Automaty a gramatiky (BI-AAG)

5. Převody mezi reg. gram., reg. výrazy a kon. automaty

Jan Holub

Katedra teoretické informatiky

Fakulta informačních technologií

ČVUT v Praze

Vztah mezi reg. výrazy a kon. automaty

regulární výraz $\rightarrow$ konečný automat

- metoda sousedů, Glushkov
- metoda derivací, Janusz A. Brzozowski
- metoda postupnou konstrukcí, Ken Thompson

Vztah mezi reg. výrazy a kon. automaty

Algoritmus Konstrukce KA pro daný regulární výraz – metoda sousedů, Glushkov
Vstup: Regulární výraz V.
Výstup: Konečný automat $M = (Q, T, \delta, q_0, F)$, $h(V) = L(M)$.
Metoda:

1. Očíslujeme čísly $1, 2, \ldots, n$ všechny výskyty symbolů z T ve výrazu V. Vzniklý regulární výraz označíme V'.
2. Množina začátečních symbolů $Z = \{x_i : x \in T$, symbolem x_i může začínat nějaký řetězec z $h(V')\}$.
3. Množinu sousedů $P = \{x_i y_j :$ symboly x_i a y_j mohou být vedle sebe v nějakém řetězci z $h(V')\}$.
4. $Q = \{q_0\} \cup \{x_i : x \in T, i \in \langle 1, n \rangle\}$.
5. Množinu koncových symbolů $F = \{x_i :$ symbolem x_i může končit nějaký řetězec z $h(V)\} \cup \{q_0 : \varepsilon \in h(V)\}$.

Vztah mezi reg. výrazy a kon. automaty

Algoritmus (pokračování):

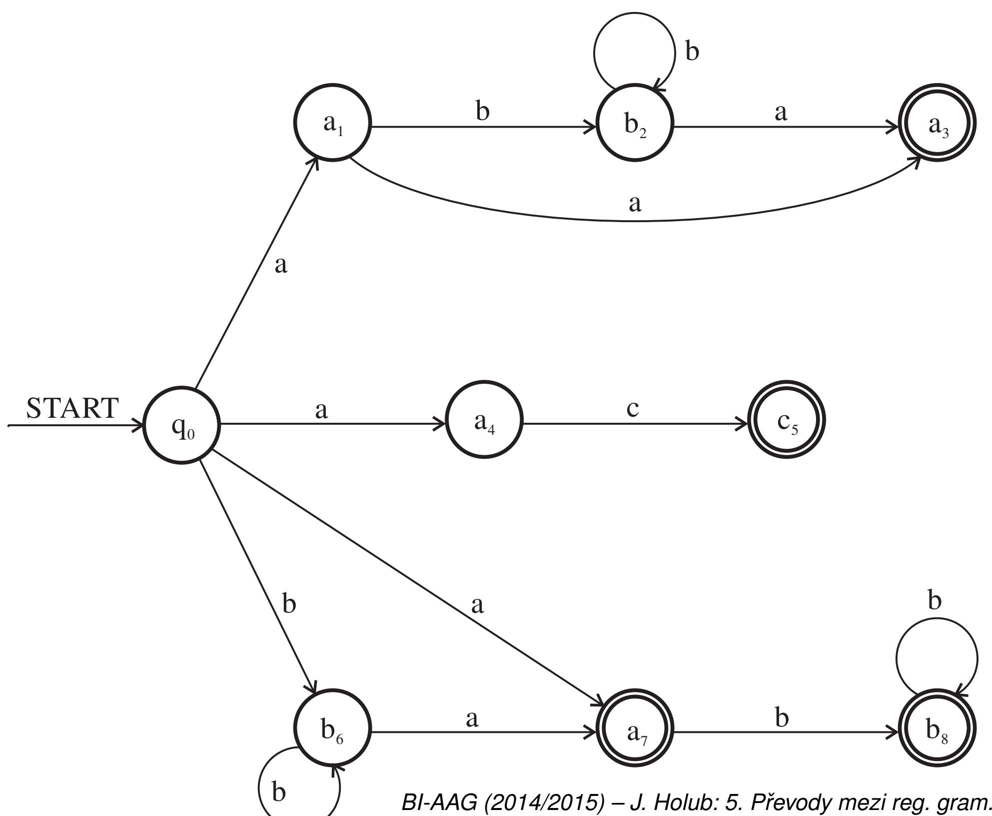
6. Zobrazení δ:
   (a) $\delta(q_0, x)$ obsahuje x_i, $\forall x_i \in Z$.
   (b) $\delta(x_i, y)$ obsahuje y_j, $\forall x_i y_j \in P$
7. Množina F je množinou koncových stavů. $\square$

Vztah mezi reg. výrazy a kon. automaty

Příklad

$V = ab^*a + ac + b^*ab^*, T = \{a, b, c\}.$

$V' = a_1 b_2^* a_3 + a_4 c_5 + b_6^* a_7 b_8^*, Z = \{a_1, a_4, b_6, a_7\}.$
$P = \{a_1 b_2, a_1 a_3, b_2 b_2, b_2 a_3, a_4 c_5, b_6 b_6, b_6 a_7, a_7 b_8, b_8 b_8\}.$
$F = \{a_3, c_5, a_7, b_8\}.$
$M = (\{q_0, a_1, b_2, a_3, a_4, c_5, b_6, a_7, b_8\}, \{a, b, c\}, \delta, q_0, \{a_3, c_5, a_7, b_8\})$



Vztah mezi reg. výrazy a kon. automaty

Algoritmus Konstrukce DKA pro daný regulární výraz – metoda derivací, Janusz A. Brzozowski.
Vstup: Regulární výraz V nad abecedou T.
Výstup: Konečný automat $M = (Q, T, \delta, q_0, F)$, $h(V) = L(M)$.
Metoda:

1. $Q = \{V\}, Q_0 = \{V\}, i := 1.$
2. $Q_i = \{\frac{dU}{da} : U \in Q_{i-1}, a \in T\} \setminus Q$.
3. Jestliže $Q_i \neq \emptyset$, $Q = Q \cup Q_i$, $i := i + 1$, jdi na krok 2.
   Jestliže $Q_i = \emptyset$, vytvoříme automat M takto:
   $M = (Q, T, \delta, V, F)$,
   $\delta(\frac{dU}{dx}, a) = \frac{dU}{d(xa)}$.
   Množina $F = \{\frac{dU}{dx} : \varepsilon \in h(\frac{dU}{dx})\}$. □

