AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Gramatyki regularne

Teoria automatów i języków formalnych

Dr inż. Janusz Majewski
Katedra Informatyki

Gramatyki regularne

G = < V,Σ,P,S > jest gramatyką prawostronnie liniową, jeśli jej produkcje mają postać:

$$\left.\begin{array}{l}(i)\ U \to xW \\ (ii)\ U \to x\end{array}\right\} \quad x \in \Sigma^* \quad U,W \in V$$

G = < V,Σ,P,S > jest gramatyką prawostronnie regularną, jeśli jej produkcje mają postać:

$$\left.\begin{array}{l}(i)\ U \to aW \\ (ii)\ U \to a\end{array}\right\} \quad a \in \Sigma \quad U,W \in V$$

oraz (iii) jeśli $(S \to \varepsilon) \in P$ to S nie występuje w prawych stronach żadnej produkcji.

(Często w definicji gramatyki regularnej pomija się warunek (iii) dotyczący niewystępowania S w prawych stronach produkcji – dopuszcza się za to produkcje $U \to \varepsilon$; $U \in V$)

Analogicznie określa się :

-gramatyki lewostronnie liniowe

$$\left.\begin{array}{l}U \to Wx \\ U \to x\end{array}\right\} \quad x \in \Sigma^* \quad U,W \in V$$

-gramatyki lewostronnie regularne

$$\left.\begin{array}{l}U \to Wa \\ U \to a\end{array}\right\} \quad a \in \Sigma \quad U,W \in V$$

Szkic algorytmu przekształcania gramatyki prawostronnie liniowej w prawostronnie regularną

Wejście: $G = < V,\Sigma,P,S > \in \mathcal{G}_{PLN}$

Wyjście: $G' = < V,\Sigma,P,S > \in \mathcal{G}_{PRG}$ taka, że $L(G) = L(G')$

Metoda:

$P' := P$; $V' := V$;

for $(A \to \alpha) \in P$ do

begin

if $\alpha = xB$ and $x = x_1 \ldots x_n \in \Sigma^*$ and $|x| \geq 2$ then

begin

$C := \{ A \to x_1C_1 ; C_1 \to x_2C_2 ; \ldots ; C_{n-1} \to x_nB \}$;

$P' := P' \setminus \{ A \to xB \} \cup C$;

$V' := V' \cup \{ C_1,\ldots,C_{n-1} \}$;

end;

if $\alpha = x$ and $x = x_1 \ldots x_n \in \Sigma^*$ and $|x| \geq 2$ then

begin

$C := \{ A \to x_1C_1 ; C_1 \to x_2C_2 ; \ldots ; C_{n\text{-}}1 \to x_nB \}$;

$P' := P' \setminus \{ A \to x \} \cup C$;

$V' := V' \cup \{ C_1,\ldots,C_{n-1} \}$;

end;

end;

Szkic algorytmu przekształcania gramatyki prawostronnie liniowej w prawostronnie regularną

- Usunąć ε - produkcje (w razie potrzeby) ;
- Usunąć reguły łańcuchowe ;
- Usunąć symbol początkowy z prawych stron produkcji (w razie potrzeby);
  /* algorytm usuwania symbolu początkowego będzie podany później */

Przykład:

$A \to abB$	$\mathbf{A \to aA_1}$		$A \to aA_1$		$A \to aA_1$
	$\mathbf{A_1 \to bB}$		$A_1 \to bB$		$A_1 \to bB$
$A \to ba$	$\mathbf{A \to bA_2}$		$A \to bA_2$		$A \to bA_2$
	$\mathbf{A_2 \to a}$		$A_2 \to a$		$A_2 \to a$
$B \to b$	$B \to b$		$B \to b$		$B \to b$
$B \to A$	$B \to A$		$B \to A$	$\Rightarrow$**	$\mathbf{B \to aA_1}$
					$\mathbf{B \to bA_2}$
$B \to \varepsilon$	$B \to \varepsilon$	$\Rightarrow$ *	$\mathbf{A_1 \to b}$		$A_1 \to b$
Gramatyka prawostronnie liniowa	* - usunięcie ε-produkcji		** - usunięcie produkcji łańcuchowych		Gramatyka prawostronnie regularna

