AGH

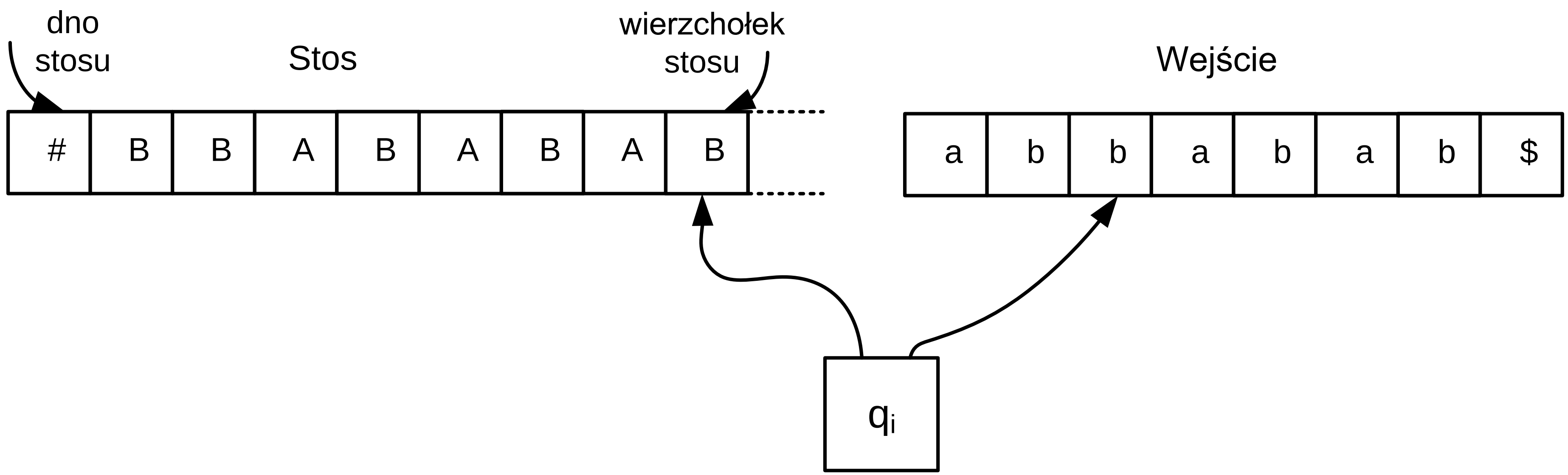
AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Automat ze stosem

