Jazyky typu 1 v Chomského hierarchii

Teorie jazyků a automatů II, přednáška č. 9

Šárka Vavrečková

Ústav informatiky, FPF SU Opava

Poslední aktualizace: 7. prosince 2016

Obsah

Gramatiky generující jazyky typu 1

Definice (Nezkracující gramatika)

Nezkracující gramatika je taková gramatika, jejíž pravidla jsou ve tvaru

$$\alpha \to \beta, \text{ kde } |\alpha| \leq |\beta|, \ \alpha \in (N \cup T)^* N (N \cup T)^*, \ \beta \in (N \cup T)*$$

Je přípustné také pravidlo $S \to \varepsilon$, pokud ε je slovo jazyka, který gramatika generuje, S se nesmí vyskytovat na pravé straně žádného pravidla.

Gramatiky generující jazyky typu 1

Definice (Kontextová gramatika)

Kontextová gramatika je gramatika s pravidly ve tvaru

$$\alpha A\beta \rightarrow \alpha\gamma\beta, \text{ kde } |\gamma| > 0, \ \alpha, \beta, \gamma \in (N \cup T)^*$$

Je přípustné také pravidlo $S \rightarrow \varepsilon$, pokud ε je slovo jazyka, který gramatika generuje, S se nesmí vyskytovat na pravé straně žádného pravidla.

Je zřejmé, že každá kontextová gramatika je zároveň gramatikou nezkracující.

Gramatiky generující jazyky typu 1

Definice (Kontextová gramatika)

Kontextová gramatika je gramatika s pravidly ve tvaru

$$\alpha A\beta \rightarrow \alpha\gamma\beta,\ \text{kde } |\gamma| > 0,\ \alpha, \beta, \gamma \in (N \cup T)^*$$

Je přípustné také pravidlo $S \rightarrow \varepsilon$, pokud ε je slovo jazyka, který gramatika generuje, S se nesmí vyskytovat na pravé straně žádného pravidla.

Je zřejmé, že každá kontextová gramatika je zároveň gramatikou nezkracující.

Vztah mezi typy gramatik

Věta

Ke každé nezkracující gramatice G lze sestrojit kontextovou gramatiku G' takovou, že $L(G') = L(G)$.

Myšlenka důkazu

Každé pravidlo typu

$$A_1 A_2 \ldots A_m \to B_1 B_2 \ldots B_n,$$

které nemá kontextový tvar, nahradíme množinou pravidel:

$\boxed{A_1}\ A_2\ A_3 \ldots A_m \to \boxed{C_1}\ A_2\ A_3 \ldots A_m$

$C_1\ \boxed{A_2}\ A_3 \ldots A_m \to C_1\ \boxed{C_2}\ A_3 \ldots A_m$

$C_1\ C_2\ \boxed{A_3} \ldots A_m \to C_1\ C_2\ \boxed{C_3} \ldots A_m$

$\ldots$

$C_1\ C_2 \ldots C_{m-1}\ \boxed{A_m} \to C_1\ C_2 \ldots C_{m-1}\ \boxed{C_m\ B_{m+1} \ldots B_n}$

$C_1\ C_2 \ldots C_{m-1} \boxed{C_m} B_{m+1} \ldots B_n \to C_1 \ldots C_{m-1} \boxed{B_m} \ldots B_n$

$C_1\ C_2 \ldots \boxed{C_{m-1}} B_m\ B_{m+1} \ldots B_n \to C_1 \ldots \boxed{B_{m-1}} B_m \ldots B_n$

$\ldots$

$\boxed{C_1}\ B_2 \ldots B_n \to \boxed{B_1} \ldots B_n$

Kde C_i jsou nově přidané neterminály, které se jinde nevyskytují.

Myšlenka důkazu

Každé pravidlo typu

$$A_1 A_2 \dots A_m \to B_1 B_2 \dots B_n,$$

které nemá kontextový tvar, nahradíme množinou pravidel:

$\boxed{A_1}\ A_2\ A_3 \dots A_m \to \boxed{C_1}\ A_2\ A_3 \dots A_m$

$C_1\ \boxed{A_2}\ A_3 \dots A_m \to C_1\ \boxed{C_2}\ A_3 \dots A_m$

$C_1\ C_2\ \boxed{A_3}\ \dots A_m \to C_1\ C_2\ \boxed{C_3}\ \dots A_m$

$\dots$

$C_1\ C_2 \dots C_{m-1}\ \boxed{A_m} \to C_1\ C_2 \dots C_{m-1}\ \boxed{C_m\ B_{m+1} \dots B_n}$

$C_1\ C_2 \dots C_{m-1} \boxed{C_m} B_{m+1} \dots B_n \to C_1 \dots C_{m-1} \boxed{B_m} \dots B_n$

$C_1\ C_2 \dots \boxed{C_{m-1}} B_m\ B_{m+1} \dots B_n \to C_1 \dots\ \boxed{B_{m-1}} B_m \dots B_n$

$\dots$

$\boxed{C_1}\ B_2 \dots B_n \to \boxed{B_1} \dots B_n$

Kde C_i jsou nově přidané neterminály, které se jinde nevyskytují.

