Opracował: dr inż. Zbigniew Buchalski

KATEDRA INFORMATYKI TECHNICZNEJ

Ćwiczenia laboratoryjne z Logiki Układów Cyfrowych

ćwiczenie 205

Temat: Zastosowanie wyrażeń regularnych do syntezy i analizy automatów skończonych

1. Cel ćwiczenia

Celem ćwiczenia jest nabycie praktycznej umiejętności projektowania i technicznej realizacji automatów przy zastosowaniu języka wyrażeń regularnych.

2. Program ćwiczenia

1. Na podstawie wyrażenia regularnego opisującego automat określić graf przejść pomiędzy stanami automatu.
2. Przeprowadzić syntezę automatu realizowanego jako automat Moore'a.
3. Realizacja techniczna automatu z zastosowaniem zestawów UNILOG.
4. Sprawdzenie poprawności działania modelu automatu.

3. Problematyka ćwiczenia

Analizę abstrakcyjną automatów przeprowadza się różnymi metodami. Przy prostych automatach proces ten polega na intuicyjnym opisie zachowania automatu na podstawie jego modelu abstrakcyjnego. Zastosowanie języka wyrażeń regularnych do syntezy i analizy automatów skończonych umożliwia przeprowadzenie tego procesu w sposób formalny.

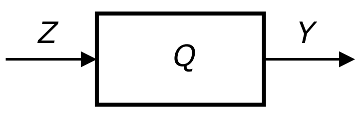
Język wyrażeń regularnych powstał w wyniku poszukiwania prostszych i bardziej funkcjonalnych od tradycyjnych sposobów opisu działania automatów.

4. Wiadomości podstawowe

Definicja 1

Automat skończony jest modelem matematycznym systemu dyskretnego działającego w dyskretnych chwilach czasu. Jego działanie jest określone na zbiorach skończonych sygnałów wejściowych, stanów wewnętrznych i sygnałów wyjściowych.

Automat skończony można zrealizować sprzętowo lub programowo. Ogólnie, schemat blokowy automatu skończonego można przedstawić następująco:



Rys. 1. Schemat blokowy automatu

gdzie:

Z – alfabet wejściowy,
Q – zbiór stanów wewnętrznych,
Y – alfabet wyjściowy,

Automat ten akceptuje słowa należące do języka regularnego. Język regularny jako zbiór słów reprezentowany jest przez wyrażenia regularne.

Załóżmy, że alfabet wejściowy automatu jest następującym zbiorem:

$$Z = \{ z_1, z_2, \dots , z_i, \dots , z_n \}$$

Z symboli tego zbioru możemy zbudować określone słowa, np.:

$$z_1\, z_2\, z_2\, z_1 \qquad z_2\, z_1\, z_3\, z_9 \dots , \quad z_1\, z_1\, z_5 \dots$$

Zbiór wszystkich możliwych słów jest zbiorem nieskończonym Z^*:

$$Z^* = \{ z_1\, z_1\, z_2, \dots , z_2\, z_1\, z_2, \dots , z_9\, z_{10}, z_{12}, \dots \}$$

Na zbiorze Z^* można określić rodzinę zbiorów S^*:

$$S^* = \{ S_1, S_2, \dots , S_i, \dots , S_n \}$$

Na słowach $S_i \in S^*$ jak również na słowach przynależnych do dowolnego zbioru $S_i \in S^*$ przeprowadzane są określone operacje. Dowolny zbiór $S_i \in S^*$ zawierający słowa wejściowe automatu nazywamy zdarzeniem.

<u>Definicja 2</u>

Do oznaczenia zbiorów powstałych w wyniku wykonania operacji sumy, konkatenacji i iteracji posługujemy się wyrażeniem nazywanym wyrażeniem regularnym. W skład wyrażenia regularnego wchodzą określone słowa połączone znakami reprezentującymi powyższe operacje.

Każde wyrażenie regularne reprezentuje sobą język regularny.

<u>Twierdzenie 1</u>

Jeżeli r jest wyrażeniem regularnym, to istnieje automat NFA with ε-moves, który akceptuje słowa języka regularnego reprezentowanego przez to wyrażenie.

Mając wyrażenie regularne możemy wykonywać następując transformacje:

- wyrażenie regularne $\rightarrow$ zbiór słów.
- wyrażenie regularne $\rightarrow$ graf przejść automatu akceptującego język reprezentowany przez to wyrażenie: $r \rightarrow S(r)$.
- wyrażenie regularne $\rightarrow$ gramatyka bezkontekstowa regularna, generująca słowa danego języka $S(r)$.

