R O Z D Z I A Ł   V I I

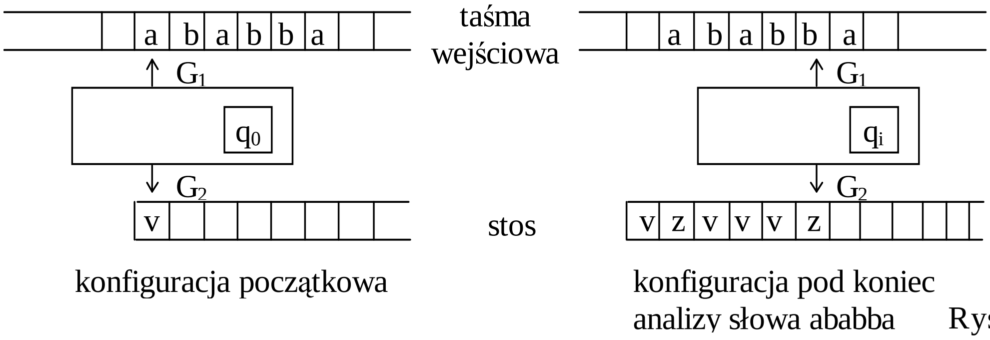
Automaty ze stosem

1. Podstawowe definicje

Rozważane w poprzednim rozdziale automaty Rabina-Scotta były urządzeniami o bardzo ograniczonej zdolności przechowywania informacji. Rzeczywista pojemność ich pamięci była wyrażona liczbą stanów wewnętrznych, bowiem jedynym sposobem zapamiętania czegokolwiek było przyjęcie odpowiedniego stanu. Stąd też struktura słów akceptowalnych przez te automaty była ściśle określona (akceptowały one mianowicie jedynie języki regularne). Celem zagwarantowania akceptacji dowolnego słowa np. języka $\{a^i b^i : i \in N\}$ musimy się posłużyć bardziej rozbudowanym automatem o nieskończonej pamięci (nie wiemy bowiem, jak długie będzie takie słowo). W tym celu, obok wewnętrznej pamięci stanowej z automatów Rabina-Scotta, będziemy stosować jeszcze dodatkową (tym razem już zewnętrzną) pamięć - tzw. pamięć stosową. W ten sposób otrzymamy właśnie automat ze stosem.

Na początku zastanówmy się, czym właściwie jest ów stos. Najbardziej obrazowo możemy go sobie wyobrazić jako plik kartek ułożonych jedna na drugą. Na każdej kartce zapisana jest pewna informacja. Ze stosu (tylko z wierzchu!) można zdejmować kartki (po jednej) lub nakładać je na niego (dowolną ilość). Wzięciu kartki odpowiada odczytanie informacji (wraz z bezpowrotnym zniszczeniem jej), a nałożeniu pliku kartek - jej zapisanie. Chcąc więc dotrzeć do informacji zapisanej wewnątrz stosu (bądź na jego spodzie) - trzeba bezpowrotnie zniszczyć wszystkie kartki (wraz z informacjami na nich umieszczonymi), leżącymi ponad nią. Tak więc stos działa według zasady „FILO”, tj. „first in - last out” (pierwszy wchodzący - ostatni wychodzi), gdyż kartka, która została w danym miejscu położona na stos, zostanie z niego usunięta jako ostatnia w stosunku do kartek, które zostały położone na stos nie wcześniej niż ona. Tę samą metodę pracy stosu opisuje równoważna jej zasada „LIFO” („last in - first out”, tj. „ostatni wchodzący - pierwszy wychodzi”). Oczywiście, tak określona pamięć stosowa jest nieograniczona (gdyż możemy układać dowolnie wysoki stos kartek). Informacja zgromadzona na stosie może być jednak wyczerpana ze stosu po przeczytaniu najniższej (najwcześniej umieszczonej na stosie) kartki.