Vztah mezi reg. výrazy a kon. automaty

Příklad

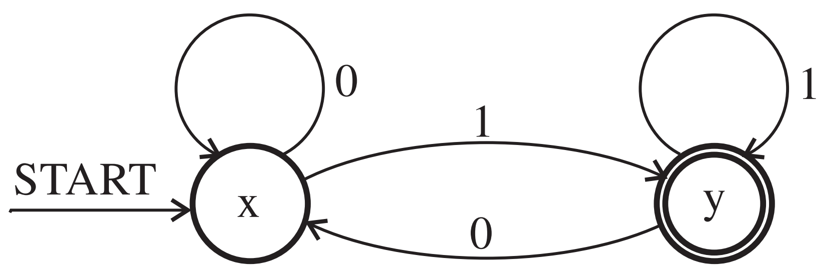
$(0+1)^*1$

1. Položíme $Q = \{(0+1)^*1\}, Q_0 = \{(0+1)^*1\}$.
2. Vypočteme Q_1:
   $\frac{d}{d1}((0+1)^*1) = (\emptyset + \varepsilon)(0+1)^*1 + \varepsilon = (0+1)^*1 + \varepsilon$
   $\frac{d}{d0}((0+1)^*1) = (\varepsilon + \emptyset)(0+1)^*1 + \emptyset = (0+1)^*1$.
   Protože $\frac{d}{d0}((0+1)^*1) = (0+1)^*1$,
   bude $Q_1 = \{(0+1)^*1 + \varepsilon\}$ a $Q = \{(0+1)^*1, (0+1)^*1 + \varepsilon\}$.
3. Vypočteme Q_2:
   $\frac{d}{d1}((0+1)^*1 + \varepsilon) = (\emptyset + \varepsilon)(0+1)^*1 + \varepsilon + \emptyset = (0+1)^*1 + \varepsilon$
   $\frac{d}{d0}((0+1)^*1 + \varepsilon) = (\varepsilon + \emptyset)(0+1)^*1 + \emptyset = (0+1)^*1$.
   Protože oba výrazy již jsou v množině Q, je množina Q_2 prázdná.

Vztah mezi reg. výrazy a kon. automaty

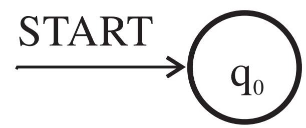
Příklad (pokračování)

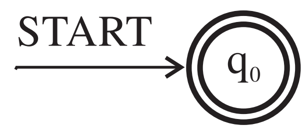
4. stav $x = (0+1)^*1$,
   stav $y = (0+1)^*1 + \varepsilon$,
   počáteční stav je x,
   koncový stav je y, protože $\varepsilon \in h((0+1)^*1 + \varepsilon)$
   $\frac{dx}{d1} = \mathrm{y}$, $\frac{dx}{d0} = x$, $\frac{dy}{d1} = y$, $\frac{dy}{d0} = x$,

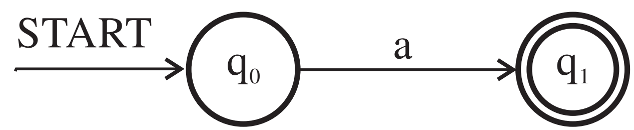


Vztah mezi reg. výrazy a kon. automaty

Algoritmus Konstrukce KA pro daný regulární výraz – postupná konstrukce, Ken Thompson.
Vstup: Regulární výraz V nad abecedou T.
Výstup: Konečný automat $M = (Q, T, \delta, q_0, F)$, $h(V) = L(M)$.
Metoda:

1. Sestrojíme KA pro elementární regulární výrazy:
   (a) Pro regulární výraz $V = \emptyset$:
   

   (b) Pro regulární výraz $V = \varepsilon$:
   

   (c) pro regulární výraz $V = a$:
   

2. Pro části regulárního výrazu tvaru $V = V_1 + V_2,\ V = V_1V_2,\ V = V_1^*$ sestrojíme konečné automaty pomocí příslušných algoritmů.

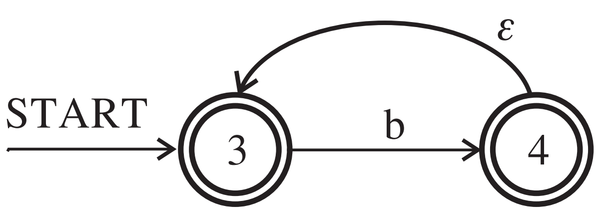
Vztah mezi reg. výrazy a kon. automaty

Příklad

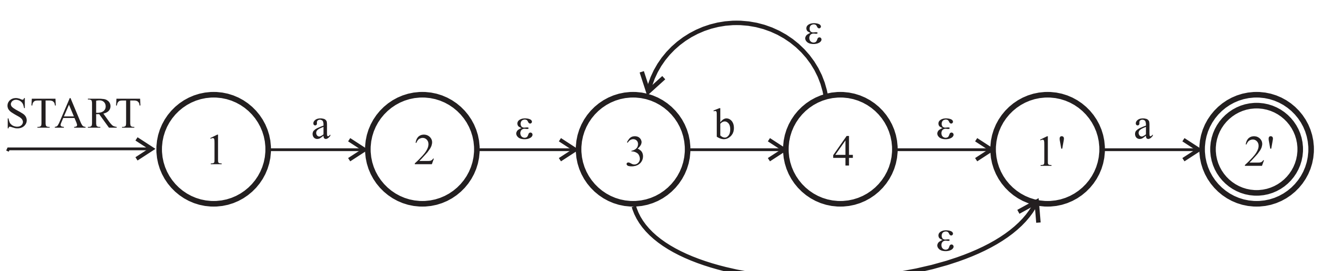
$V = ab^*a + ab$.