Synteza abstrakcyjna automatów skończonych

Definicja 3

Synteza abstrakcyjna automatu to określenie takiego opisu formalnego automatu, na podstawie którego można zbudować tabele przejść i wyjść automatu. Synteza ta sprowadza się do przejścia od algorytmu działania automatu do grafu przejść automatu.

Poszczególne etapy tej syntezy to:

1. algorytm słowny
2. przedstawienie algorytmu słownego w postaci wyrażeń regularnych
3. określenie grafu przejść

Przykład 1

$$S_1 = z_1 z_2 + z_1 z_1 z_1 \quad | \, y_1$$
$$S_2 = z_1 z_2 z_2 + z_2 z_2 \quad | \, y_2$$
$$S_3 = \overline{S_1 + S_2} \qquad | \, y_0 = \varepsilon$$

gdzie:

S_1 – zdarzenie warunkujące pojawienie się na wyjściu automatu y_1
S_2 – zdarzenie warunkujące pojawienie się na wyjściu automatu y_2
ε – sygnał pusty

W celu określenia stanów automatu wprowadza się pojęcie "miejsca" w wyrażeniu regularnym. Miejscem jest położenie pomiędzy literami, między literą i znakiem dysjunkcji (OR) oraz początek i koniec wyrażenia. Miejscom tym przyporządkowuje się stany automatu.

$$S_1 = \begin{array}{ccccccccccccc} | & z_1 & | & z_2 & | & + & | & z_1 & | & z_1 & | & z_1 & | \\ 0 & & 1 & & 2 & & 0 & & 1 & & 3 & & 4 \end{array}$$

$$S_2 = \begin{array}{ccccccccccccc} | & z_1 & | & z_2 & | & z_2 & | & + & | & z_2 & | & z_2 & | \\ 0 & & 1 & & 2 & & 5 & & 0 & & 6 & & 7 \end{array}$$

Dla uproszczenia rozpatrujemy automat Moore'a. Aby określić na przykład przejście ze stanu 1 pod wpływem sygnału z_2 należy odnaleźć miejsce oznaczone numerem 1, na prawo od którego stoi symbol z_2, numer 2 z prawej strony symbolu z_2 pokazuje stan następny.

Tabela 1

wyjście	y_0	y_0	$\mathbf{y_1}$	y_0	$\mathbf{y_1}$	$\mathbf{y_2}$	y_0	$\mathbf{y_2}$	y_0
stany / wejście	0	1	2	3	4	5	6	7	*
z_1	1	3	*	4	*	*	*	*	*
z_2	6	2	5	*	*	*	7	*	*

$5 \equiv 7$

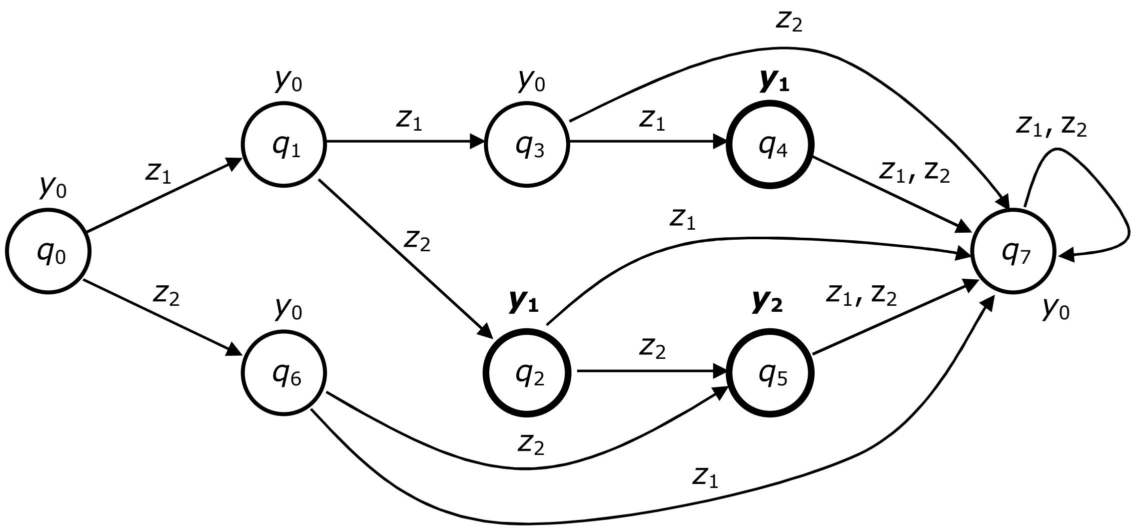
* - do powyższego zapisu wprowadzamy stan dodatkowy, do którego automat przechodzi, gdy pojawi się słowo należące do S_3