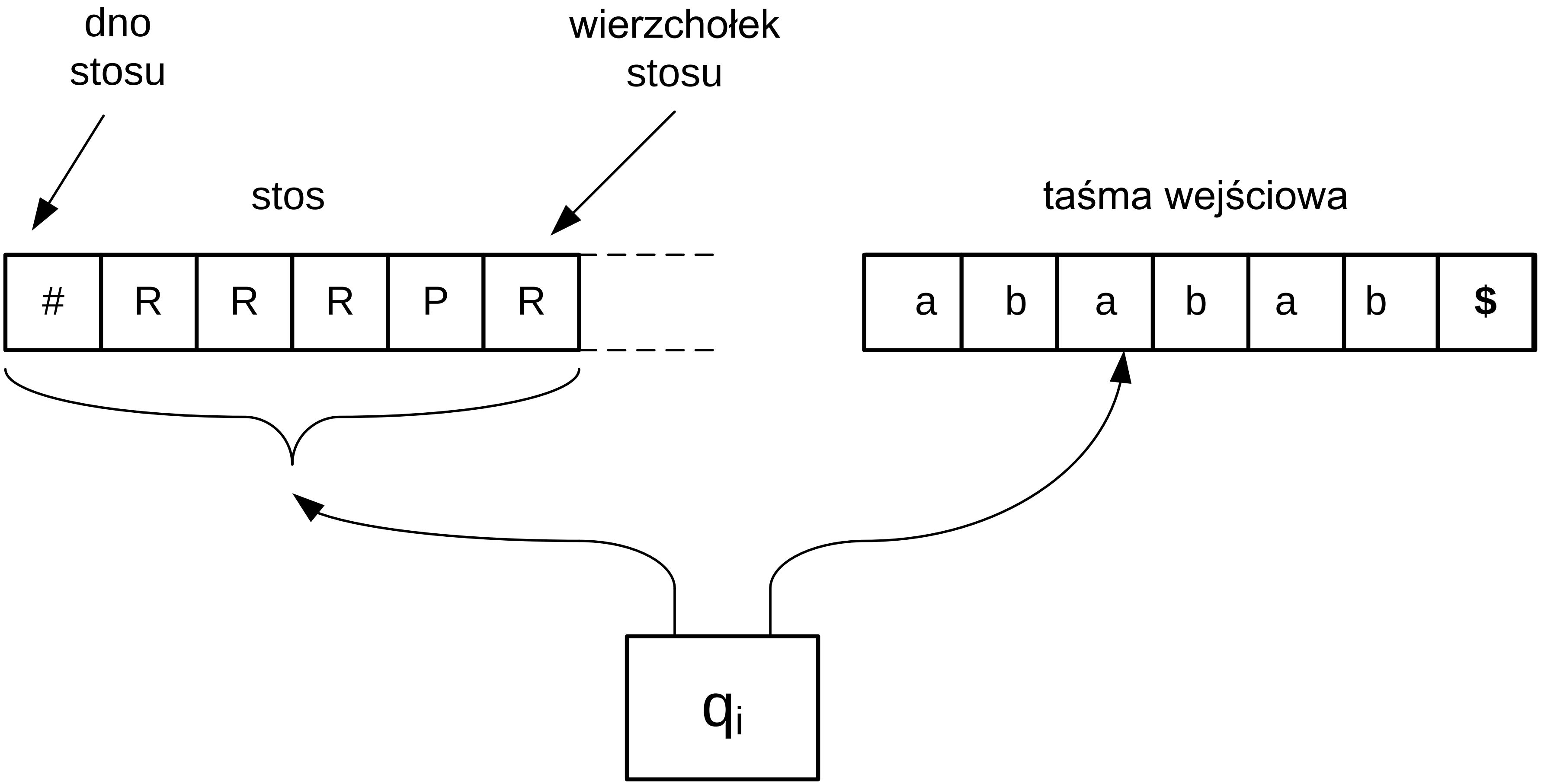
Języki formalne i automaty

Dr inż. Janusz Majewski
Katedra Informatyki

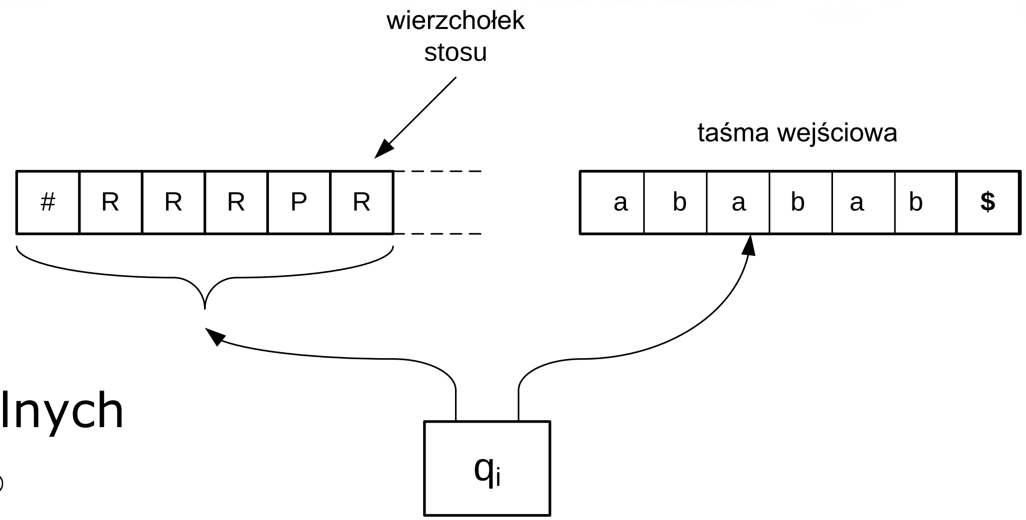
Automat ze stosem (1)



Automat ze stosem (1) c.d.