Przekształcenie gramatyki lewostronnie regularnej w prawostronnie regularną

Wejście: $G = < V,\Sigma,P,S > \in \mathcal{G}_{LRG}$;   G – nie zawiera symbolu początkowego S w prawych stronach produkcji

Wyjście: $G' = < V',\Sigma,P',S'> \in \mathcal{G}_{PRG}$ taka, że $L(G') = L(G)$

Metoda:

$P' := \varnothing$;

$V' := V \cup \{S'\} - \{S\}$

<u>for</u> $(A\rightarrow a) \in P : a\in\Sigma$ <u>do</u>

    <u>if</u> A=S

        <u>then</u> $P' := P' \cup \{S'\rightarrow a\}$

        <u>else</u> $P' := P' \cup \{S'\rightarrow aA\}$;

<u>for</u> $(A\rightarrow Ba) \in P : B\in V , a\in\Sigma$ <u>do</u>

    <u>if</u> A=S

        <u>then</u> $P' := P' \cup \{B\rightarrow a\}$

        <u>else</u> $P' := P' \cup \{B\rightarrow aA\}$;

Przekształcenie gramatyki lewostronnie regularnej w prawostronnie regularną

<u>Przykład:</u>

$G=<\{S,A\},\{a,b\},P,S>$	$G'=<\{S',A\},\{a,b\},P',S'>$
$S \rightarrow a$	$S' \rightarrow a$
$S \rightarrow Ab$	$A \rightarrow b$
$A \rightarrow a$	$S' \rightarrow aA$
$A \rightarrow Ab$	$A \rightarrow bA$

Usuwanie produkcji końcowych (kosztem wprowadzenia ε-produkcji)

Produkcje końcowe: $U \to a$: $U \in V$, $a \in \Sigma$

Wejście: $G = < V,\Sigma,P,S > \in \mathcal{G}_{PRG}$

Wyjście: $G' = < V',\Sigma,P',S >$ - bez produkcji końcowych, taka że $L(G') = L(G)$

Metoda:

```
V' := V;
P' := P;
for (A → x) ∈ P do
    if x∈Σ then
    begin
        V' := V' ∪ { A_x };
        P' := P' ∪ { A → xA_x , A_x → ε }
                - { A → x };
    end;
```

Usuwanie produkcji końcowych (kosztem wprowadzenia ε-produkcji)

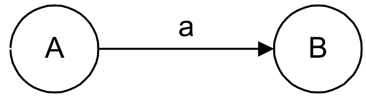
Przykład:

$G = < \{S,A,B,C,R,Q\}, \{a,b\}, P, S >$

$S \to bS$	$S \to bS$
$S \to aA$	$S \to aA$
$S \to aB$	$S \to aB$
$B \to bC$	$B \to bC$
$C \to aA$	$C \to aA$
$A \to bR$	$A \to bR$
$Q \to aB$	$Q \to aB$
$A \to b$	**$A \to b\underline{D}$, $\underline{D} \to \varepsilon$**

D – Symbol końcowy (nie mylić z symbolem terminalnym)

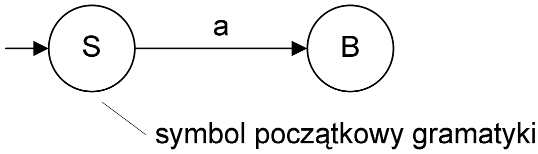
Wykres gramatyki (bez produkcji końcowych) w postaci grafu zorientowanego



$A \to aB$

$A,B \in V \quad a \in \Sigma$

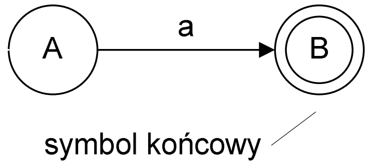
$A \neq S \quad (B \to \varepsilon) \notin P$