Myšlenka důkazu

Každé pravidlo typu

$$A_1 A_2 \dots A_m \to B_1 B_2 \dots B_n,$$

které nemá kontextový tvar, nahradíme množinou pravidel:

$\boxed{A_1}\ A_2\ A_3 \dots A_m \to \boxed{C_1}\ A_2\ A_3 \dots A_m$

$C_1\ \boxed{A_2}\ A_3 \dots A_m \to C_1\ \boxed{C_2}\ A_3 \dots A_m$

$C_1\ C_2\ \boxed{A_3}\ \dots A_m \to C_1\ C_2\ \boxed{C_3}\ \dots A_m$

$\dots$

$C_1\ C_2 \dots C_{m-1}\ \boxed{A_m} \to C_1\ C_2 \dots C_{m-1}\ \boxed{C_m\ B_{m+1} \dots B_n}$

$C_1\ C_2 \dots C_{m-1} \boxed{C_m} B_{m+1} \dots B_n \to C_1 \dots C_{m-1} \boxed{B_m} \dots B_n$

$C_1\ C_2 \dots \boxed{C_{m-1}} B_m\ B_{m+1} \dots B_n \to C_1 \dots\ \boxed{B_{m-1}} B_m \dots B_n$

$\dots$

$\boxed{C_1}\ B_2 \dots B_n \to \boxed{B_1} \dots B_n$

Kde C_i jsou nově přidané neterminály, které se jinde nevyskytují.

Myšlenka důkazu

Každé pravidlo typu

$$A_1A_2\dots A_m \to B_1B_2\dots B_n,$$

které nemá kontextový tvar, nahradíme množinou pravidel:

$\boxed{A_1}\ A_2\ A_3\dots A_m \to \boxed{C_1}\ A_2\ A_3\dots A_m$

$C_1\ \boxed{A_2}\ A_3\dots A_m \to C_1\ \boxed{C_2}\ A_3\dots A_m$

$C_1\ C_2\ \boxed{A_3}\ \dots A_m \to C_1\ C_2\ \boxed{C_3}\ \dots A_m$

$\dots$

$C_1\ C_2\dots C_{m-1}\ \boxed{A_m} \to C_1\ C_2\dots C_{m-1}\ \boxed{C_m\ B_{m+1}\dots B_n}$

$C_1\ C_2\dots C_{m-1}\boxed{C_m}B_{m+1}\dots B_n \to C_1\dots C_{m-1}\boxed{B_m}\dots B_n$

$C_1\ C_2\dots\boxed{C_{m-1}}B_m\ B_{m+1}\dots B_n \to C_1\dots\ \boxed{B_{m-1}}B_m\dots B_n$

$\dots$

$\boxed{C_1}\ B_2\dots B_n \to \boxed{B_1}\dots B_n$

Kde C_i jsou nově přidané neterminály, které se jinde nevyskytují.

Myšlenka důkazu

Každé pravidlo typu

$$A_1A_2\dots A_m \to B_1B_2\dots B_n,$$

které nemá kontextový tvar, nahradíme množinou pravidel:

$\boxed{A_1}\ A_2\ A_3\dots A_m \to \boxed{C_1}\ A_2\ A_3\dots A_m$

$C_1\ \boxed{A_2}\ A_3\dots A_m \to C_1\ \boxed{C_2}\ A_3\dots A_m$

$C_1\ C_2\ \boxed{A_3}\ \dots A_m \to C_1\ C_2\ \boxed{C_3}\ \dots A_m$

...

$C_1\ C_2\dots C_{m-1}\ \boxed{A_m} \to C_1\ C_2\dots C_{m-1}\ \boxed{C_m\ B_{m+1}\dots B_n}$

$C_1\ C_2\dots C_{m-1}\boxed{C_m}B_{m+1}\dots B_n \to C_1\dots C_{m-1}\boxed{B_m}\dots B_n$

$C_1\ C_2\dots\boxed{C_{m-1}}B_m\ B_{m+1}\dots B_n \to C_1\dots\ \boxed{B_{m-1}}B_m\dots B_n$

...

$\boxed{C_1}\ B_2\dots B_n \to \boxed{B_1}\dots B_n$

Kde C_i jsou nově přidané neterminály, které se jinde nevyskytují.