Potřebné elementární výrazy jsou:

Dále sestrojíme automat pro výraz b^*:



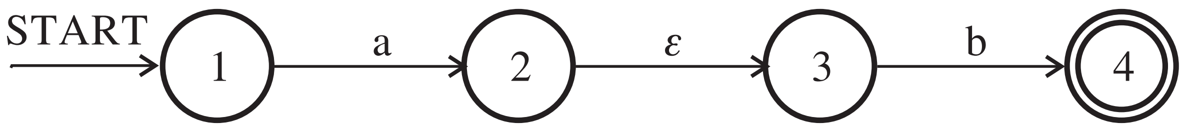
Pro výraz ab^*a má konečný automat tvar:



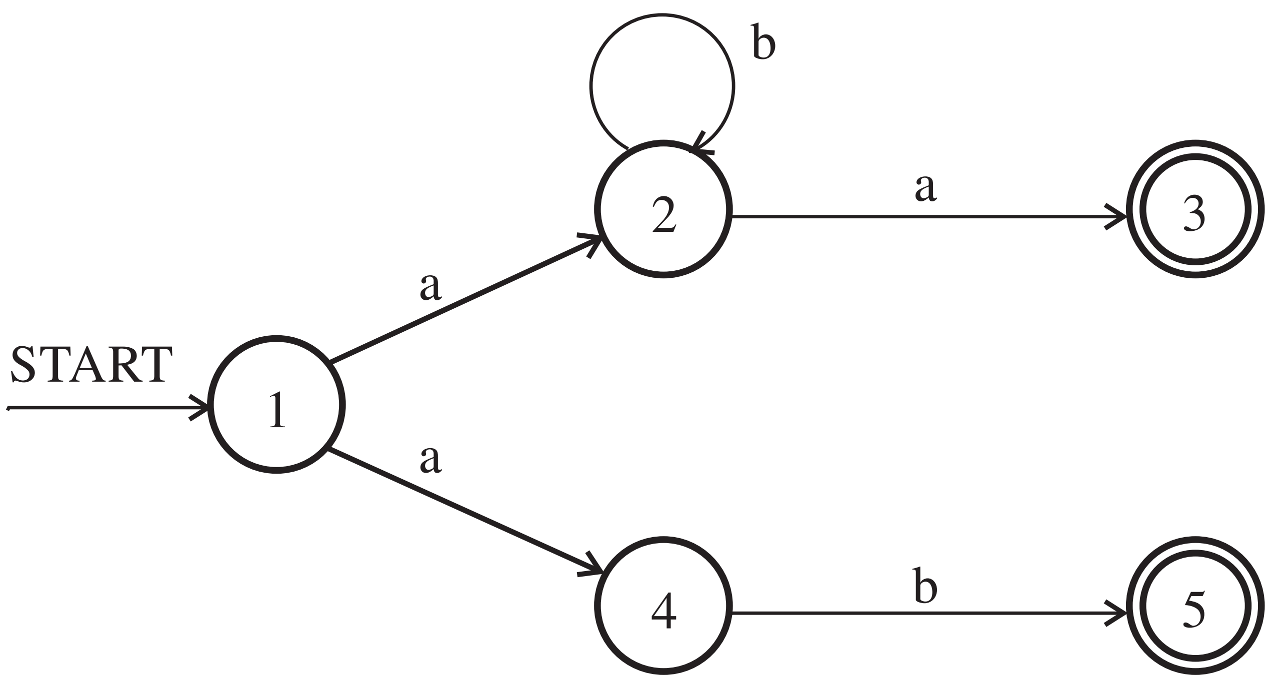
Vztah mezi reg. výrazy a kon. automaty

Příklad (pokračování)

Pro výraz ab má konečný automat tvar:



Pro výraz $V = ab^*a + ab$:



Vztah mezi kon. automaty a reg. výrazy

konečný automat $\rightarrow$ regulární výraz

- metoda eliminací stavů
- metoda regulárních rovnic – příchozí přechody
- metoda regulárních rovnic – odchozí přechody

Vztah mezi kon. automaty a reg. výrazy

Věta

Pro každý konečný automat M lze sestavit regulární výraz V takový, že $L(M) = h(V)$. $\square$

Vztah mezi kon. automaty a reg. výrazy

Definice

Rozšířený konečný automat $M = (Q, T, \gamma, q_0, F)$, kde
Q je konečná množina vnitřních stavů,
T je konečná vstupní abeceda,
γ je zobrazení z $Q \times Q$ do R_T,
$q_0 \in Q$ je počáteční stav,
$F \subseteq Q$ je množina koncových stavů.
$\gamma(p, q) = \emptyset$, když přechod z p do q není definován.

Definice

Jazyk přijímaný rozšířeným konečným automatem M je
$L(M) = \{x : x \in T^*, x = x_1 x_2 \dots x_n$ a existuje posloupnost stavů
$q_0, q_1, \dots, q_n,\ q_n \in F$ taková, že
$x_1 \in h(\gamma(q_0, q_1)),\ x_2 \in h(\gamma(q_1, q_2)), \dots,\ x_n \in h(\gamma(q_{n-1}, q_n))\}$.