Po otrzymaniu tabeli 1, przeprowadzamy minimalizację stanów równoważnych 5 ≡ 7, tworząc tabelę 2 (stany 0 ... 6 oznaczamy jako q_0 ... q_6 a stan * jako q_7):

Tabela 2

wyjście	y_0	y_0	$\mathbf{y_1}$	y_0	$\mathbf{y_1}$	$\mathbf{y_2}$	y_0	y_0
wejście \ stany	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7
z_1	q_1	q_3	q_7	q_4	q_7	q_7	q_7	q_7
z_2	q_6	q_2	q_5	q_7	q_7	q_7	q_5	q_7

Na podstawie tabeli 2 można narysować graf automatu przedstawiony na rys. 2.



Rys. 2. Graf automatu Moore'a z przykładu 1

Wyznaczenie stanów automatów stają się bardziej złożone, gdy w wyrażeniu regularnym występuje operacja iteracji. W tym przypadku wyrażenie regularne dzieli się na miejsca "podstawowe" i "przedpodstawowe".

Miejscami "podstawowymi" nazywamy te miejsca w wyrażeniu regularnym, na lewo od których stoi litera oraz miejsce początkowe.

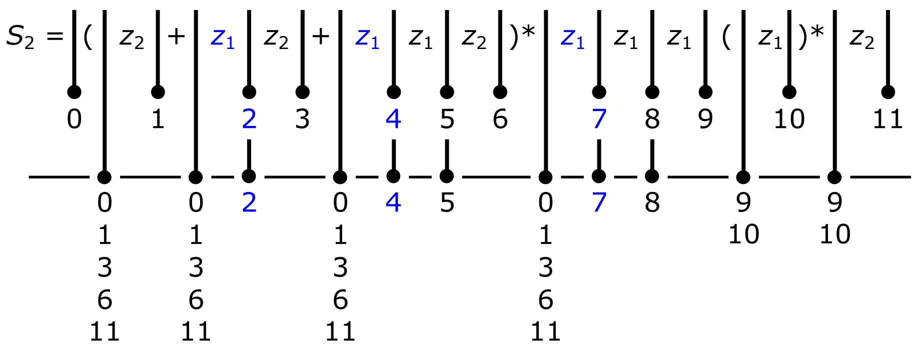
Miejscami "przedpodstawowymi" nazywamy te miejsca w wyrażeniu regularnym, na prawo od których stoi litera.

Przykład 2

$S_1 = (z_2 + z_1 z_2 + z_1 z_1 z_2)^* z_1 z_1 z_1 \quad | y_1$

$S_2 = (z_2 + z_1 z_2 + z_1 z_1 z_2)^* z_1 z_1 z_1 (z_1)^* z_2 \quad | y_2$

$S_3 = \overline{S_1 + S_2} \quad | y_0 = \varepsilon$



Miejsca "przedpodstawowe" oznacza się odpowiednimi symbolami miejsc "podstawowych". Stosujemy przy tym następujące reguły:

Reguła 1

Symbole miejsca "podstawowego" przed nawiasem iteracyjnym rozmieszcza się w miejscach "przedpodstawowych" we wszystkich miejscach początkowych wszystkich członów dysjunktywnych stojących w danym nawiasie.

Reguła 2

Symbol miejsca końcowego dowolnego członu dysjunktywnego zamkniętego w nawiasy iteracyjne rozmieszczamy w miejscach początkowych ("przedpodstawowych") wszystkich członów dysjunktywnych zamkniętych w danym nawiasie.

Reguła 3

Symbole miejsc "podstawowych", na lewo i prawo od których stoją litery nie rozmieszcza się nigdzie więcej.

Reguła 4

Symbol miejsca końcowego wyrażenia rozmieszcza się we wszystkich tych miejscach "przedpodstawowych", gdzie znajduje się symbol miejsca początkowego.