Automat ze stosem (2)



$A = < \Sigma, Q, F, q_0 , \Gamma, Z_0 , \delta, \$ >$

Σ – zbiór symboli terminalnych

Q – zbiór stanów $\#Q < \infty$

$F \subseteq Q$ – zbiór stanów końcowych

$q_0 \in Q$ – stan początkowy

Γ – zbiór symboli stosowych

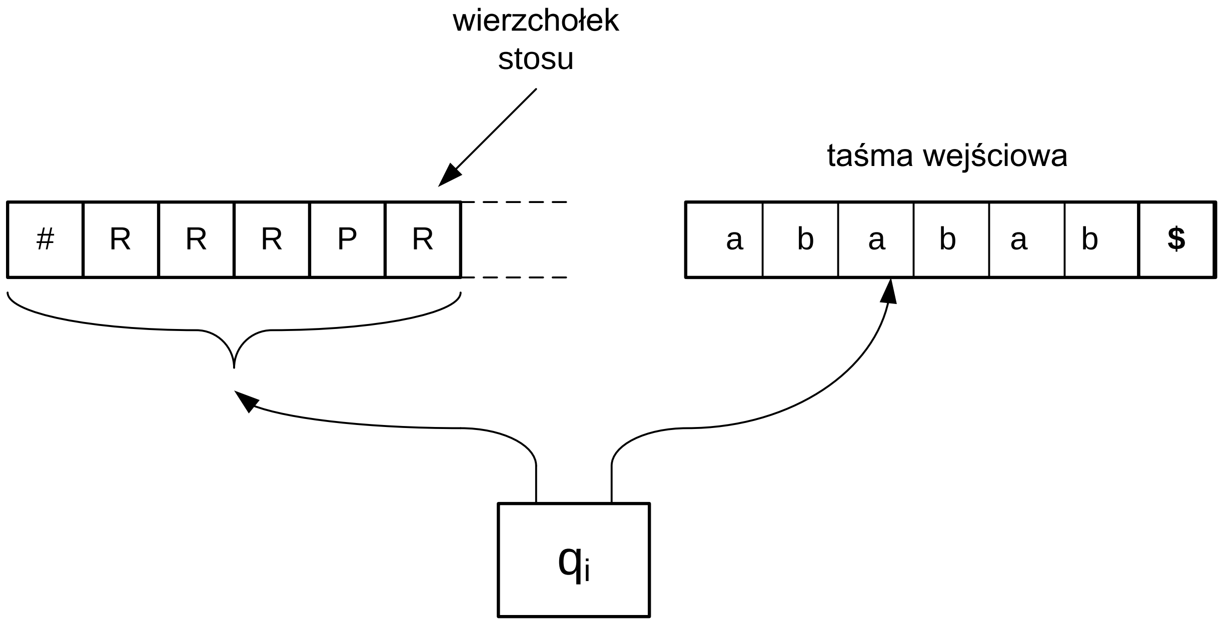
$Z_0 \in \Gamma \cup \{\varepsilon\}$ – symbol początkowy stosu

δ - funkcja przejścia

$$\delta : Q \times (\Sigma \cup \{\varepsilon, \$\}) \times \Gamma^* \ \alpha \ 2^{Q \times \Gamma^*}$$

\$ - ogranicznik końca słowa wejściowego

Konfiguracja automatu (opis chwilowy)