Vztah mezi kon. automaty a reg. výrazy

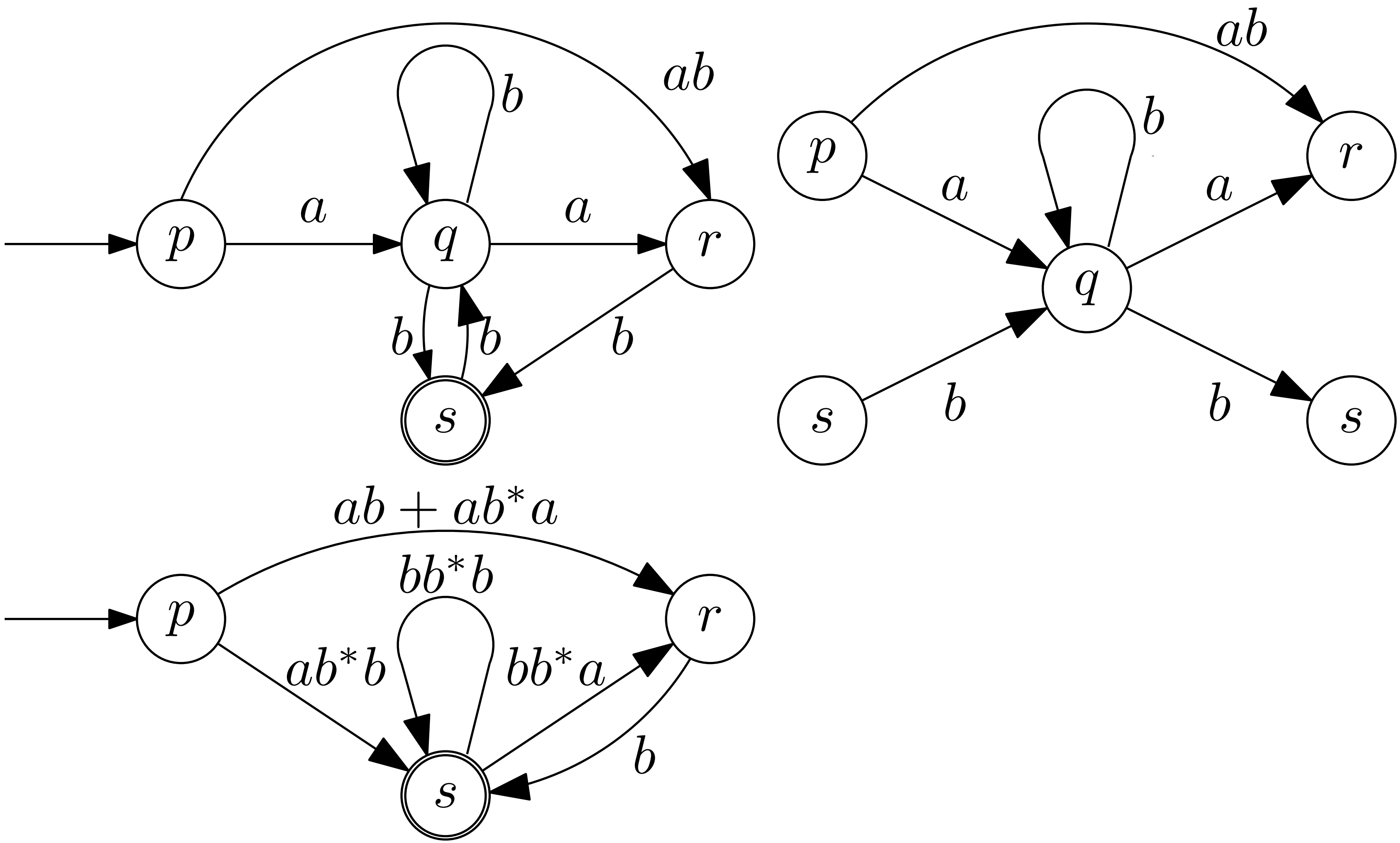
Věta

Nechť $M = (Q, T, \gamma, q_0, F)$ je rozšířený konečný automat. Předpokládejme, že q není ani počáteční ani koncový stav. Potom ekvivalentní rozšířený konečný automat $M' = (Q \setminus \{q\}, T, \gamma', q_0, F)$, kde zobrazení γ' je definováno pro každou dvojici $p, r \in (Q \setminus \{q\})$ takto:

$\gamma'(p, r) = \gamma(p, r) + \gamma(p, q)\gamma(q, q)^*\gamma(q, r)$. $\square$

Vztah mezi kon. automaty a reg. výrazy

Příklad



Vztah mezi kon. automaty a reg. výrazy

Druhý člen ve výrazu $\gamma'(p,r) = \gamma(p,r) + \gamma(p,q)\gamma(q,q)^*\gamma(q,r)$ se neuplatní tehdy, když $\gamma(p,q) = \emptyset$ nebo $\gamma(q,r) = \emptyset$.

Vztah mezi kon. automaty a reg. výrazy

Algoritmus Sestrojení regulárního výrazu pro daný KA – eliminace stavů.

Vstup: Konečný automat $M = (Q, T, \delta, q_0, F)$.

Výstup: Regulární výraz V takový, že $h(V) = L(M)$.

Metoda:

1. Sestrojíme $M_R = (Q, T, \gamma, q_0, F)$, kde $\gamma(p, q) = \{a\},\ a \in T \cup \{\varepsilon\}$, když $\delta(p, a)$ obsahuje q.
2. Automat M_R rozšíříme:
   (a) Jestliže počáteční stav $q_0 \in F$ nebo $\gamma(q, q_0) \neq \emptyset$, pak $Q = Q \cup \{q_0'\}$ a $\gamma(q_0', q_0) = \varepsilon$. q_0' bude počáteční stav.
   (b) Jestliže $|F| > 1$, pak $Q = Q \cup \{f\}$ a $\gamma'(q, f) = \varepsilon, \forall q \in F$. Množina konc. stavů $F' = \{f\}$.

   Po těchto úpravách bude $M_R' = (Q', T, \gamma', q_0', F')$.

Vztah mezi kon. automaty a reg. výrazy

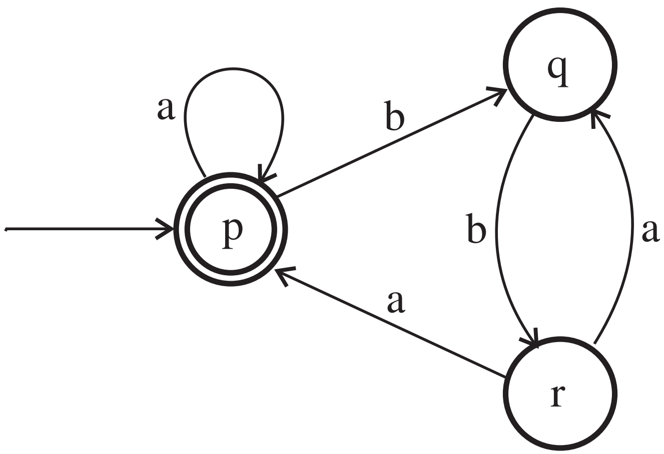
Algoritmus (pokračování):