Reguła 5

Symbol miejsca końcowego dowolnego członu dysjunktywnego zamkniętego w nawiasy iteracyjne rozmieszcza się w miejscu "przedpodstawowym" bezpośrednio za danym nawiasem.

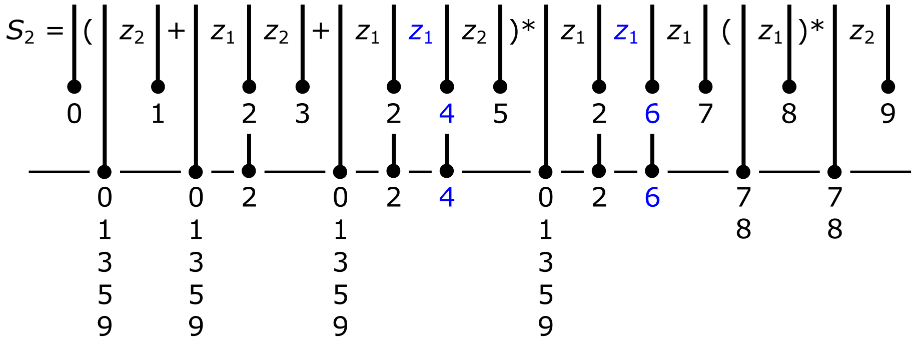
Reguła 6

Symbol miejsca przed nawiasem iteracyjnym zapisuje się w miejscu "przedpodstawowym" znajdującym się za tym nawiasem.

Następnie przeprowadza się minimalizację stanów. Jeżeli kilka miejsc "przedpodstawowych" oznakowane jest jednakowym zbiorem symboli i na prawo od tych miejsc zapisane są takie same litery, to wówczas miejsca "podstawowe" położone na prawo od tych liter są sobie równoważne.

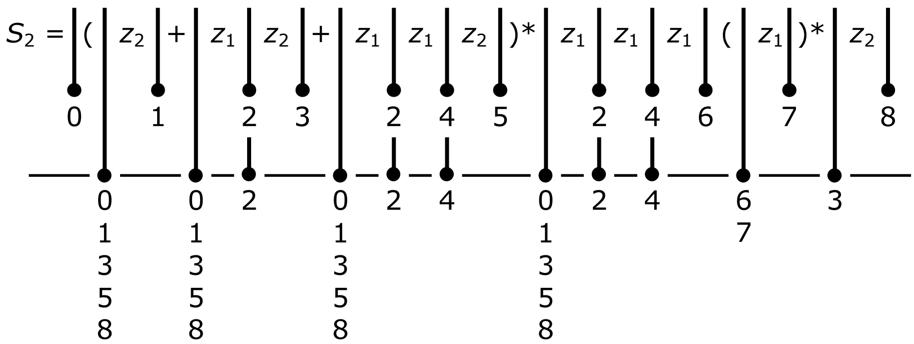
Wracając do przykładu 2 można zauważyć, że stany $2 \equiv 4 \equiv 7$ są sobie równoważne. Wynika to z tego, że w wyrażeniu są trzy takie miejsca, w których z lewej strony litery z_1 jest stan oznaczony takim samym zbiorem symboli $\{\,0, 1, 3, 6, 11\,\}$ a z prawej symbole oznaczające stany 2, 4 i 7. Gdyby stany te nie były równoważne, prowadziłoby to do niedeterminizmu (przejścia do trzech różnych stanów pod wpływem tej samej litery z_1). Łączymy te trzy stany w jeden ($4, 7 \to 2$) i dokonujemy zmiany numerów stanów o numerach powyżej 4:

$5 \to 4,\quad 6 \to 5,\quad 8 \to 6,\quad 9 \to 7,\quad 10 \to 8,\quad 11 \to 9$



W następnej fazie ujawniają się stany równoważne $4 \equiv 6$. Łączymy je w jeden stan ($6 \to 4$) i zmieniamy numery stanów powyżej 6:

$7 \to 6,\quad 8 \to 7,\quad 9 \to 8$



Na podstawie końcowego przyporządkowania numerów stanów tworzymy tabelę przejść:

Tabela 3

wyjście	y_0	y_0	y_0	y_0	y_0	y_0	$\boldsymbol{y_1}$	y_0	$\boldsymbol{y_2}$
stany / wejście	0	1	2	3	4	5	6	7	8
z_1	2	2	4	2	6	2	7	7	2
z_2	1	1	3	1	5	1	8	8	1