W automacie ze stosem, za model stosu obieramy taśmę zakończoną z lewej strony. Występuje ona obok (znanej już nam z poprzedniego rozdziału) taśmy wejściowej zawierającej czytane słowo. W konfiguracji początkowej (patrz lewa strona rys. 7.1) głowica G_1 jest ustawiona na klatce z taśmy wejściowej, w której znajduje się pierwsza litera czytanego słowa, a głowica G_2 - na pierwszej klatce w stosie (jest on cały pusty, za wyjątkiem tej właśnie klatki, w której to znajduje się symbol początkowy stosu). Taśma wejściowa może się przesuwać w jedna stronę (podobnie, jak w automatach skończonych), a dodatkowy (w stosunku do automatów skończonych) stos - w dwie. Równoważnie możemy oczywiście mówić, że wyżej wymienione taśmy wcale się nie przesuwają, lecz że przesuwają się po nich głowice tego automatu. Głowica G_2 wskazuje zawsze na wierzch stosu, a w przypadku, gdy stos jest pusty - na miejsce bezpośrednio przed nim.



Rys. 7.1.

Formalnie, a u t o m a t e m z e s t o s e m nazywamy uporządkowaną siódemkę $\mathcal{A}$ = <Z, K, T, M, z_0, q_0, H>, w której:

– Z jest alfabetem stosu,
– K jest zbiorem stanów automatu,
– T jest alfabetem taśmy wejściowej,
– M jest relacją przejścia: $M \subseteq (Z \times K \times (T \cup \{\lambda\})) \times (Z^* \times K)$; ponieważ (ogólnie rzecz biorąc) jest to relacja (a nie funkcja), więc opisany nią konkretny automat może być tak deterministyczny, jak i niedeterministyczny,
– $z_0 \in Z$ jest symbolem początkowym stosu,
– $q_0 \in K$ jest stanem początkowym automatu,
– $H \subseteq K$ jest zbiorem stanów akceptowalnych przez automat (nazywany go też zbiorem stanów końcowych).

Tak więc, w stosunku do definicji automatu skończonego, doszły tu Z i z_0 - alfabet i symbol początkowy stosu, tj. elementy charakteryzujące stos.

Na naszą szczególną uwagę zasługuje relacja M, gdyż to ona właśnie opisuje poszczególne cykle pracy automatu. Wygląda to następująco:

– automat (za pomocą głowicy G_2) odczytując symbol z wierzchu stosu Z (wymazując go zarazem), sam będąc w stanie wewnętrznym ze zbioru K i (za pomocą głowicy G_1) czytając z taśmy wejściowej symbol alfabetu T (jest to kolejna litera analizowanego słowa)
– zapisuje na stosie element z Z^* i przechodzi w stan wewnętrzny K.

Na stos zapisuje on więc choćby po kilka elementów, a czyta i zrzuca z niego po jednym elemencie. Głowica G_2 ustawiona jest zawsze na ostatnim elemencie stosu, a głowica G_1 na kolejnym elemencie analizowanego słowa. Tak więc powyższa formalna definicja automatu ze stosem opisuje (tu: poprzez relację przejścia) w identyczny sposób działanie stosu, jak to sami uczyniliśmy poprzednio.

Tak np., automat analizując, czy słowo aaabbb należy do języka $L = \{a^i b^i : i \in N\}$, najpierw czytając a - odkłada je na stosie, a następnie czytając b zdejmuje ze stosu po jednym a. Gdy moment wyzerowania stosu do jego stanu początkowego zbiegnie się z momentem przeczytania całego słowa - będzie to oznaczać, że automat ten akceptuje to słowo.

K o n f i g u r a c j ą w automacie ze stosem nazywamy wyrażenie $Wq \in Z^*K$, gdzie $W \in Z^*$, a $q \in K$. Są to więc jedynie jego dane wewnętrzne: ciąg symboli na stosie wraz z aktualnym stanem automatu.