$S \to aB$

$a \in \Sigma,\ B \in V$

$(B \to \varepsilon) \notin P$



$A \to aB$

$B \to \varepsilon$

$a \in \Sigma;\ A,B \in V$

Przykład (1)

$S \to bS$

$S \to aA$

$S \to aB$

$B \to bC$

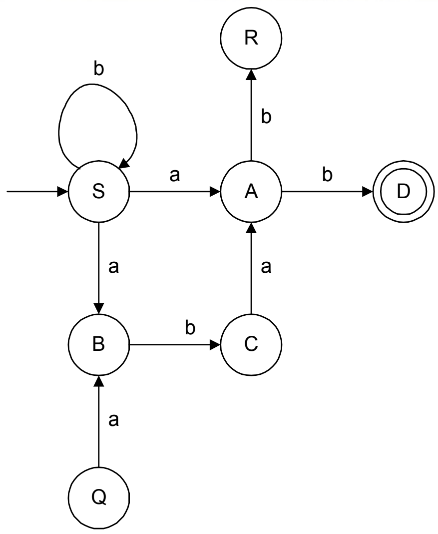
$C \to aA$

$A \to bR$

$Q \to aB$

$A \to bD$

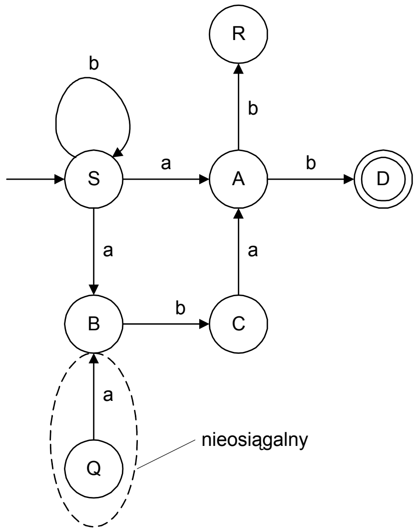
$D \to \varepsilon$



Przykład (2)

Usuwanie symboli nieosiągalnych

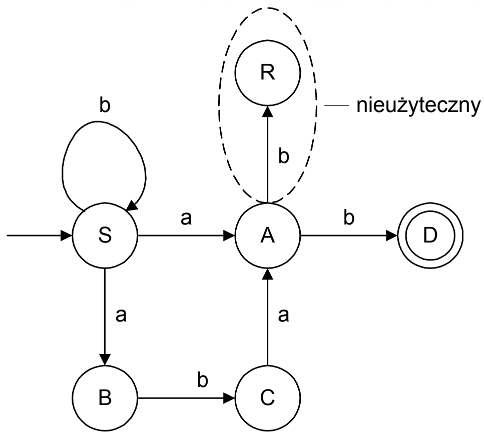
Można usunąć każdą produkcję U → aW, taką że U≠S oraz symbol U nie występuje po prawej stronie żadnej produkcji.



Przykład (3)

Usuwanie symboli nieużytecznych

Można usunąć wszystkie produkcje U → aW, gdzie W nie jest symbolem końcowym oraz W nie występuje po lewej stronie żadnej produkcji, z wyjątkiem być może produkcji typu W → aW.



Powyższe stwierdzenia nie są precyzyjne. Dokładne algorytmy podano dla gramatyk bezkontekstowych. ($\mathcal{G}_{RG} \subset \mathcal{G}_{BK}$)

Przypomnienie o ścieżkach w grafie skierowanym

Ścieżka – ciąg wierzchołków grafu zgodny z istniejącymi krawędziami i ich zorientowaniem.