$0 \equiv 1 \equiv 3 \equiv 5$

W tabeli 3 można zauważyć stany równoważne $0 \equiv 1 \equiv 3 \equiv 5$. Łączymy je w jeden stan $(1, 3, 5 \to 0)$ i zmieniamy numery stanów powyżej 1:

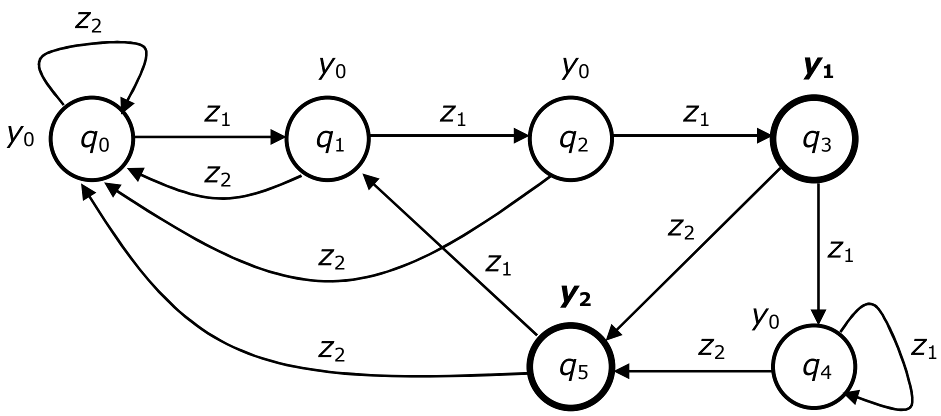
$2 \to 1, \quad 4 \to 2, \quad 6 \to 3, \quad 7 \to 4, \quad 8 \to 5$

Po przyjęciu oznaczeń $q_0 \ldots q_5$ dla stanów $0 \ldots 5$ otrzymujemy przedstawioną poniżej, końcową postać tabeli przejść automatu:

Tabela 4

wyjście	y_0	y_0	y_0	$\boldsymbol{y_1}$	y_0	$\boldsymbol{y_2}$
stany / wejście	q_0	q_1	q_2	q_3	q_4	q_5
z_1	q_1	q_2	q_3	q_4	q_4	q_1
z_2	q_0	q_0	q_0	q_5	q_5	q_0

Graf automatu uzyskany na podstawie tabeli 4 przedstawiony jest na rys. 3.

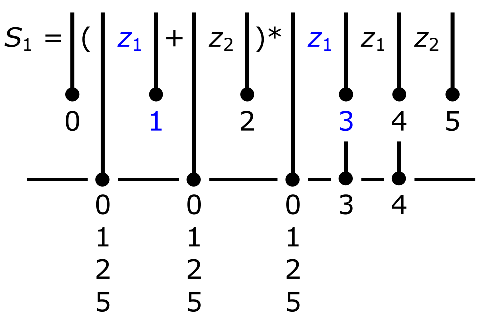


Rys. 3. Graf automatu Moore'a z przykładu 2

Przykład 3

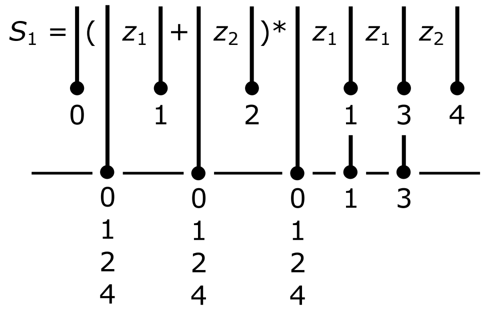
$S_1 = (z_1 + z_2)^* \, z_1 z_1 z_2 \quad | \, y_1$

$S_2 = \overline{S_1} \qquad | \, y_0 = \varepsilon$



Można zauważyć, że stany $1 \equiv 3$ są równoważne. Łączymy je w jeden stan $(3 \to 1)$ i zmieniamy numery stanów powyżej 3:

$4 \to 3, \quad 5 \to 4$



Przy tworzeniu tabeli przejść występuje taka sytuacja, że istnieją dwa różne przejścia ze stanu 1 dla z_1 (do stanów 1 lub 3). W takim przypadku wprowadzamy specjalny stan, oznaczony jako $1{\vee}3$. Zapisując przejścia z takiego stanu musimy uwzględnić przejścia z obu stanów składowych 1 i 3. Może to spowodować pojawienie się kolejnych, nowych stanów:

1 :	$z_1 \to 1{\vee}3$	$z_2 \to 2$
3 :	$z_1 \to *$	$z_2 \to 4$
$1{\vee}3$:	$z_1 \to 1{\vee}3{\vee}*$	$z_2 \to 2{\vee}4$

W tym przypadku otrzymamy nowe stany $1{\vee}3{\vee}*$ i $2{\vee}4$. Dalej postępujemy podobnie, dopóki będą pojawiać się nowe stany.