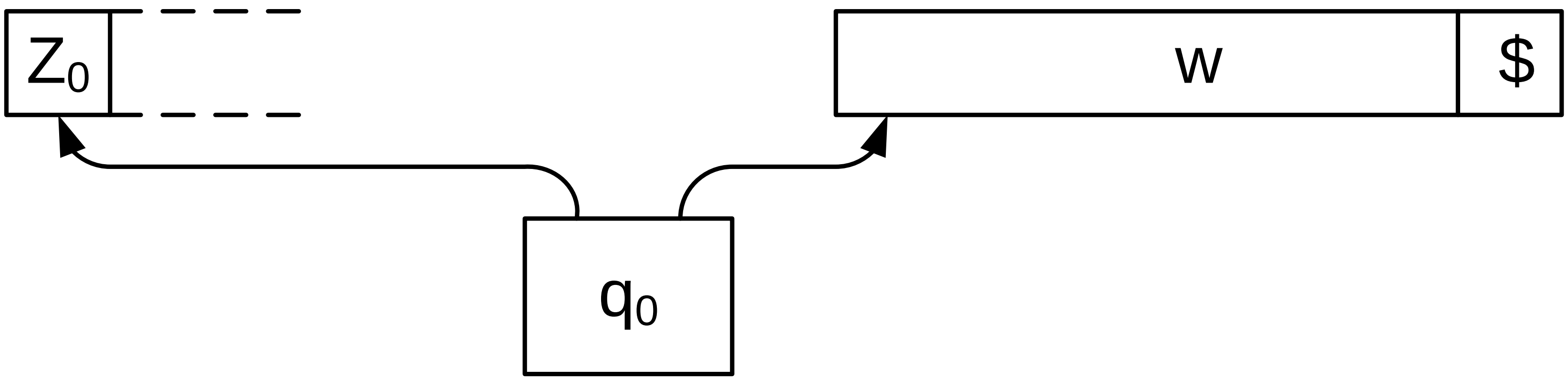


Konfiguracja automatu: (#RRRPR, q_i , abab$)

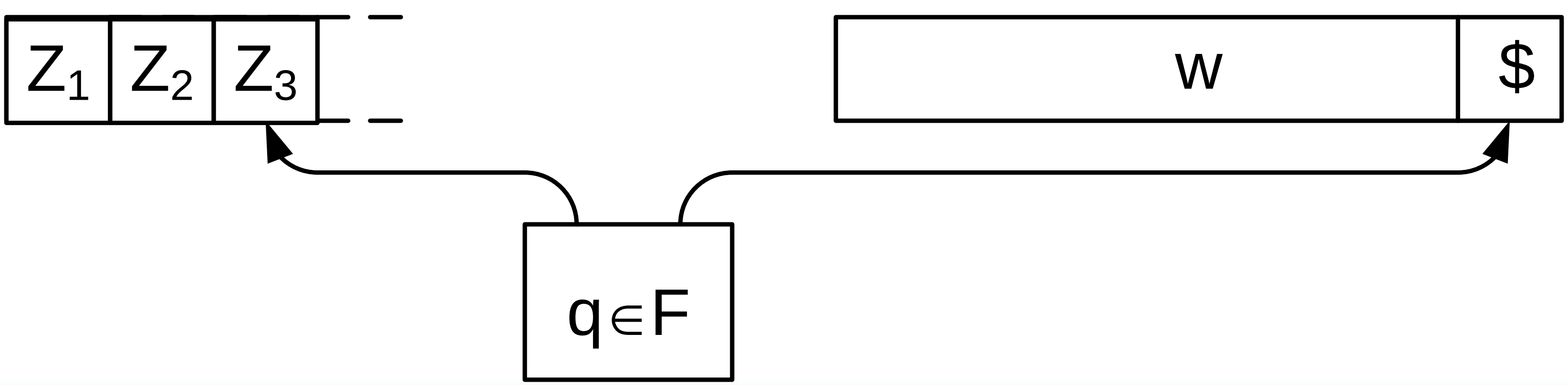
stos (wierzchołek stosu) stan nieprzeczytana część taśmy wejściowej

Akceptacja przez stan końcowy

Konfiguracja początkowa:

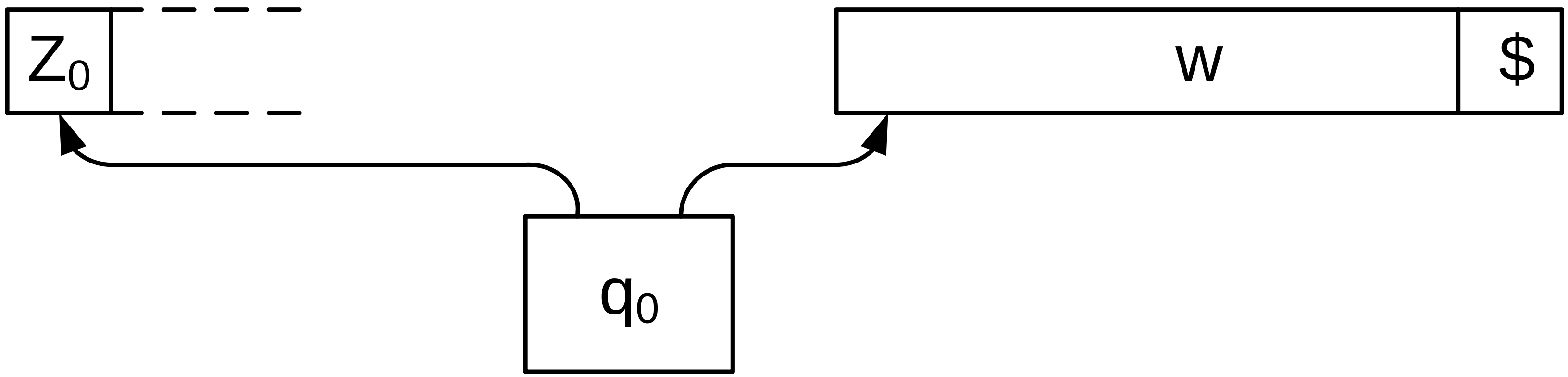


Konfiguracja akceptująca – akceptacja przez stan końcowy

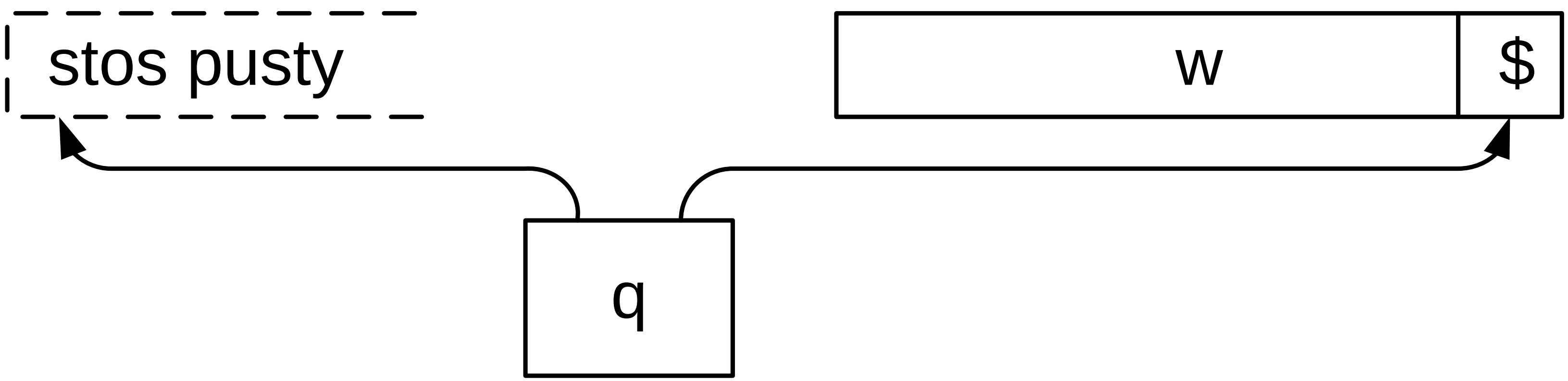


Akceptacja przez pusty stos

Konfiguracja początkowa:



Konfiguracja akceptująca – akceptacja przez pusty stos



Pojedynczy krok automatu

Funkcja przejścia:

δ(aktualny_stan, wejście, aktualny_wierzchołek_stosu) =
= (nowy_stan, nowy_wierzchołek_stosu)

Kroki automatu:

$\boldsymbol{\delta(q_n, a, XYZ) = (q_k, ST)}$ — zmiana stanu z q_n na q_k, przeczytanie *a* z wejścia z przesunięciem głowicy w prawo, podmiana *XYZ* na *ST* na wierzchołku stosu

$\boldsymbol{\delta(q_n, a, XYZ) = (q_k, \varepsilon)}$ — zmiana stanu z q_n na q_k, przeczytanie *a* z wejścia z przesunięciem głowicy w prawo, zdjęcie *XYZ* z wierzchołka stosu

$\boldsymbol{\delta(q_n, a, \varepsilon) = (q_k, ST)}$ — zmiana stanu z q_n na q_k, przeczytanie *a* z wejścia z przesunięciem głowicy w prawo, dopisanie *ST* na wierzchołku stosu

$\boldsymbol{\delta(q_n, \varepsilon, XYZ) = (q_k, ST)}$ — zmiana stanu z q_n na q_k, podmiana *XYZ* na *ST* na wierzchołku stosu bez czytania wejścia i bez przesuwania głowicy wejściowej