S ł o w e m w automacie ze stosem nazywamy zaś wyrażenie $WqV \in Z^*KT^*$, gdzie $W \in Z^*$, $q \in K$, a $V \in T^*$. Jest to więc konfiguracja automatu (jego środowisko

(*)
$z_a\, z_b\, z_c\, z_d\ \ z\ q\ a\ \ a_a\, a_b\, a_c\, a_d$ stos = pamięć zewnętrzna
stan = pamięć wewnętrzna
stos stan analizowane słowo
KONFIGURACJA = środowisko wewnętrzne
SŁOWO = środowisko zewnętrzne Diagram 7.1.

wewnętrzne) wraz z nieprzeczytanym jeszcze fragmentem analizowanego słowa (jego środowisko zewnętrzne). Tak więc mamy sytuację jak na diagramie 7.1.

Niech:

- z oznacza ostatni symbol na stosie, ową trójkę zaznaczy-
- q oznacza stan wewnętrzny automatu, liśmy na powyższym

- a - odczytywany właśnie symbol z taśmy diagramie przez „(*)”
(ową trójkę oznaczyliśmy na powyższym diagramie przez „(∗)”).

Wtedy $M(z,q,a) = \{(z_1,q_1), \ldots, (z_m,q_m)\}$, gdzie $z_i \in Z^*$, $q_i \in K$, zaś m jest dowolną liczbą naturalną (zauważmy, że mamy tu zbiór rozwiązań, a nie tylko jedno rozwiązanie, bo zgodnie z definicją M jest relacją, a nie koniecznie funkcją). Czytając więc trójkę (z,q,a) - automat ów zamienia fragment słowa w jedną z par (z_i, q_i), gdzie $1 \le i \le m$. Dodajmy tu jeszcze, że dla oznaczenia, że $(z_i, q_i) \in M(z,q,a)$ stosujemy zapis $zqa \to z_iq_i$.

Widzimy więc, że za sprawą relacji M konfiguracja $Wq = W_1zq$ (tj. taka, w której z jest ostatnim symbolem w W) przechodzi w konfigurację $\underbrace{W_1\, z_i}_{\text{stos}}\ \underset{\text{stan}}{q_i}$
gdzie $W, W_1, z_i \in Z^*$, $z \in Z$, zaś $q, q_i \in K$.

Jeżeli przy powyższych oznaczeniach w „$zqa \to z_iq_i$”:

– $z_i = \lambda$, to oznacza to, że po zdjęciu z ze stosu, nic nie weszło z powrotem na jego miejsce, w wyniku czego długość stosu zmniejszyła się o 1,
– $z_i = z'$ (gdzie $z' \in Z$), to oznacza to, że w stosie na miejscu elementu z znalazł się element z' (czyli długość stosu nie zmieniła się); jeżeli przy tym $z' = z$, to stos w ogóle się nie zmienił,
– $z_i = z'w$ (gdzie $z' \in Z$, zaś $w \in Z^*$), to oznacza to, że długość stosu zwiększyła się o $|w|$ (długość słowa w); jeżeli przy tym $z' = z$, to wówczas w stosunku do stanu początkowego na stos dodatkowo zostały położone elementy w.

Uwaga.

Możemy spotkać się także z instrukcją postaci $zq \to z_iq_i$, gdzie $z \in Z$, $z_i \in Z^*$, zaś $q, q_i \in K$ (jest to oczywiście instrukcja $zqa \to z_iq_i$ z $a = \lambda$). Powoduje ona po wymazaniu ze stosu z zapisanie na nim z_i, pozostawiając wejście bez zmian (taśma wejściowa nie zmienia się i stoi w miejscu). Instrukcję tę wykonujemy bez względu na to, co stoi na wejściu (λ, czy też słowo niepuste).

Instrukcję tę wykorzystamy w dowodzie twierdzenia 7.1, gdzie za u wstawimy z'z (gdzie $z, z' \in Z$). Będzie więc ona oznaczała wpisanie na stos dodatkowego elementu bez „ruszania” analizowanego słowa. Ze względu na to, że będzie ona tam wykonywana tylko na początku działania automatu - będzie oznaczała wpisanie dodatkowego elementu z' na samo dno stosu.