Przy łączeniu stanów może wystąpić przypadek różnych wyjść (na przykład dla stanu $2{\vee}4$ mamy: $2 \leftrightarrow y_0$, $4 \leftrightarrow y_1$). Wówczas przyjmujemy wyjście odpowiadające akceptacji wyrażenia regularnego, dla stanu $2{\vee}4$ będzie to y_1.

Tabela 5

wyjście	y_0	y_0	y_0	y_0	$\mathbf{y_1}$	y_0	y_0	y_0	$\mathbf{y_1}$	$\mathbf{y_1}$	y_0	y_0
stany / wejście	0	1	2	3	4	*	$1{\vee}3$	$1{\vee}3{\vee}*$	$2{\vee}4$	$2{\vee}4{\vee}*$	$1{\vee}*$	$2{\vee}*$
z_1	1	$1{\vee}3$	1	*	1	*	$1{\vee}3{\vee}*$	$1{\vee}3{\vee}*$	1	$1{\vee}*$	$1{\vee}3{\vee}*$	$1{\vee}*$
z_2	2	2	2	4	2	*	$2{\vee}4$	$2{\vee}4{\vee}*$	2	$2{\vee}*$	$2{\vee}*$	$2{\vee}*$

$0 \equiv 2$ $\quad$ $4 \equiv 2{\vee}4$

W tabeli 3 widać stany równoważne $0 \equiv 2$, łączymy je w jeden stan $(2 \to 0)$. Podobnie dla drugiej pary stanów równoważnych $4 \equiv 2{\vee}4$ $(2{\vee}4 \to 4)$. Po połączeniu stanów równoważnych wykonujemy pokazaną poniżej zmianę oznaczeń stanów:

$3 \to 2, \quad 4 \to 3, \quad * \to 4, \quad 1{\vee}3 \to 5,$

$1{\vee}3{\vee}* \to 6, \quad 2{\vee}4{\vee}* \to 7, \quad 1{\vee}* \to 8, \quad 2{\vee}* \to 9$

po której otrzymujemy następującą postać tabeli przejść automatu:

Tabela 6

wyjście	y_0	y_0	y_0	$\boldsymbol{y_1}$	y_0	y_0	y_0	$\boldsymbol{y_1}$	y_0	y_0
stany / wejście	0	1	2	3	4	5	6	7	8	9
z_1	1	5	4	1	4	6	6	8	6	8
z_2	0	0	3	0	4	3	7	9	9	9

Stany 2 i 4 w tabeli 6 zostają wykreślone, ponieważ nie mogą być nigdy osiągnięte. Po zmianie numeracji:

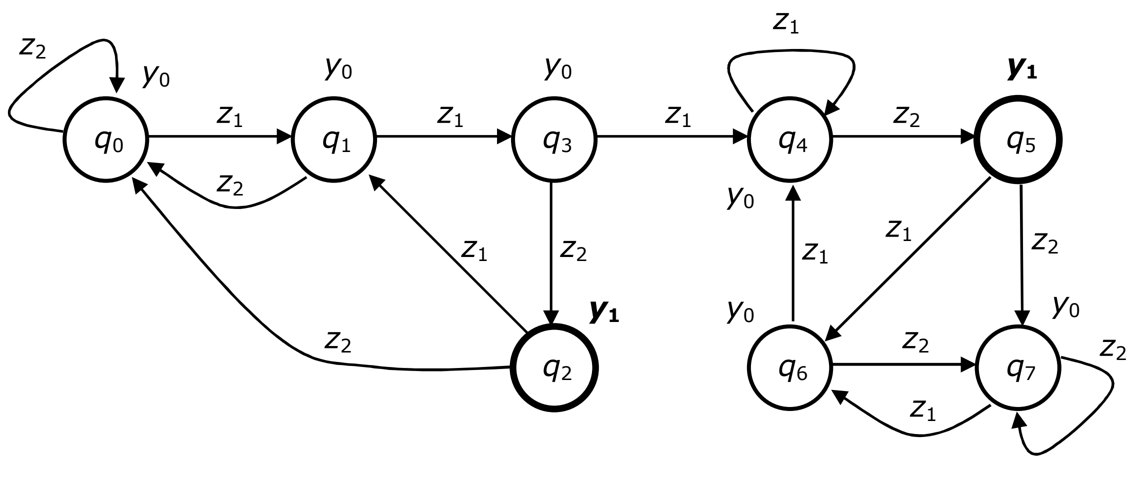
$3 \rightarrow 2, \quad 5 \rightarrow 3, \quad 6 \rightarrow 4, \quad 7 \rightarrow 5, \quad 8 \rightarrow 6, \quad 9 \rightarrow 7$