$\boldsymbol{\delta(q_n, \varepsilon, XYZ) = (q_k, \varepsilon)}$ — zmiana stanu z q_n na q_k, zdjęcie *XYZ* z wierzchołka stosu bez czytania wejścia i bez przesuwania głowicy wejściowej

$\boldsymbol{\delta(q_n, \varepsilon, \varepsilon) = (q_k, ST)}$ — zmiana stanu z q_n na q_k, dopisanie *ST* na wierzchołku stosu bez czytania wejścia i bez przesuwania głowicy wejściowej

Przykład: Automat ze stosem akceptujący język $L = \{ 0^n 1^n \mid n = 0,1,\ldots\}$

$A = < \Sigma, Q, F, q_0, \Gamma, Z_0, \delta, \$ >$

- $\Sigma = \{ 0, 1\}$
- $Q = \{ q_0, q_1, q_2\}$
- $F = \{q_0\}$
- q_0 – stan początkowy
- $\Gamma = \{R, 0 \}$
- $Z_0 = R$

(1) $\delta(q_0, 0, R) = \{(q_1, R0)\}$
(2) $\delta(q_1, 0, 0) = \{(q_1, 00)\}$
(3) $\delta(q_1, 1, 0) = \{(q_2, \varepsilon)\}$
(4) $\delta(q_2, 1, 0) = \{(q_2, \varepsilon)\}$
(5) $\delta(q_2, \$, R) = \{(q_0, \varepsilon)\}$
(6) $\delta(q_0, \$, R) = \{(q_0, \varepsilon)\}$

Analizowane słowo: <u>000111$</u>

$$
\begin{aligned}
& (R, q_0, 000111\$) \\
\vdash^{(1)} \quad & (R0, q_1, 00111\$) \\
\vdash^{(2)} \quad & (R00, q_1, 0111\$) \\
\vdash^{(2)} \quad & (R000, q_1, 111\$) \\
\vdash^{(3)} \quad & (R00, q_2, 11\$) \\
\vdash^{(4)} \quad & (R0, q_2, 1\$) \\
\vdash^{(4)} \quad & (R, q_2, \$) \\
\vdash^{(5)} \quad & (\varepsilon, q_0, \$)
\end{aligned}
$$

Przykład konstrukcji automatu ze stosem (1)

Budujemy automat ze stosem akceptujący wszystkie łańcuchy zbudowane z liter ***a*** oraz ***b***, w których liczba liter ***a*** jest równa liczbie liter ***b***.

Początkowo na stosie umieścimy symbol ***#***.

Jeżeli na wierzchołku stosu nie ma żadnej z liter ***a*** lub ***b***, to przeczytaną z wejścia literę umieścimy na stosie:

$\boldsymbol{\delta(q_0, a, \#) = \{(q_0, \#a)\}}$

$\boldsymbol{\delta(q_0, b, \#) = \{(q_0, \#b)\}}$

Jeżeli kolejna czytana litera jest identyczna, jak ta znajdująca się na wierzchołku stosu, to dokładamy ją na stos:

$\boldsymbol{\delta(q_0, a, a) = \{(q_0, aa)\}}$

$\boldsymbol{\delta(q_0, b, b) = \{(q_0, bb)\}}$

Przykład konstrukcji automatu ze stosem (2)

Jeżeli kolejna czytana litera jest różna od tej znajdującej się na wierzchołku stosu, to zdejmujemy literę z wierzchołka stosu:

$\boldsymbol{\delta(q_0, a, b) = \{(q_0, \varepsilon)\}}$

$\boldsymbol{\delta(q_0, b, a) = \{(q_0, \varepsilon)\}}$

Jeżeli widzimy na wejściu znacznik końca *$* i na wierzchołku stosu jest #, to akceptujemy:

$\boldsymbol{\delta(q_0, \$, \#) = \{(q_1, \varepsilon)\}}$

(Stan q_1 będzie stanem akceptującym).