Podobnie, instrukcja ta jest stosowana, gdy automat nie ma już co czytać, gdyż całe słowo zostało już przeczytane (wtedy oczywiście obligatoryjnie $a = \lambda$), lecz jego analiza przez automat nadal jest kontynuowana. W sytuacji tej taśma wejściowa oczywiście stoi, co (przy niewłaściwym doborze instrukcji) może spowodować zapętlenie się pracy automatu. □

Taśma może stać również wówczas, gdy stan przejścia automatu $\mathcal{A}$ w danej sytuacji nie jest wyznaczony przez relację M (nie musi być bowiem ona całkowita). Ze stanami nieokreślonymi dla danej trójki (z,q,a) mamy do czynienia wówczas, gdy zależy nam, aby automat nie akceptował słów, w których teraz miałoby wystąpić a. Automat ze stosem nie musi być więc zupełny. Oznacza to z kolei, że w tym przypadku w czasie analizy (złego) słowa może on stanąć.

Wspomnijmy jeszcze w tym miejscu, że stosu oczywiście nigdy nie da się zatrzymać w trakcie pracy automatu (zawsze bowiem będzie on aktywny - najpierw automat zrzuci z niego pierwszy symbol z wierzchu, a następnie zapisze na nim pewną liczbę symboli).

Podajmy jeszcze pozostałe, potrzebne nam, definicje.

Mówimy, że automat ze stosem $\mathcal{A}$ r e d u k u j e w j e d n y m k r o k u słowo

$X \in Z^*KT^*$ do słowa $Y \in Z^*KT^*$ (co oznaczamy: $X \underset{\mathcal{A}}{\rightarrow} Y$) witw, gdy:

biorąc pod uwagę, że X i Y są odpowiednio następującej postaci

$$X = W\ z\ q\ a\ P \qquad \text{(gdzie } q, p \in K,$$

$$Y = \underbrace{W\ u}_{\text{stos}}\ \underset{\text{stan}}{\underset{\downarrow}{p}}\ \underbrace{P}_{\text{taśma}} \qquad W, u \in Z^*, z \in Z,$$

$$P \in T^*, a \in T \cup \{\lambda\})$$

mamy, że: $(u,p) \in M(z,q,a)$.

Widzimy, że automat w pojedyńczym kroku czytając - zdejmuje z taśmy po jednym symbolu (a), a zdejmując ze stosu po jednym symbolu (z) - zapisuje na nim dowolną liczbę symboli alfabetu stosu (u). W wyniku tej operacji automat przechodzi z jednego stanu (q) w drugi (p), być może nawet ten sam (p = q).

Mówimy, że X r e d u k u j e s i ę do Y (lub: automat $\mathcal{A}$ redukuje X do Y) witw,

gdy istnieje ciąg $X = X_1, \ldots, X_n = Y$ wyrażeń nad Z^*KT^* taki, że $X_i \underset{\mathcal{A}}{\rightarrow} X_{i+1}$ dla

każdego i = 1, …, n - 1.

Przez L($\mathcal{A}$) oznaczamy j ę z y k a k c e p t o w a n y p r z e z a u t o m a t z e s t o s e m $\mathcal{A}$, zdefiniowany w następujący sposób:

$$L(\mathcal{A}) = \{P \in T^* : z_0\, q_0\, P \xrightarrow[\mathcal{A}]{*} Wp,\ W \in Z^*,\ p \in H\ (\subseteq K)\},$$

czyli jest to zbiór tych wszystkich słów nad alfabetem T^*, dla których (poprzedzonych konfiguracją początkową z_0q_0) istnieje taki przebieg (czyli ciąg kolejnych redukcji), że w momencie, gdy automat przejdzie całą taśmę - musi się on znajdować w stanie akceptowalnym, lecz jego stos nie koniecznie musi być pusty.

Przykład 7.1.

Napisać automat ze stosem akceptujący język L = $\{a^nb^n : n > 0\}$.