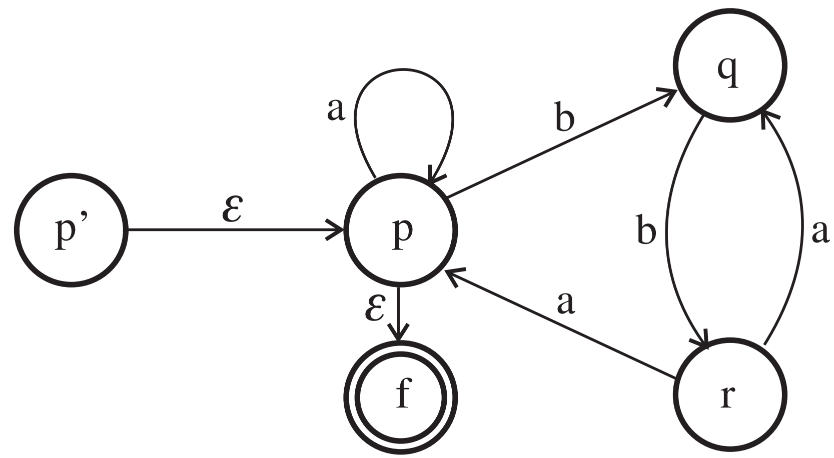
3. Jestliže $Q' = \{q'_0, f\}$, pak regulární výraz je $\gamma'(q'_0, f)\gamma'(f, f)^*$. Konec.
   Jinak pokračuj krokem 4.
4. Vyber $q \in Q', q \notin \{q'_0, f\}$. $Q' = Q' \setminus \{q\}$ a γ' jsou odpovídajícím způsobem upraveny. Pokračuj krokem 3.

Vztah mezi kon. automaty a reg. výrazy

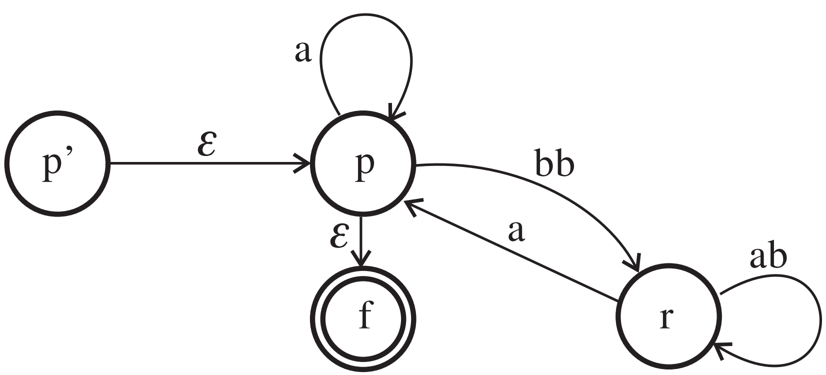
Příklad

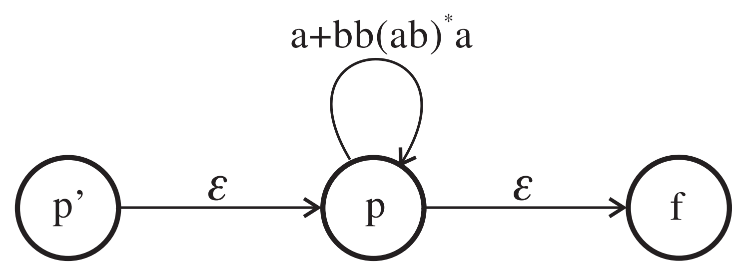
$M = (\{p, q, r\}, \{a, b\}, \delta, p, \{p\})$





Sestrojíme rozšířený konečný automat a přidáme nový počáteční a koncový stav. V dalším kroku vynecháme stavy q a r.

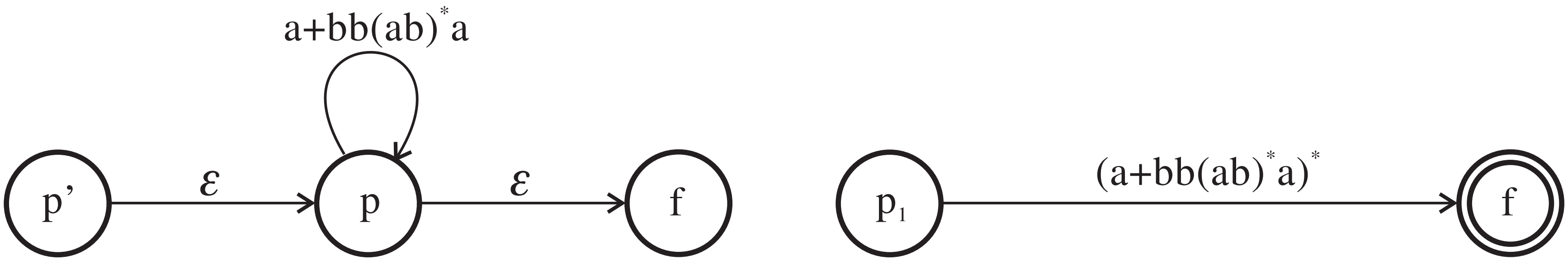




Vztah mezi kon. automaty a reg. výrazy

Příklad (pokračování)

Nakonec vynecháme stav p.



Výsledný regulární výraz je $V = (a + bb(ab)^*a)^*$.

Vztah mezi kon. automaty a reg. výrazy

Algoritmus Sestrojení reg. výrazu pro daný konečný automat – řešení regulárních rovnic, příchozí přechody

Vstup: Konečný automat $M = (Q, T, \delta, q_0, F)$.

Výstup: Regulární výraz V, $h(V) = L(M)$.