Przykład konstrukcji automatu ze stosem (3)

Pełna definicja skonstruowanego automatu będzie następująca:

$A= < \Sigma, Q, F, q_0, \Gamma, Z_0, \delta, \$ >$

- $\Sigma = \{a, b\}$
- $Q = \{q_0, q_1\}$
- $F = \{q_1\}$
- q_0 – stan początkowy
- $\Gamma = \{\#, a, b\}$
- $Z_0 = \#$

$\delta(q_0, a, \#) =\{(q_0, \#a)\}$
$\delta(q_0, b, \#) =\{(q_0, \#b)\}$
$\delta(q_0, a, a) =\{(q_0, aa)\}$
$\delta(q_0, b, b) =\{(q_0, bb)\}$
$\delta(q_0, a, b) =\{(q_0, \varepsilon)\}$
$\delta(q_0, b, a) =\{(q_0, \varepsilon)\}$
$\delta(q_0, \$, \#) =\{(q_1, \varepsilon)\}$

Akceptacja języka przez automat ze stosem

$x \in \Sigma^*$ jest słowem akceptowanym przez automat A (ze stosem) przy stanie końcowym jeżeli **istnieje** taka sekwencja kroków automatu, która powoduje przejście automatu z konfiguracji początkowej do konfiguracji akceptującej przez stan końcowy.

Język L jest akceptowany przez automat A przy stanie końcowym (co oznaczamy L(A)) $\Leftrightarrow$

$$L = L(A) = \{x \in \Sigma^* \mid x \text{ jest akceptowane przez A przy stanie końcowym}\}$$

$x \in \Sigma^*$ jest słowem akceptowanym przez automat A (ze stosem) przy pustym stosie jeżeli **istnieje** taka sekwencja kroków automatu, która powoduje przejście automatu z konfiguracji początkowej do konfiguracji akceptującej przez pusty stos.

Język L jest akceptowany przez automat A przy pustym stosie (co oznaczamy N(A)) $\Leftrightarrow$

$$L = N(A) = \{x \in \Sigma^* \mid x \text{ jest akceptowane przez A przy pustym stosie}\}$$

Języki akceptowane przez automaty ze stosem

Poniższe trzy klasy języków pokrywają się ze sobą:

- Języki bezkontekstowe, czyli języki generowane przez gramatyki, których lewe strony produkcji są pojendynczymi symbolami nieterminalnymi
- Języki akceptowane przez automaty ze stosem przy stanie końcowym
- Języki akceptowane przez automaty ze stosem przy pustym stosie

Przykład automatu ze stosem odtwarzającego wywód lewostronny metodą top-down dla gramatyki wyrażeń

Przykład:

$G = <\{E, T, F\}, \{\underline{id}, +, *, (\ ,)\},$

$P = \{ E \to E+T \ | \ T$

$T \to T*F \ | \ F$

$F \to (E) \ | \ \underline{id}\}, \qquad E >$

$A = <\{\underline{id}, +, *, (,)\}, \{q\}, \varnothing, q, \{E, T, F, \underline{id}, +, *, (,)\}, E, \delta, \$>$

$\delta(q, \varepsilon, E) = \{_{(1)}(q, T+E), _{(2)}(q, T)\}$

$\delta(q, \varepsilon, T) = \{_{(3)}(q, F*T), _{(4)}(q, F)\}$

$\delta(q, \varepsilon, F) = \{_{(5)}(q,)E( ), _{(6)}(q, \underline{id})\}$

$\delta(q, b, b) = \{_{(pop)}(q, \varepsilon)\}$ dla wszystkich $b \in \{ \underline{id}, +, *, (\ ,)\}$

Analiza słowa: „ id + id * id ”

E,	q,	id + id * id $	⊢ (1)
T+E,	q,	id + id * id $	⊢ (2)
T+T,	q,	id + id * id $	⊢ (4)
T+F,	q,	id + id * id $	⊢ (6)
T+ id ,	q,	id + id * id $	⊢ (pop)
T+ ,	q,	+ id * id $	⊢ (pop)
T ,	q,	id * id $	⊢ (3)
F * T,	q,	id * id $	⊢ (4)
F * F ,	q,	id * id $	⊢ (6)
F * id,	q,	id * id $	⊢ (pop)
F * ,	q,	* id $	⊢ (pop)
F ,	q,	id $	⊢ (6)
id,	q,	id $	⊢ (pop)
ε,	q,	$	

E
E + T
T + T
F + T
id + T
id + T
id + T
id + T * F
id + F * F
id + id * F
id + id * F
id + id * F
id + id * id
id + id * id

Wyprowadzenie lewostronne top-down

Akceptacja przez pusty stos

Symbole zdjęte przez (pop)