Niech $\mathcal{A}$ = <$\{z_0, a\}$, $\{q_0, q_1, q_2\}$, {a, b}, M, z_0, q_0, $\{q_2\}$>, gdzie relacja M (będąca tu funkcją) zdefiniowana jest w następujący sposób:

M:	a	b	λ
$(z_0,\ q_0)$	$(z_0\, a,\ q_0)^1$	∅	∅
$(a,\ q_0)$	$(a\ a,\ q_0)^2$	$(\lambda,\ q_1)^3$	∅
$(z_0,\ q_1)$	∅	∅	$(\lambda,\ q_2)^5$
$(a,\ q_1)$	∅	$(\lambda,\ q_1)^4$	∅
$(z_0,\ q_2)$	∅	∅	∅
$(a,\ q_2)$	∅	∅	∅

← czyta z taśmy ($\in T \cup \{\lambda\}$)

∅ - to stan nieokreślony

Stany określone, w które może przejść automat dodatkowo ponumerowano indeksami. Są to zarazem numery operacji.

↑ ↑

ma na stosie ($\in$ Z) — jest w stanie ($\in$ K)

Tab. 7.1.

Tak więc np., jeśli automat ten ma na stosie z_0 i jest w stanie q_0, a z taśmy czyta a - to odłoży on na stos z_0a i przejdzie z powrotem w stan q_0 (stan nr 1).

Zobaczmy dla przykładu, jak ten automat akceptuje słowo $a^3\, b^3$.

Na początku mamy wyrażenie $z_0q_0$aaabbb, tj. z_0q_0P, gdzie P = aaabbb = $a^3\, b^3$.

Pokażemy, że $z_0\, q_0\, P \xrightarrow[\mathcal{A}]{*} Wp$, gdzie $W \in Z^*$, a $p \in H$.

Dokonując poszczególnych redukcji - z bazowego słowa $z_0\, q_0$ aaabbb, po kolei otrzymujemy:

$z_0\ q_0$ aaabbb			1
z_0a q_0 aabbb			2
$z_0$aa q_0 abbb			2
$z_0$aaa q_0 bbb			3
$z_0$aaλ q_1 bb	=	$z_0$aa q_1 bb	4
z_0aλ q_1 b	=	z_0a q_1 b	4
$z_0\lambda\ q_1$	=	$z_0\ q_1\ \lambda$	5
q_2			
stos stan taśma		stos stan taśma	numer operacji

Diagram 7.2.

Otrzymaliśmy W = $\lambda \in Z^*$ i p = $q_2 \in$ H = $\{q_2\}$. Zatem badane słowo P = a^3b^3 jest akceptowane przez ten automat. □

Zadanie 7.1. Niech T = {a,b}. Zbuduj automat ze stosem akceptujący tylko i wyłącznie wszystkie słowa:

a) o krotności „a” równej krotności „b” (musi więc on także akceptować słowo puste λ),

b) o krotności „a” większej od krotności „b”,

c) o postaci uu^{-1}, gdzie $u \in T^* \setminus \{\lambda\}$. □

2. Automaty z pustym stosem

Przez N($\mathcal{A}$) oznaczamy j ę z y k a k c e p t o w a n y p r z e z a u t o m a t $\mathcal{A}$ z p u s t y m s t o s e m (bądź równoważnie: przez automat zerujący stos), zdefiniowany w następujący sposób:

$$N(\mathcal{A}) = \{P \in T^*: z_0\ q_0\ P \underset{\mathcal{A}}{\overset{*}{\rightarrow}} p,\ p \in K\}.$$

Zauważmy, że:

– automat zerujący stos akceptuje wszystkie słowa, których analiza doprowadza do wyzerowania stosu (bo p po prawej stronie wyprowadzenia, to inaczej λp, gdzie $\lambda = W \in Z^*$),

– nie jest tu istotne, by p $\in$ H (zbiór stanów akceptowalnych); wystarczy, że p $\in$ K (czyli że jest jakimkolwiek stanem tego automatu).

Zachodzą następujące dwa, „działające w przeciwne strony”, twierdzenia odnoszące się do automatów ze stosem.