Definicje: Ścieżka końcowa ⇔ ścieżka $K_0K_1 \dots K_n$ taka, że

$K_0 = S$

$K_n \in$ zbiór symboli końcowych

Ścieżka wyznaczona przez słowo $x=x_1x_2 \dots x_n$ ⇔ ścieżka końcowa $K_0K_1 \dots K_n$ taka, że

$(K_0 \rightarrow x_1K_1) \in P$

$(K_1 \rightarrow x_2K_2) \in P$

...

$(K_{n-1} \rightarrow x_nK_n) \in P$

$(K_n \rightarrow \varepsilon) \in P$

$x \in L(G) \Leftrightarrow \exists$ ścieżka wyznaczona przez słowo x.

Graf automatu skończonego ≡ graf gramatyki prawostronnie regularnej bez produkcji końcowych

Automat skończony ⇒ wyrażenie regularne

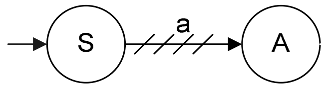
Twierdzenie: Język generowany przez gramatykę prawostronnie regularną bez produkcji końcowych (język akceptowany przez automat skończony) jest językiem regularnym (tzn. określonym przez wyrażenie regularne).

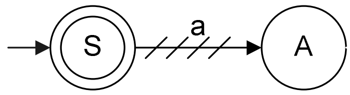
[Dowód przez indukcję względem liczby krawędzi w grafie].

Z gramatyki prawostronnie regularnej usuwamy jedną produkcję typu $S \rightarrow aA$ (S – symbol początkowy).

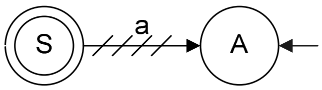
Automat skończony ⇒ wyrażenie regularne

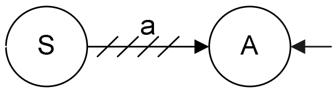
Rozważamy cztery gramatyki:

G_1: 

G_2: 

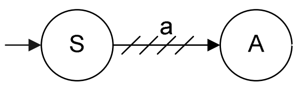
W G_2 wierzchołkiem początkowym
i <u>jedynym</u> końcowym jest S.

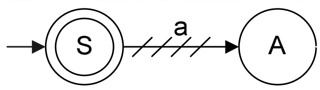
G_3: 

G_4: 

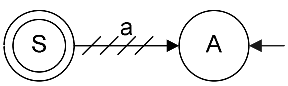
W G_3 wierzchołkiem początkowym jest A
zaś <u>jedynym</u> końcowym jest S

Automat skończony ⇒ wyrażenie regularne

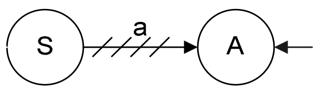
G_1: 

G_2: 

W G_2 wierzchołkiem początkowym
i <u>jedynym</u> końcowym jest S.

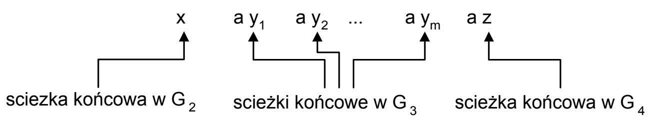
G_3: 

W G_3 wierzchołkiem początkowym jest A
zaś <u>jedynym</u> końcowym jest S

G_4: 

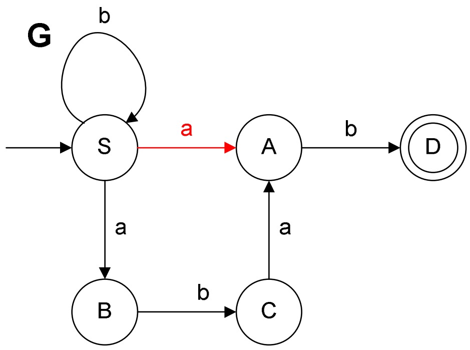
$$L(G) = L(G_1) \cup L(G_2)\ [\, aL(G_3)]^* aL(G_4)$$

Każda ścieżka końcowa w G jest albo końcowa w G_1 albo może być przedstawiona w postaci:

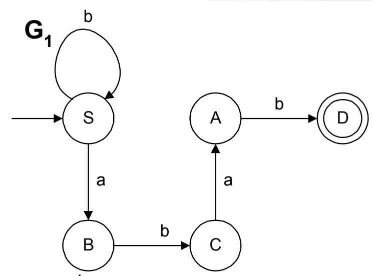


Automat skończony $\Rightarrow$ wyrażenie regularne Przykład (1)

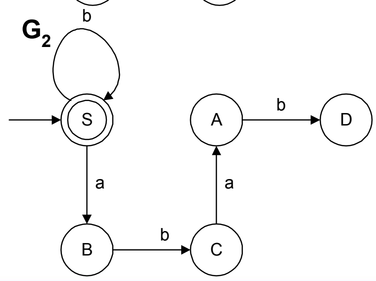
- Zbudować wyrażenie regularne opisujące język akceptowany przez automat skończony dany grafem (generowany przez gramatykę daną grafem):



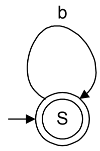
Automat skończony $\Rightarrow$ wyrażenie regularne Przykład (2)



$L(G_1)= \{b^n abab \mid n \geq 0\} =$ **b*abab**

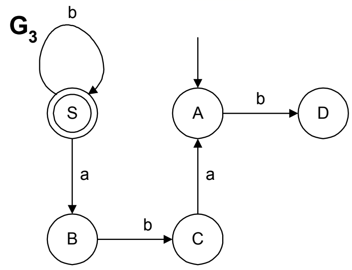


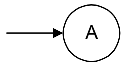
$\equiv$



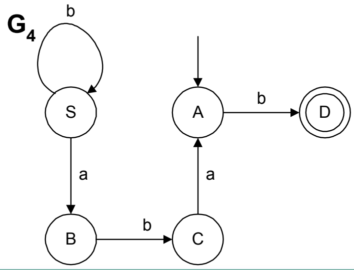
$L(G_2) = \{b^n | n \geq 0\} =$ **b***

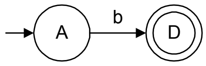
Automat skończony $\Rightarrow$ wyrażenie regularne
Przykład (3)



$\equiv$ 

$L(G_3) = \varnothing$



$\equiv$ 

$L(G_4)$ = {b}= **b**

Automat skończony $\Rightarrow$ wyrażenie regularne
Przykład (4)

$L(G_1)$= **b*abab**
$L(G_2)$ = **b***
$L(G_3) = \varnothing$
$L(G_4)$ = **b**

$L(G) = L(G_1) \cup L(G_2)\ [aL(G_3)]^*aL(G_4) =$
$= \mathbf{b^*abab\ |\ b^*(a\varnothing)^*ab}$

b*abab | b*(aφ)*ab =
= b*abab | b*φ*ab =
= b*abab | b*εab =
= b*abab | b*ab =
= b*ab(ab|ε)

L(G) = **b*ab(ab|ε)**

Usuwanie symbolu początkowego z prawych stron produkcji

We : $G = \langle V,\Sigma,P,S\rangle$ - gramatyka regularna bez produkcji końcowych

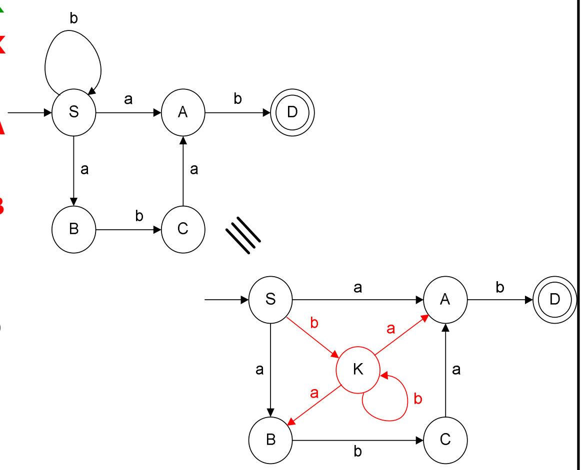
Wy : $G'=\langle V',\Sigma,P',S\rangle$ - gramatyka regularna bez S w prawych stronach produkcji, taka że $L(G') = L(G)$