i wprowadzeniu oznaczeń stanów w postaci q_i otrzymujemy:

Tabela 7

wyjście	y_0	y_0	$\boldsymbol{y_1}$	y_0	y_0	$\boldsymbol{y_1}$	y_0	y_0
stany / wejście	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7
z_1	q_1	q_3	q_1	q_4	q_4	q_6	q_4	q_6
z_2	q_0	q_0	q_0	q_2	q_5	q_7	q_7	q_7

Graf automatu uzyskany z tabeli 7 przedstawiony jest na rys. 4.



Rys. 4. Graf automatu Moore'a z przykładu 3

W pokazanych wcześniej przykładach wykonywana była minimalizacja automatów oparta na pojęciu stanów równoważnych. Dwa stany automatu Moore'a mające przypisane to samo wyjście i jednakowe przejścia dla wszystkich sygnałów wejściowych uznawane były za równoważne i łączone w jeden stan. Metoda ta nie gwarantuje jednak wykrycia wszystkich stanów nadmiarowych. Poniżej pokazana zostanie inna metoda minimalizacji, wykorzystująca pojecie stanów zgodnych, mogąca dać lepsze rezultaty.

Dwa stany wewnętrzne q_i i q_j są zgodne, jeżeli dla każdego sygnału wejściowego z_k uzyskujemy niesprzeczne wyjścia, a ich stany następne są takie same lub niesprzeczne.

Stany q_i i q_j są sprzeczne, jeżeli dla pewnego $z_k \in Z$ otrzymujemy sprzeczne wyjścia. Stany które nie są sprzeczne są niesprzeczne.

W pierwszym kroku prezentowanego algorytmu minimalizacji stany wewnętrzne dzielimy na podzbiory według przypisanych im symboli wyjściowych y_j. Dla automatu z przykładu 3 otrzymujemy dwa podzbiory:

$A = \{ q_0, q_1, q_3, q_4, q_6, q_7 \}$ dla y_0
$B = \{ q_2, q_5 \}$ dla y_1

Na tej podstawie tworzymy nową tabelę przejść zgodnie z następującymi zasadami:

- stany q_i grupujemy według symboli podzbiorów (A lub B) do których należą,
- dla każdego stanu q_i zamiast symbolu stanu następnego q_j wpisujemy symbol podzbioru (A lub B) do którego należy q_j.

W wyniku uzyskujemy następującą, zmodyfikowaną tabelę przejść:

Tabela 8

	A						B	
stany / wejście	q_0	q_1	q_3	q_4	q_6	q_7	q_2	q_5
z_1	A	A	A	A	A	A	A	A
z_2	A	A	B	B	A	A	A	A

Można zauważyć, że stanom q_0, q_1, q_6 i q_7 przypisano jednakowe kombinacje symboli podzbiorów AA (odpowiednio dla z_1 i z_2), natomiast stanom q_3 i q_4 przypisano kombinację AB. Na podstawie tych kombinacji podzbiór A zostanie podzielony na kolejne dwa podzbiory A_1 i A_2. W rezultacie otrzymujemy nowe podzbiory:

$A_1 = \{ q_0, q_1, q_6, q_7 \}$
$A_2 = \{ q_3, q_4 \}$
$B = \{ q_2, q_5 \}$

Tworzymy kolejną, zmodyfikowaną tabelę przejść, jednak do grupowania stanów oraz opisu stanów następnych używamy teraz symboli podzbiorów A_1, A_2 oraz B:

Tabela 9

	A_1				A_2		B	
stany / wejście	q_0	q_1	q_6	q_7	q_3	q_4	q_2	q_5
z_1	A_1	A_2	A_2	A_1	A_2	A_2	A_1	A_1
z_2	A_1	A_1	A_1	A_1	B	B	A_1	A_1

Na podstawie otrzymanej tabeli 9 podzbiór A_1 dzielony jest na dwa kolejne podzbiory:

$A_{11} = \{ q_0, q_7 \}$
$A_{12} = \{ q_1, q_6 \}$

Podzbiory A_2 i B nie podlegają już dalszym podziałom. Otrzymujemy kolejną postać zmodyfikowanej tabeli przejść:

Tabela 10

	A_{11}		A_{12}		A_2		B	
stany / wejście	q_0	q_7	q_1	q_6	q_3	q_4	q_2	q_5
z_1	A_{12}	A_{12}	A_2	A_2	A_2	A_2	A_{12}	A_{12}
z_2	A_{11}	A_{11}	A_{11}	A_{11}	B	B	A_{11}	A_{11}

Ponieważ dalszy podział podzbiorów na kolejne podzbiory nie jest już możliwy, algorytm minimalizacji kończy działanie. Stany zawarte w poszczególnych podzbiorach należy więc uznać za zgodne (a także równoważne). Automat po minimalizacji zawiera cztery stany (oznaczymy je symbolami a_i), które reprezentują podzbiory stanów występujące w kolejnych fazach przekształceń.

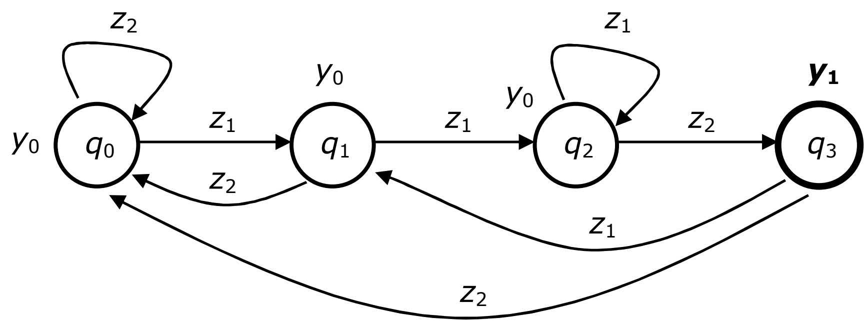
$a_0 = A_{11} = \{ q_0, q_7 \}$
$a_1 = A_{12} = \{ q_1, q_6 \}$
$a_2 = A_2 = \{ q_3, q_4 \}$
$a_3 = B = \{ q_2, q_5 \}$

Tabela 11 przedstawia funkcję przejść będącą końcowym efektem minimalizacji.

Tabela 11

wyjście	y_0	y_0	y_0	$\mathbf{y_1}$
stany / wejście	a_0	a_1	a_2	a_3
z_1	a_1	a_2	a_2	a_1
z_2	a_0	a_0	a_3	a_0

Graf automatu według tabeli 11 przedstawiony został na rys. 5. Dla przejrzystości przyjęto standardowe oznaczenia stanów a_i jako q_i.



Rys. 5. Graf automatu Moore'a z przykładu 3 po minimalizacji

5. Przebieg ćwiczenia

Dla automatu zadanego przez prowadzącego, w postaci wyrażenia regularnego, należy wykonać syntezę abstrakcyjną. Automat może być podany również w postaci opisu słownego, wówczas w pierwszym kroku należy określić wyrażenie regularne.

Dla uzyskanego grafu automatu należy wykonać syntezę strukturalną a następnie uruchomić i przetestować układ na zestawie UNILOG. Zależnie od wskazówek prowadzącego testy mogą być też przeprowadzone w postaci symulacji komputerowej.

6. Sprawozdanie z ćwiczenia

W sprawozdaniu należy umieścić:

- temat i cel ćwiczenia,
- schematy zamodelowanych automatów,
- wyniki testowania automatów,
- wnioski z ćwiczenia.

7. Literatura

[1]. Bromirski J.: Teoria automatów, WNT, Warszawa, 1969

[2]. Kazimierczak J., Kluska J., Kaczmarek A.: Podstawy teorii automatów - laboratorium, Wydawnictwa Politechniki Rzeszowskiej, Rzeszów, 1984

[3]. Wawiłow E. N., Portnoj G. P.: Synteza układów elektronicznych maszyn cyfrowych, WNT, Warszawa, 1967