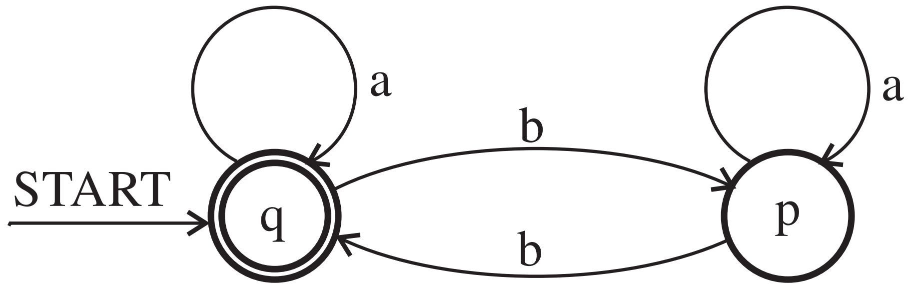
Metoda:

1. Pro každý stav q z Q:
   $X_q = X_{p_1}a_1 + X_{p_2}a_2 + \cdots + X_{p_n}a_n$, když $q \in \delta(p_i, a_i)$.
   V případě, že q je počáteční stav, přidáme ε.
2. Soustavu levých regulárních rovnic řešíme substituční metodou.
3. Výsledný regulární výraz:
   $V = X_{p_1} + X_{p_2} + ... + X_{p_n}$, jestliže $p_i \in F,\ i = 1, 2, ..., n$.

Vztah mezi kon. automaty a reg. výrazy

Příklad

$M = (\{q, p\}, \{a, b\}, \delta, q, \{q\})$



$$X_q = X_q a + X_p b + \varepsilon$$
$$X_p = X_p a + X_q b$$

$$X_p = X_q b a^*$$

$$X_q = X_q a + X_q b a^* b + \varepsilon = X_q(a + ba^*b) + \varepsilon$$

Výsledný regulární výraz: $V = X_q = (a + ba^*b)^*$

Vztah mezi kon. automaty a reg. výrazy

Algoritmus Sestrojení reg. výrazu pro daný konečný automat – řešení regulárních rovnic, odchozí přechody

Vstup: Konečný automat $M = (Q, T, \delta, q_0, F)$.

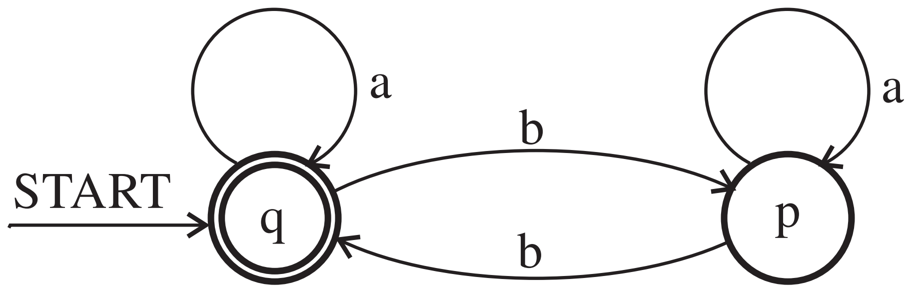
Výstup: Regulární výraz V, $h(V) = L(M)$

Metoda:

1. Pro každý stav q z Q:
   $X_q = a_1 X_{p_1} + a_2 X_{p_2} + ... + a_n X_{p_n}$, když $p_i \in \delta(q, a_i)$.
   V případě, že q je koncový stav, přidáme ε.
2. Vzniklou soustavu pravých regulárních rovnic řešíme substituční metodou.
3. Výsledný regulární výraz je výraz pro počáteční stav q_0 (pro proměnnou X_{q_0}).

Vztah mezi kon. automaty a reg. výrazy

Příklad



$X_q = aX_q + bX_p + \varepsilon$
$X_p = aX_p + bX_q$
Vyjádříme neznámou X_p: $X_p = a^*bX_q$
Dosadíme za X_p do první rovnice:
$X_q = aX_q + ba^*bX_q + \varepsilon$
$X_q = (a + ba^*b)X_q + \varepsilon$
Vyjádříme neznámou X_q
$X_q = (a + ba^*b)^* = V$

Vztah mezi reg. gram. a reg. výrazy

regulární gramatika $\rightarrow$ regulární výraz

- metoda eliminací neterminálních symbolů
- metoda regulárních rovnic

Vztah mezi reg. gram. a reg. výrazy

Věta

Pro každou regulární gramatiku $G = (N, T, P, S)$ lze sestrojit regulární výraz V takový, že $L(G) = h(V)$.

Definice

Rozšířená (pravá) regulární gramatika je čtveřice $G = (N, T, P, S)$, kde
N je konečná množina neterminálních symbolů,
T je konečná množina terminálních symbolů,
P je množina pravidel tvaru $A \rightarrow \alpha B$ nebo
$A \rightarrow \alpha,\ A, B \in N,\ \alpha \in R_T$,
$S \in N$ je počáteční symbol.

Vztah mezi reg. gram. a reg. výrazy

Pro jednoduchost budeme předpokládat, že v rozšířené regulární gramatice jsou nejvýše dvě pravidla tvaru $A \to \alpha B$ nebo $A \to \alpha$.
Úprava: Dvojici pravidel $A \to \alpha B,\ A \to \beta B$ nahradíme pravidlem $A \to (\alpha + \beta)B$.

Definice

Jazyk generovaný rozšířenou regulární gramatikou $L(G) = \{x : x \in T^*, x = x_1 x_2 \dots x_n$ a existuje derivace $S \Rightarrow \alpha_1 A_1 \Rightarrow \alpha_1 \alpha_2 A_2 \Rightarrow \dots \Rightarrow \alpha_1 \alpha_2 \dots \alpha_{n-1} A_{n-1} \Rightarrow \alpha_1 \alpha_2 \dots \alpha_{n-1} \alpha_n$ taková, že $x_i \in h(\alpha_i),\ 1 \leq i \leq n\}$.

Vztah mezi reg. gram. a reg. výrazy

Věta

Nechť $G = (N, T, P, S)$ je rozšířená regulární gramatika.
Předpokládejme, že neterminální symbol A není počáteční symbol.
Potom ekvivalentní rozšířená regulární gramatika
$G' = (N \setminus \{A\}, T, P', S)$, kde pravidla v P' vytvoříme takto:
Jsou-li v gramatice G pravidla:

$B \to \alpha_1 C$

$B \to \alpha_2 A$

$A \to \alpha_3 A$

$A \to \alpha_4 C$,

pak v gramatice G' budou v P' pravidla:

$B \to (\alpha_1 + \alpha_2 \alpha_3^* \alpha_4) C$.