Metoda:

$P_1 := P;$
$V' := V;$
<u>for</u> $(A\rightarrow aB) \in P$ <u>do</u>
    <u>if</u> $B=S$ <u>then</u>
    <u>begin</u>
        $V' := V' \cup \{K\};$
        $P_1 := P_1 - \{A \rightarrow aS\} \cup \{A \rightarrow aK\};$
    <u>end</u>;
$P' := P_1;$
<u>for</u> $(A \rightarrow X) \in P_1$ <u>and</u> $(X = aB$ <u>or</u> $X = \varepsilon)$ <u>do</u>
    <u>if</u> $A=S$ <u>then</u>
        $P' := P' \cup \{K \rightarrow X\};$

Usuwanie symbolu początkowego z prawych stron produkcji

$S \rightarrow bS$	**$S \rightarrow bK$**	**$S \rightarrow bK$**
		$K \rightarrow bK$
$S \rightarrow aA$	$S \rightarrow aA$	$S \rightarrow aA$
		$K \rightarrow aA$
$S \rightarrow aB$	$S \rightarrow aB$	$S \rightarrow aB$
		$K \rightarrow aB$
$B \rightarrow bC$	$B \rightarrow bC$	$B \rightarrow bC$
$C \rightarrow aA$	$C \rightarrow aA$	$C \rightarrow aA$
$A \rightarrow bD$	$A \rightarrow bD$	$A \rightarrow bD$
$D \rightarrow \varepsilon$	$D \rightarrow \varepsilon$	$D \rightarrow \varepsilon$



Konstrukcja sumy teoriomnogościowej, złożenia i domknięcia Kleene’go języków regularnych

Twierdzenie: L_1 i L_2 – języki regularne generowane przez gramatyki

$G_1 = <V_1,\Sigma_1,P_1,S_1>$

$G_2 = <V_2,\Sigma_2,P_2,S_2>$

Wówczas języki :

- $L_1 \cup L_2$
- L_1L_2
- L_1*

są regularne.

Konstrukcję gramatyki $G = <V,\Sigma,P,S>$, takiej, że:

a) $L(G) = L_1(G_1) \cup L_2(G_2)$

b) $L(G) = L_1(G_1)\ L_2(G_2)$

c) $L(G) = [L_1(G_1)]*$

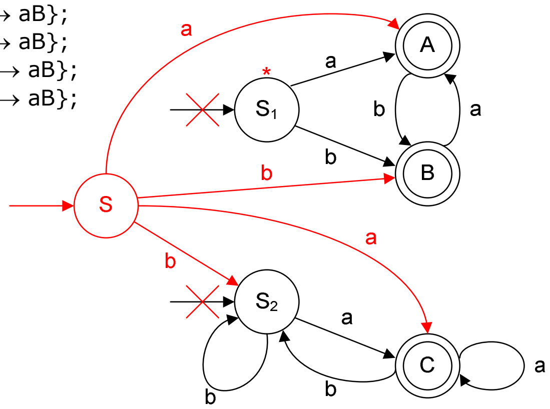
dokonujemy przy założeniach i oznaczeniach :

- $\Sigma = \Sigma_1 \cup \Sigma_2$
- $V_1 \cap V_2 = \varnothing$ (jeśli nie, to można nieterminale pomalować na różne kolory)
- G_1 i G_2 - gramatyki regularne bez produkcji końcowych
- F_1 i F_2 - zbiór nieterminalnych symboli końcowych gramatyk G1 i G2
- F - zbiór symboli końcowych gramatyki G

Konstrukcja sumy teoriomnogościowej, złożenia i domknięcia Kleene’go języków regularnych (a)