Przykład automatu ze stosem odtwarzającego wywód prawostronny metodą bottom-up dla gramatyki wyrażeń

$G = <\{E, T, F\}, \{\underline{id}, +, *, (\ ,)\},$

$\{ E \rightarrow E+T \mid T$

$T \rightarrow T^* F \mid F$

$F \rightarrow (E) \mid \underline{id}\ \},\qquad E >$

$A = <\{\underline{\ id}, +, *, (\ ,)\}, \{q_0, q_1\}, \{q_1\}, q_0, \{E, T, F, \underline{id}, +, *, (\ ,), \#\}, \#, \$>$

(shift) $\delta(q_0, b,\ \varepsilon\} = \{(q_0, b)\}$ dla wszystkich $b \in \{\ \underline{id}, +, *, (\ ,)\}$

(1) $\delta(q_0, \varepsilon, E + T\} = \{(q_0, E)\}$

(2) $\delta(q_0, \varepsilon, T\ \} = \{(q_0, E)\}$

(3) $\delta(q_0, \varepsilon, T^*F\ \} = \{(q_0, T)\}$

(4) $\delta(q_0, \varepsilon, F\} = \{(q_0, T)\}$

(5) $\delta(q_0, \varepsilon, (E)\} = \{(q_0, F)\}$

(6) $\delta(q_0, \varepsilon, \underline{id}\ \} = \{(q_0, F)\}$

(acc) $\delta(q_0, \$, \#E\ \} = \{(q_1, \varepsilon)\}$

Analiza słowa: „ id + id * id”

#,	q_0,	id + id * id $	↱(shift)	id + id * id
# id,	q_0,	+ id * id $	↱(6)	
# F,	q_0,	+ id * id $	↱(4)	F + id * id
#T,	q_0,	+ id * id $	↱(2)	T + id * id
#E,	q_0,	+ id * id $	↱(shift)	E + id * id
#E +,	q_0,	id * id $	↱(shift)	
#E + id,	q_0,	* id $	↱(6)	
#E + F,	q_0,	* id $	↱(4)	E + F * id
#E + T,	q_0,	* id $	↱(shift)	E + T * id
#E + T*,	q_0,	id $	↱(shift)	
#E + T* id,	q_0,	$	↱(6)	
#E + T *F,	q_0,	$	↱(3)	E + T * F
#E + T ,	q_0,	$	↱(1)	E + T
#E ,	q_0,	$	↱(acc)	E
ε,	q_1,	$		

Wyprowadzenie prawostronne bottom-up.
Wierzchołek stosu przy redukcjach
to prawa granica osnowy

Deterministyczny automat ze stosem

Automat ze stosem jest deterministyczny wtedy i tylko wtedy, gdy w każdej konfiguracji może wykonać co najwyżej jeden ruch.

Twierdzenie: Klasa języków akceptowanych przez deterministyczne automaty ze stosem jest właściwą podklasą klasy języków akceptowanych przez automaty ze stosem.

Innymi słowy: nie dla każdego automatu ze stosem istnieje równoważny mu deterministyczny automat ze stosem.

$$L(A_{\text{Deterministyczny ze Stosem}}) \subset L(A_{\text{ze Stosem}})$$

Przykład

$L=\{xx^R \mid x \in \Sigma^*\}$ – jest językiem nieakceptowalnym przez deterministyczny automat ze stosem.

Przypuśćmy, że A jest automatem ze stosem akceptującym język *L* i niech $y \in \Sigma^*$ będzie dowolnym słowem przeznaczonym do analizy przez automat. Aby sprawdzić, czy *y* jest postaci xx^R trzeba przepisać lewą połowę słowa y na stos, tzn. przejść od konfiguracji (ε, q_0, $xx^R\$$) do konfiguracji (x, q, $x^R\$$), a następnie przystąpić do sprawdzenia, czy słowo na stosie jest zwierciadlanym odbiciem słowa pozostającego na wejściu. Takie postępowanie wymaga umiejętności odszukiwania połowy (środka) słowa *y*, co przy jednokrotnym jego czytaniu jest oczywiście niemożliwe.

Można pokazać, że deterministyczny automat akceptujący język *L* istnieje, nie jest to już jednak automat ze stosem, ale tzw. automat liniowo ograniczony lub maszyna Turinga.