Twierdzenie 7.1.

Dla każdego automatu ze stosem $\mathcal{A}$, można skonstruować automat ze stosem $\mathcal{A}'$ taki, że: $L(\mathcal{A}) = N(\mathcal{A}')$.

Twierdzenie 7.2.

Dla każdego automatu ze stosem $\mathcal{A}$, można skonstruować automat ze stosem $\mathcal{A}'$ taki, że: $N(\mathcal{A}) = L(\mathcal{A}')$.

Twierdzenia te, łącznie wzięte, dają nam równoważność automatów ze stosem z automatami zerującymi stos (tj. innymi słowy, że siła akceptacyjna tych automatów jest taka sama). Poniżej przedstawiamy dowody tych twierdzeń.

Dowód tw. 7.1.

Jest to twierdzenie konstrukcyjne, więc i jego dowód będzie konstrukcyjny.

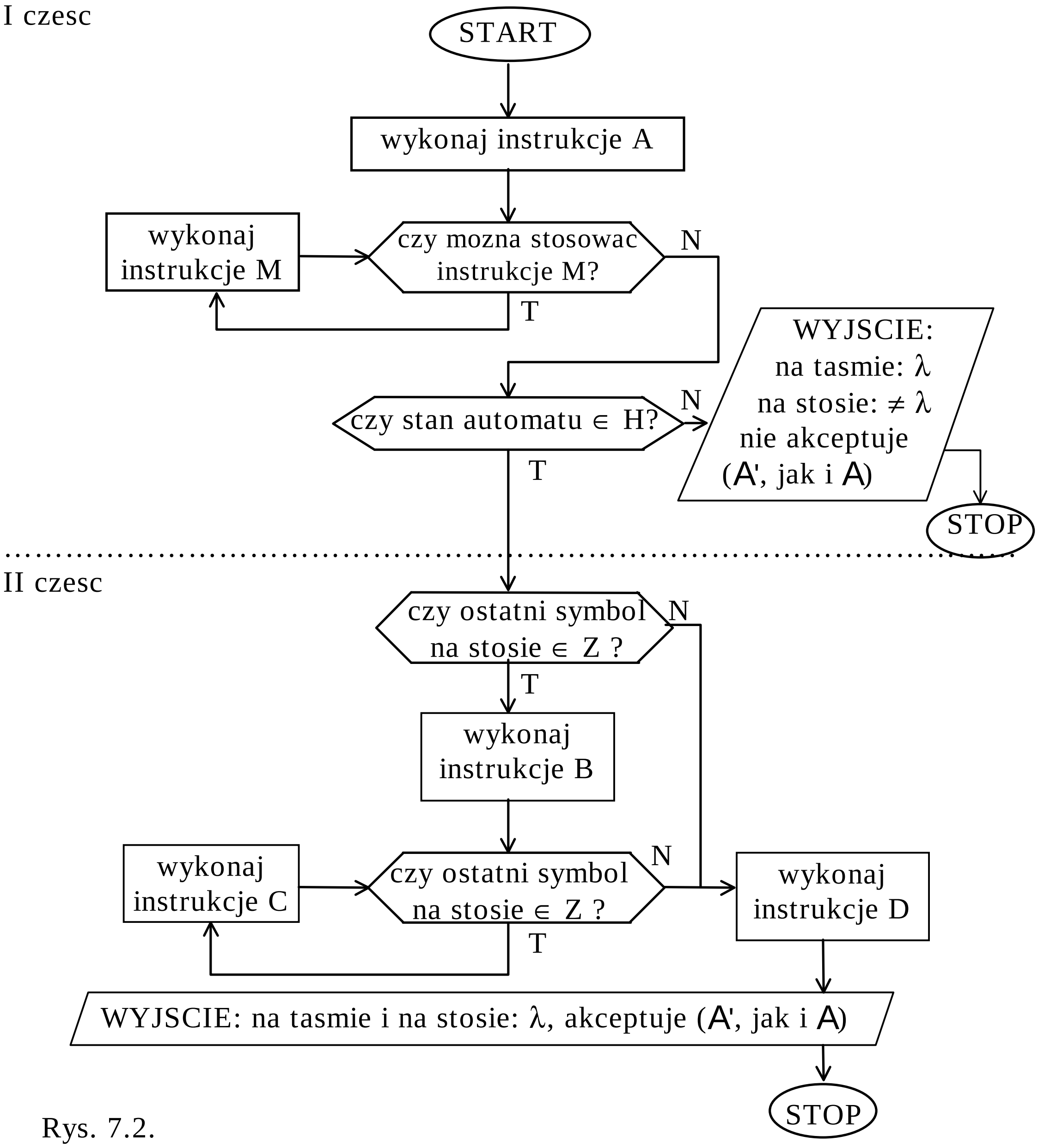
Niech $\mathcal{A} = \langle Z, K, T, M, z_0, q_0, H \rangle$.