Vztah mezi reg. gram. a reg. výrazy

Poznámka:
Pokud některé z uvedených pravidel se v gramatice G nevyskytuje, pak je nahrazeno pravidlem $X \to \emptyset Y$, kde $X \in \{A, B, C\},\ Y \in \{A, C\}$.

Odpovídající pravidla nebudou ani v G'.

Například pokud bude v G chybět pravidlo $A \to \alpha_4 C$, nahradíme jej pravidlem $A \to \emptyset C$ a potom pravidlo $B \to (\alpha_1 + \alpha_2 \alpha_3^* \emptyset) C$ degeneruje na pravidlo $B \to \alpha_1 C$.

Vztah mezi reg. gram. a reg. výrazy

Algoritmus Sestrojení regulárního výrazu pro danou regulární gramatiku – eliminace neterminálních symbolů.

Vstup: Pravá regulární gramatika $G = (N, T, P, S)$.

Výstup: Regulární výraz V takový, že $h(V) = L(G)$.

Metoda:

1. Rozšířená pravá regulární gramatika $G_R = (N, T, P_R, S)$ ekvivalentní G. Všechny n-tice pravidel tvaru $A \to \alpha_1 B, A \to \alpha_2 B, \ldots, A \to \alpha_n B$ nahradíme pravidlem $A \to (\alpha_1 + \alpha_2 + \ldots + \alpha_n)B$, kde $A \in N,\ B \in N \cup \{\varepsilon\}$.

Vztah mezi reg. gram. a reg. výrazy

Algoritmus (pokračování):

2. Gramatiku G_R rozšíříme dále takto:
   (a) nový počáteční neterminální symbol $S' \notin N$ a přidáme $S' \rightarrow S$,
   (b) nový koncový neterminální symbol $F \notin N$, všechna pravidla tvaru $A \rightarrow \alpha$ nahradíme pravidly $A \rightarrow \alpha F$ a přidáme $F \rightarrow \varepsilon$.

   $G'_R = (N', T, P'_R, S')$, kde $N' = N \cup \{S', F\}$,

   $$P'_R = \{A \rightarrow \alpha B : A \rightarrow \alpha B \in P_R\} \cup \{A \rightarrow \alpha F : A \rightarrow \alpha \in P_R\} \cup \{S' \rightarrow S,\ F \rightarrow \varepsilon\}.$$

3. Jestliže $N' = \{S'\}$ a $P'_R = \{S' \rightarrow \alpha F, F \rightarrow \varepsilon\}$, pak $V = \alpha$ a algoritmus končí. Jinak se pokračuje krokem 4.
4. Vybereme $A \in N'$ takový, že $A \notin S'$. $N' = N' \setminus \{A\}$ a pravidla P_R jsou odpovídajícím způsobem upravena. Pokračujeme krokem 3. $\square$

Vztah mezi reg. gram. a reg. výrazy

Příklad

$G = (\{S, A, B\}, \{a, b\}, P, S)$, kde P:

$$S \rightarrow bA \mid aS \mid \varepsilon$$

$$A \rightarrow bB$$

$$B \rightarrow aA \mid aS.$$

$S' \rightarrow S$, S' nový neterminální symbol

$S \rightarrow \varepsilon$ nahradíme pravidlem $S \rightarrow F$ a pro F doplníme pravidlo $F \rightarrow \varepsilon$.

$G'_R = (\{S', S, A, B, F\}, \{a, b\}, P', S')$:

$$S' \rightarrow S$$

$$S \rightarrow bA \mid aS \mid F$$

$$A \rightarrow bB$$

$$B \rightarrow aA \mid aS$$

$$F \rightarrow \varepsilon.$$

Vztah mezi reg. gram. a reg. výrazy

Příklad (pokračování)

Vyloučíme symbol A. $G_R^1 = (\{S', S, B, F\}, \{a, b\}, P^1, S')$:

$$
\begin{aligned}
S' &\rightarrow S \\
S &\rightarrow bbB \mid aS \mid F \\
B &\rightarrow abB \mid aS \\
F &\rightarrow \varepsilon.
\end{aligned}
$$

Vyloučíme symbol B. $G_R^2 = (\{S', S, F\}, \{a, b\}, P^2, S')$:

$$
\begin{aligned}
S' &\rightarrow S \\
S &\rightarrow (a + bb(ab)^*a)S \mid F \\
F &\rightarrow \varepsilon.
\end{aligned}
$$

Vyloučíme symbol S.
$$
\begin{aligned}
S' &\rightarrow (a + bb(ab)^*a)^* \mid F \\
F &\rightarrow \varepsilon.
\end{aligned}
$$

$$V = (a + bb(ab)^*a)^*$$

Vztah mezi reg. gram. a reg. výrazy

Algoritmus Sestrojení regulárního výrazu pro danou regulární gramatiku

Vstup: Regulární gramatika $G = (N, T, P, S)$.

Výstup: Regulární výraz V takový, že $h(V) = L(G)$.

Metoda:

1. Pro každý neterminální symbol z N sestrojíme regulární rovnici.
2. Vzniklou soustavu regulárních rovnic řešíme substituční metodou pro počáteční symbol gramatiky S. $\square$

Vztah mezi reg. gram. a reg. výrazy

Příklad

$G = (\{S, A\}, \{0, 1\}, P, S)$, P:

$$S \to 0S \mid 1A \mid 1$$
$$A \to 1S \mid 0A \mid 0$$

Soustava regulárních rovnic má tvar:

$$S = 0S + 1A + 1$$
$$A = 1S + 0A + 0$$

Soustavu vyřešíme eliminací:

$$S = 0^*(1A + 1) = 0^*1(A + \varepsilon)$$
$$A = 10^*1A + 10^*1 + 0A + 0$$
$$A = (10^*1 + 0)A + 10^*1 + 0$$
$$A = (10^*1 + 0)^*(10^*1 + 0)$$
$$S = 0^*1((10^*1 + 0)^*(10^*1 + 0) + \varepsilon) = 0^*1(10^*1 + 0)^*$$

Výsledný regulární výraz popisující jazyk $L(G)$ má tvar:

$$S = 0^*1(10^*1 + 0)^*$$

Vztah mezi reg. výrazy a reg. gram.

regulární výraz $\rightarrow$ regulární gramatika

- metoda postupnou konstrukcí
- metoda derivací

Vztah mezi reg. výrazy a reg. gram.

