroński R., Problemy bioniki w systemach wielkich, WMON, Warszawa 1975
- [18] Gill A., Introduction to the theory of finite-state machines, Mc Graw-Hill Book Company, New York 1962
- [19] Gill A., Single-channel and multichannel finite-state machines, IEEE Transactions on Computers, November 1970, 1073-1078
- [20] Golomb S.W., Baumert L.D., Backtrack programming, Journal of ACM, October 1965, 516-524
- [21] Hennie F.C., Finite state models for logical machines, Wiley, New York 1968
- [22] IBM System/370, Principles of operations, IBM, 1973
- [23] Katzan H., Operating systems, Van Nostrand Reinhold Company, New York ... 1973
- [24] Kazimierczak J., Automaty rozgrywające parametryczne - Synteza i zastosowanie, Monografie, Politechnika Wrocławska, Wrocław 1974
- [25] Kazimierczak J., Concept of the parametric automaton and its synthesis, Instytut Cybernetyki Technicznej Politechniki Wrocławskiej, Komunikat 297, Wrocław 1975
- [26] Kazimierczak J., Koncepcja systemu operacyjnego SC działającego na bazie automatu parametrycznego, Prace VII Krajowej Konferencji Automatyki, Rzeszów 1977
- [27] Kazimierczak J., Ogólna koncepcja i model abstrakcyjny automatu

- skończonego parametrycznego, Instytut Cybernetyki Technicznej Politechniki Wrocławskiej, Komunikat 111, Wrocław 1974
- [28] Kempisty M., Pamięć skojarzeniowa - Model cybernetyczny, PWN, Warszawa 1968
 - [29] Kozub V.M., Sistemy prerywanija CWM, Sovetskoje Radio, Moskva 1976
 - [30] Kulikowski J.L., Cybernetyczne układy rozpoznające, PWN, Warszawa 1972
 - [31] Kuratowski K., Wstęp do teorii mnogości i topologii, PWN, Warszawa 1972
 - [32] Magott J., Automat parametryczny jako uczący się organ sterujący systemu sterowania, Prace VII Krajowej Konferencji Automatyki, Rzeszów 1977
 - [33] Magott J., Automatowy model uczenia się i jego zastosowanie w procesie uczenia się systemu cyfrowego, Instytut Cybernetyki Technicznej Politechniki Wrocławskiej, Komunikat 449, Wrocław 1976
 - [34] Magott J., Learning process in cybernetical system with application of parametric automaton, Systems Science IV, International Conference on Systems Science, Abstracts of papers, Wrocław 1977
 - [35] Melichov A.N., Orentirovannyje grafy i konečnyje avtomaty, Nauka, Moskva 1971
 - [36] Mielcarek J., Primenenie modelirujuščich programm dlja ocenki nekotorygo sposoba opredelenija prioritetov v vyčislitel'nych sistemach, Struktura i organizacija paketov programm, Tezisy dokladov, Vyčislitel'nyj Centr AN Gruzinskoj SSR Tbilisi, Mecniereva 1976
 - [37] Modelirowanije obučenija i povedenija, Nauka, Moskva 1975
 - [38] Neumann J., von, Obščaja i logičeskaja teorija avtomatov, Fizmatgiz, Moskva 1960

- [39] Neumann J., von, Vychislitel naja masina i mozg, Kibernetičeskij sbornik, Nr 1, Moskva 1960
- [40] Operating systems techniques, Ed. by C.A.R. Hoare, Academic Press, London 1972
- [41] Paz A., Introduction to probabilistic automata, Academic Press, New York and London, 1971
- [42] Problemy psychologii matematycznej, Pod redakcją Józefa Kozieleckiego, PWN, Warszawa, 1971
- [43] Puszkin W.N., Psychologia i cybernetyka, WSiP, Warszawa 1976
- [44] Rasiowa H., Wstęp do matematyki współczesnej, PWN, Warszawa 1973
- [45] Slagle J.R., Artificial intelligence: The heuristic programming, Mc Graw-Hill Book Company, New York 1971
- [46] Smith W.R., Rice R., Chesley G., ..., SYMBOL - A large experimental system exploring major hardware replacement of software, Spring Joint Computer Conference, 1971
- [47] Stewart D.J., Automaton theory and learning systems, Academic Press, London and New York 1967
- [48] Tichomirow O.K., Struktura czynności myślenia człowieka, PWN, Warszawa 1976
- [49] Tou J.T., Advances in information systems science, Plenum Press, New York 1969
- [50] Trachtenbrot V.A., Barzdin J.M., Konečnyje avtomaty, Nauka, Moskva 1970
- [51] Watanabe S., Creative learning and propensity automaton, IEEE Transactions on Systems, Man and Cybernetics, November, 1975, 603-609
- [52] Wee W.C., Fu K.S., A formulation of fuzzy automata and its application as a model of learning systems, IEEE Transactions on Systems Science and Cybernetics, July 1969.