(a) Konstrukcja $G = <V_1 \cup V_2 \cup \{S\}, \Sigma, P, S>$ takiej, że $L(G) = L_1(G_1) \cup L_2(G_2)$

<u>if</u> $\varepsilon \notin L_1 \cup L_2$ <u>then</u> $F := F_1 \cup F_2$
<u>else</u> $F := F_1 \cup F_2 \cup \{S\}$;
$P := \varnothing$;
<u>for</u> $(A \to aB) \in P_1$ <u>do</u> $P := P \cup \{A \to aB\}$;
<u>for</u> $(A \to aB) \in P_2$ <u>do</u> $P := P \cup \{A \to aB\}$;
<u>for</u> $(S_1 \to aB) \in P_1$ <u>do</u> $P := P \cup \{S \to aB\}$;
<u>for</u> $(S_2 \to aB) \in P_2$ <u>do</u> $P := P \cup \{S \to aB\}$;
<u>for</u> $A \in F$ <u>do</u> $P := P \cup \{A \to \varepsilon\}$;



*S_1 stał się nieosiągalny

Konstrukcja sumy teoriomnogościowej, złożenia i domknięcia Kleene'go języków regularnych (b)

(b) Konstrukcja G = < $V_1 \cup V_2$, Σ, P, S_1> takiej, że L(G) = $L_1(G_1)L_2(G_2)$

<u>if</u> $S_2 \notin F_2$ <u>then</u> F:= F_2 <u>else</u> F:= $F_1 \cup F_2$;

P:=∅;

<u>for</u> $(A \to aB) \in P_1$ <u>and</u> $A \in V_1 - F_1$ <u>do</u>

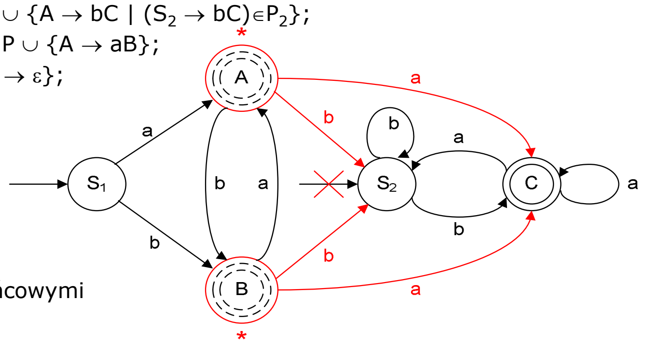
P:= P ∪ {A → aB };

<u>for</u> $(A \to aB) \in P_1$ <u>and</u> $A \in F_1$ <u>do</u>

P:= P ∪ {A → aB} ∪ {A → bC | $(S_2 \to bC) \in P_2$};

<u>for</u> $(A \to aB) \in P_2$ <u>do</u> P:= P ∪ {A → aB};

<u>for</u> A ∈ F <u>do</u> P:= P ∪ {A → ε};



*A oraz B przestają być końcowymi

Konstrukcja sumy teoriomnogościowej, złożenia i domknięcia Kleene'go języków regularnych (c)

(c) Konstrukcja G = <$V_1 \cup \{S\}$, Σ, P, S> takiej , że : L(G) = $[L_1(G_1)]^*$

F := $F_1 \cup \{S\}$;

P:= ∅;

<u>for</u> a∈Σ <u>and</u> $A \in V_1 - F_1$ <u>do</u>

P := P ∪ {A → aB | (A → aB) ∈ P_1};

<u>for</u> a∈Σ <u>and</u> $A \in F_1$ <u>do</u>

<u>begin</u>

P := P ∪ {A → aB | (A → aB) ∈ P_1};

P := P ∪ {A → aB | $(S_1 \to aB) \in P_1$};

<u>end</u>;

<u>for</u> a ∈ Σ <u>do</u>

P := P ∪ {S_1 → aB | (S → aB) ∈ P_1};

<u>for</u> A ∈ F <u>do</u> P := P ∪ {A → ε};