Věta

Pro každý regulární výraz V lze sestrojit regulární gramatiku G takovou, že $L(G) = h(V)$.

Vztah mezi reg. výrazy a reg. gram.

Algoritmus Konstrukce pravé regulární gramatiky pro daný regulární výraz – postupná konstrukce.

Vstup: Regulární výraz V nad abecedou T.

Výstup: $G = (N, T, P, S)$, $L(G) = h(V)$

Metoda:

1. $\forall a \in T$ sestrojíme gramatiky:
   $G_a = (\{A\}, \{a\}, \{A \to a\}, A)$.
2. Sestrojíme gramatiku $G_\varepsilon = (\{E\}, \emptyset, \{E \to \varepsilon\}, E)$.
3. Gramatiky pro podvýrazy tvaru $x_1 + x_2$, $x_1 x_2$, x_1^*, pokud jsme již sestrojili gramatiky pro podvýrazy x_1, x_2.
4. Vyloučíme ε-pravidla a jednoduchá pravidla.

Vztah mezi reg. výrazy a reg. gram.

Příklad

$(ab+\varepsilon)^*$

1. $G_a = (\{A\}, \{a\}, \{A \to a\}, A)$
   $G_b = (\{B\}, \{b\}, \{B \to b\}, B)$.
2. $G_\varepsilon = (\{E\}, \emptyset, \{E \to \varepsilon\}, E)$.
3. Postupně sestrojíme gramatiky pro jazyky, které jsou hodnotou výrazů $ab, ab+\varepsilon, (ab+\varepsilon)^*$:
   $G(ab) = (\{A, B\}, \{a, b\}, \{A \to aB, B \to b\}, A)$
   $G(ab+\varepsilon) = (\{N_1, A, B, E\}, \{a, b\}, P_1, N_1)$, kde P_1:
   $N_1 \to A \mid E$, $E \to \varepsilon$, $A \to aB$, $B \to b$.
   $G((ab+\varepsilon)^*) = (\{N_2, N_1, A, B, E\}, \{a, b\}, P_2, N_2)$, kde P_2:
   $N_2 \to N_1 \mid \varepsilon$, $N_1 \to A \mid E$, $E \to N_2$, $A \to aB$, $B \to bN_2$.

Vztah mezi reg. výrazy a reg. gram.

Příklad (pokračování)

4. Vyloučíme ε-pravidla $\left(N_\varepsilon = \{N_2, E, N_1\}\right)$:
   $N_2' \to N_2 \mid \varepsilon$, $N_2 \to N_1$, $N_1 \to A \mid E$, $E \to N_2$, $A \to aB$,
   $B \to bN_2 \mid b$.
   Dále vyloučíme jednoduchá pravidla:
   $N_{N2'} = \{N_2', N_2, N_1, A, E\}$, $N_{N2} = \{N_2, N_1, A, E\}$,
   $N_{N1} = \{A, E, N_2, N_1\}$, $N_E = \{E, N_2, N_1, A\}$, $N_A = \{A\}$,
   $N_B = \{B\}$ a dostaneme pravidla:
   $N_2' \to \varepsilon \mid aB$, $N_2 \to aB$, $N_1 \to aB$, $E \to aB$, $A \to aB$,
   $B \to bN_2 \mid b$.

Symboly N_1, E, A jsou zbytečné:
$G = (\{N_2', N_2, B\}, \{a, b\}, P, N_2')$, kde P:
$N_2' \to \varepsilon \mid aB$, $B \to bN_2 \mid b$, $N_2 \to aB$

Vztah mezi reg. výrazy a reg. gram.

Algoritmus Konstrukce pravé regulární gramatiky pro daný regulární výraz – metoda derivací.

Vstup: Regulární výraz V nad abecedou T.

Výstup: $G = (N, T, P, S)$, $L(G) = h(V)$.

Metoda:

1. $N = \{V\}, N_0 = \{V\}, i = 1$.
2. Vytvoříme derivace všech výrazů z N_{i-1} podle všech symbolů z abecedy T. Do množiny N_i vložíme všechny výrazy vzniklé derivací výrazů z N_{i-1}.
3. Jestliže $N_i \neq \emptyset$, přidáme N_i do N, $i = i + 1$ a přejdeme na krok 2.

   Jinak vytvoříme gramatiku $G = (N, T, P, V)$, kde P:

   $\frac{dV}{dx} \to a\frac{dV}{d(xa)}$ přidáme do P.

   $\frac{dV}{dx} \to a$ přidáme do P v případě, že $\varepsilon \in h\left(\frac{dV}{d(xa)}\right)$.

   $V \to \varepsilon$ přidáme do P v případě, že $\varepsilon \in h(V)$.

Vztah mezi reg. výrazy a reg. gram.

Příklad

$(ab + \varepsilon)^*$

1. $N = \{(ab + \varepsilon)^*\}, N_0 = \{(ab + \varepsilon)^*\}, i = 1.$
2. $\frac{d(ab+\varepsilon)^*}{da}$ =$(b + \emptyset)(ab + \varepsilon)^* = b(ab + \varepsilon)^*$
   $\frac{d(ab+\varepsilon)^*}{db}$ = $(\emptyset + \emptyset)(ab + \varepsilon)^* = \emptyset$
   $N_1 = \{b(ab + \varepsilon)^*\}, N = \{(ab + \varepsilon)^*, b(ab + \varepsilon)^*\}.$
3. $\frac{d(b(ab+\varepsilon)^*)}{da}$ = $\emptyset$
   $\frac{d(b(ab+\varepsilon)^*)}{db}$ = $\varepsilon(ab + \varepsilon)^* = (ab + \varepsilon)^*$
   $N_2 = \emptyset, N = \{(ab + \varepsilon)^*, b(ab + \varepsilon)^*\}$

Označíme-li $A = (ab + \varepsilon)^*, B = b(ab + \varepsilon)^*$, dostaneme gramatiku $G = (\{A, B\}, \{a, b\}, P, A)$, kde P obsahuje pravidla:

$$A \rightarrow aB \mid \varepsilon$$
$$B \rightarrow bA \mid b$$