Skonstruujemy automat $\mathcal{A}'$, który robi to samo, co automat $\mathcal{A}$, ale dodatkowo jeszcze na koniec (tj., gdy będzie już miał pustą taśmę), gdy będzie w stanie akceptowalnym automatu $\mathcal{A}$, to wówczas zrzuci jeszcze wszystko ze stosu (o ile będzie jeszcze coś na nim miał). Żeby jednak automat ten nie akceptował słów, podczas analizy których automat $\mathcal{A}$ zerował stos, nie przechodząc jednak do jednego ze stanów H - w automacie $\mathcal{A}'$ wprowadzamy dodatkowo jeden symbol na samo dno stosu.

Niech mianowicie $\mathcal{A}' = \langle Z', K', T, M', z_0', q_0', \varnothing \rangle$, gdzie:

- $K' = K \cup \{q', q_0'\}$, przy czym $q', q_0' \notin K$,
- $Z' = Z \cup \{z_0'\}$, przy czym $z_0' \notin Z$,
- $M' = M \cup A \cup B \cup C \cup D$, gdzie:
  - $A = \{(z_0'q_0'\lambda, z_0'z_0q_0)\}$,
  - $B = \{(zq\lambda, \lambda q_n'): z \in Z, q \in H\}$,
  - $C = \{(zq_n'\lambda, \lambda q_n'): z \in Z\}$,
  - $D = \{(z_0'q_n'\lambda, \lambda q_n')\}$.

Zbiór stanów akceptowalnych, jako nieistotny w automatach zerujących stos, przyjmujemy za równy $\varnothing$.



Rys. 7.2.

Sposób działania tak zdefiniowanego automatu $\mathcal{A}'$ jest przedstawiony na schemacie blokowym (rys. 7.2). Schemat ten jest zarazem dowodem równoważności automatów $\mathcal{A}$ i $\mathcal{A}'$, gdyż odczytujemy z niego, że zbiór słów akceptowalnych przez automat $\mathcal{A}$ pokrywa się ze zbiorem słów akceptowalnych przez automat $\mathcal{A}'$.

Widzimy, że automat $\mathcal{A}'$:

1) w pierwszej części swego działania najpierw przy pomocy instrukcji A umożliwia przejście do takiego sposobu czytania danego słowa, jak to czynił automat $\mathcal{A}$, a następnie wykonuje już instrukcje w identyczny sposób, jak to robił automat $\mathcal{A}$ (w tym momencie konfiguracja automatu $\mathcal{A}'$ różni się od końcowej konfiguracji

automatu $\mathcal{A}$ jedynie tym, że na początku stosu znajduje się tu dodatkowo nowy element z_0');

2) następnie, jeżeli stan automatu jest stanem końcowym automatu $\mathcal{A}$ (tj., gdy jest to słowo akceptowalne przez automat $\mathcal{A}$), to w drugiej części swego działania:
   - wszystkie symbole, które automat $\mathcal{A}$ zostawiał na stosie - automat $\mathcal{A}'$ zdejmuje ze stosu (pierwszy za pomocą instrukcji B, a pozostałe poprzez wielokrotne wykonywanie instrukcji C),
   - za pomocą instrukcji D, automat $\mathcal{A}'$ zdejmuje ostatni element ze stosu (jest nim oczywiście z_0'). □

Dowód tw. 7.2.