Niniejszy komunikat otrzymują:

1. OINT	1 egz.
2. Biblioteka Główna Politechniki Wrocławskiej	1 egz.
3. Biblioteka Międzyinstytutowa I-6	1 egz.
4. I-szy Z-ca Dyrektora I-6	1 egz.
5. Z-ca Dyrektora I-6 d/s Badań Naukowych i Współpracy z Przemysłem	1 egz.
6. Prof. dr hab. J. Jaroń	1 egz.
7. Prof. dr hab. J.L. Kulikowski	1 egz.
8. Doc. dr hab. J. Kazimierzczak	1 egz.
9. Seminarium Systemów Informatyki I-6	1 egz.
10. Seminarium Ogólnej Teorii Systemów I-6	1 egz.
11. Seminarium Systemów Sterowania I-6	1 egz.
12. Seminarium Metodologii Projektowania I-6	1 egz.
13. Instytut Maszyn Matematycznych Politechniki Warszawskiej	1 egz.
14. Instytut Podstawowych Problemów Informatyki PAN - Warszawa	1 egz.
15. Instytut Maszyn Matematycznych - Katowice	1 egz.
16. Instytut Automatyki i Informatyki AGH - Kraków	1 egz.
17. Instytut Automatyki Politechniki Poznańskiej	1 egz.
18. Instytut Informatyki Politechniki Gdańskiej	1 egz.
19. Komitet Informatyki PAN - Warszawa	1 egz.
20. Instytut Informatyki Uniwersytetu Jagiellońskiego	1 egz.
21. Zespół 10502	1 egz.
22. Autor	4 egz.

R A Z E M:

25 egz.

		1 2 3 4 5 6 7 8	
* N *		N.N.P.V.I.T.A.N	
Rozpocz. pr.		Zakończ. pr.	
Opust. pr.		Instytut	
01.7.6		1.1.7.7	
Nr zlecenia		Nr archiwizacji	
		1.06.1.K-56.51.7.7*	
Symbol UKD		77:Inst.Cybern.	
681.322.06		PWr	
Systemy cyfrowe		MNSZWLT	
		pol.	
Opis bibliograficzny			
MAGOTT Jan			
Model formalny uczącego się automatu parametrycznego dla potrzeb syntezy systemu operacyjnego SC.			
Komunikaty Inst.Cybern. PWr 1977 nr 565			
440s., 50 rys., 52 poz. bibliogr.			
Rozprawa doktorska			
Instytut Cybernetyki Technicznej			
Politechniki Wrocławskiej			
Promotor: doc.dr hab.inż. Jan Kazimierzak			
Charakter pracy podstawowa		Rozpowaszcznienie	
Materiały adaptacyjne A			

Analiza dokumentacyjna

<D

W pracy zaproponowano koncepcję uczącego się automatu parametrycznego zdolnego do opanowywania nowych zadań i nowych wariantów rozwiązania tych zadań, do których realizacji automat ten zdolny był przed procesem uczenia się. W celu realizacji nowych zadań lub zadań poprzednio realizowalnych zgodnie z nowymi wariantami ich rozwiązania, rozważany automat przeprowadza syntezę automatów nazwanych automatami zagregowanymi. Uczący się automat parametryczny odwzorowuje swym działaniem otrzymane w procesie uczenia się automaty zagregowane. W pracy przedstawiono podstawy formalne do syntezy strukturalnej tego typu automatu. Zaproponowany automat może wchodzić w skład systemu operacyjnego SC pracującego na zasadzie działania automatu parametrycznego będącego wielozadaniowym systemem sterującym. Wyszczególniono wybrane dziedziny funkcjonowania systemu operacyjnego, w których zakresie problem uczenia się odgrywa istotną rolę.

Imię i Nazwisko autora analizy

Jan Magott



Słowa kluczowe

<S

Automat parametryczny, automat uczący się, system operacyjny, teoria grafów, automat zagregowany

**0480 * 00* _____

<A _____ *B _____ *C _____ *D _____

*E _____ *F _____ *G _____ *H _____

Tylko PRL	CINTE	APW	Podpis ed.	Podpis asyst. d/s badań	Potwierdzenie przyjęcia poprawki.	Potwierdzenie przyjęcia karty w Oddziale Dokumentacji.
NIE	TAK	TAK				
Wpisać TAK lub NIE						