Również i to twierdzenie jest konstrukcyjne, więc i jego dowód będzie konstrukcyjny. Niech $\mathcal{A} = \langle Z, K, T, M, z_0, q_0, \varnothing \rangle$.

Połóżmy: $K' = K \cup \{k'\}$, gdzie $k' \notin K$,

$H' = \{k'\}$,

$M' = M \cup \{(\lambda k_i\, \lambda, \lambda k') : k' \in K\}$.

Otrzymujemy automat $\mathcal{A}' = \langle Z, K', T, M', z_0, q_0, H' \rangle$.

Pracuje on identycznie, jak automat A aż do momentu osiągnięcia konfiguracji $\lambda k_i \lambda$ (gdzie $k_i \in K$), tj. do momentu wyzerowania stosu (bez względu na stan, w jakim się znajduje). Wówczas może przejść tylko do jedynego swego stanu końcowego k' (tj., gdy dla pewnego słowa, $\mathcal{A}$ stosu nie zeruje, to $\mathcal{A}'$ go nie akceptuje, przy czym działa to również w drugą stronę). □

Zadanie 7.2. O ile skonstruowane przez ciebie automaty z zadania 7.1 nie zerują stosu - znajdź również odpowiadające im automaty zerujące stos. □

3. Automaty ze stosem, a gramatyki bezkontekstowe

Zachodzą następujące dwa twierdzenia:

Twierdzenie 7.3.

Dla każdej gramatyki bezkontekstowej G, można skonstruować automat ze stosem $\mathcal{A}$, taki że: $L(\mathcal{A}) = L(G)$. □

Twierdzenie 7.4.

Dla każdego automatu ze stosem $\mathcal{A}$, można skonstruować gramatykę bezkontekstową G, taką że: $L(\mathcal{A}) = L(G)$. □

Twierdzenia te dają nam równość klasy języków generowanych przez gramatyki bezkontekstowe z klasą języków rozpoznawalnych przez automaty ze stosem.

Tak więc:

1) każdy język bezkontekstowy jest akceptowalny przez pewien automat ze stosem, a każdy automat ze stosem rozpoznaje jedynie pewien język bezkontekstowy;
2) skoro zachodzi też równość między klasą języków rozpoznawalnych przez automaty ze stosem, a klasą języków rozpoznawalnych przez automaty zerujące stos, zatem z przechodniości otrzymujemy równość klasy języków rozpoznawalnych przez automaty zerujące stos z klasą języków bezkontekstowych, co w sumie daje nam, że
3) następujące klasy języków pokrywają się:
   - klasa języków bazkontekstowych,
   - klasa języków rozpoznawalnych przez automaty ze stosem,
   - klasa języków rozpoznawalnych przez automaty zerujące stos.

Sytuacja ta została przedstawiona na poniższym diagramie.

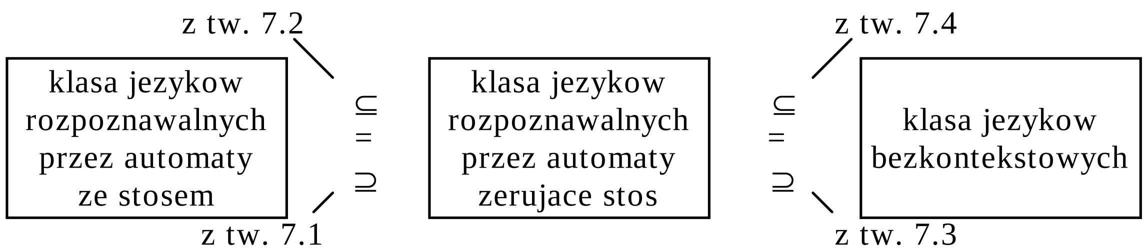


Diagram 7.3.