Myšlenka důkazu

Každé pravidlo typu

$$A_1 A_2 \ldots A_m \rightarrow B_1 B_2 \ldots B_n,$$

které nemá kontextový tvar, nahradíme množinou pravidel:

$\boxed{A_1}\ A_2\ A_3 \ldots A_m \rightarrow \boxed{C_1}\ A_2\ A_3 \ldots A_m$

$C_1\ \boxed{A_2}\ A_3 \ldots A_m \rightarrow C_1\ \boxed{C_2}\ A_3 \ldots A_m$

$C_1\ C_2\ \boxed{A_3} \ldots A_m \rightarrow C_1\ C_2\ \boxed{C_3} \ldots A_m$

$\ldots$

$C_1\ C_2 \ldots C_{m-1}\ \boxed{A_m} \rightarrow C_1\ C_2 \ldots C_{m-1}\ \boxed{C_m\ B_{m+1} \ldots B_n}$

$C_1\ C_2 \ldots C_{m-1} \boxed{C_m} B_{m+1} \ldots B_n \rightarrow C_1 \ldots C_{m-1} \boxed{B_m} \ldots B_n$

$C_1\ C_2 \ldots \boxed{C_{m-1}} B_m\ B_{m+1} \ldots B_n \rightarrow C_1 \ldots\ \boxed{B_{m-1}} B_m \ldots B_n$

$\ldots$

$\boxed{C_1}\ B_2 \ldots B_n \rightarrow \boxed{B_1} \ldots B_n$

Kde C_i jsou nově přidané neterminály, které se jinde nevyskytují.

Myšlenka důkazu

Každé pravidlo typu

$$A_1 A_2 \dots A_m \to B_1 B_2 \dots B_n,$$

které nemá kontextový tvar, nahradíme množinou pravidel:

$\boxed{A_1}\ A_2\ A_3 \dots A_m \to \boxed{C_1}\ A_2\ A_3 \dots A_m$

$C_1\ \boxed{A_2}\ A_3 \dots A_m \to C_1\ \boxed{C_2}\ A_3 \dots A_m$

$C_1\ C_2\ \boxed{A_3}\ \dots A_m \to C_1\ C_2\ \boxed{C_3}\ \dots A_m$

$\dots$

$C_1\ C_2 \dots C_{m-1}\ \boxed{A_m} \to C_1\ C_2 \dots C_{m-1}\ \boxed{C_m\ B_{m+1} \dots B_n}$

$C_1\ C_2 \dots C_{m-1} \boxed{C_m} B_{m+1} \dots B_n \to C_1 \dots C_{m-1} \boxed{B_m} \dots B_n$

$C_1\ C_2 \dots \boxed{C_{m-1}} B_m\ B_{m+1} \dots B_n \to C_1 \dots\ \boxed{B_{m-1}} B_m \dots B_n$

$\dots$

$\boxed{C_1}\ B_2 \dots B_n \to \boxed{B_1} \dots B_n$

Kde C_i jsou nově přidané neterminály, které se jinde nevyskytují.

Myšlenka důkazu

Každé pravidlo typu

$$A_1 A_2 \dots A_m \to B_1 B_2 \dots B_n,$$

které nemá kontextový tvar, nahradíme množinou pravidel:

$\boxed{A_1}\ A_2\ A_3 \dots A_m \to \boxed{C_1}\ A_2\ A_3 \dots A_m$

$C_1\ \boxed{A_2}\ A_3 \dots A_m \to C_1\ \boxed{C_2}\ A_3 \dots A_m$

$C_1\ C_2\ \boxed{A_3}\ \dots A_m \to C_1\ C_2\ \boxed{C_3}\ \dots A_m$

$\dots$

$C_1\ C_2 \dots C_{m-1}\ \boxed{A_m} \to C_1\ C_2 \dots C_{m-1}\ \boxed{C_m\ B_{m+1} \dots B_n}$

$C_1\ C_2 \dots C_{m-1} \boxed{C_m} B_{m+1} \dots B_n \to C_1 \dots C_{m-1} \boxed{B_m} \dots B_n$

$C_1\ C_2 \dots \boxed{C_{m-1}} B_m\ B_{m+1} \dots B_n \to C_1 \dots\ \boxed{B_{m-1}} B_m \dots B_n$

$\dots$

$\boxed{C_1}\ B_2 \dots B_n \to \boxed{B_1} \dots B_n$

Kde C_i jsou nově přidané neterminály, které se jinde nevyskytují.

$L = \{ww \mid w \in \{a,b\}^*\}$

$S \to XZ_aZ_a \mid XZ_bZ_b \mid aa \mid bb \mid \varepsilon$

$XZ_a \to aZ_aX_a \mid bZ_aX_b \mid aaX_a \mid baX_b$

$X_aa \to aX_a$
$X_ab \to bX_a$
$X_ba \to aX_b$
$X_bb \to bX_b$

$X_aZ_a \to YaZ_a \mid aa$
$X_bZ_a \to YbZ_a \mid ba$

$aY \to Ya$
$bY \to Yb$

$Z_aY \to XZ_a$

Dále totéž pro Z_b místo Z_a:

$XZ_b \to aZ_bX_a \mid bZ_bX_b \mid abX_a \mid bbX_b$
$Z_bY \to XZ_b$

$X_aZ_b \to YaZ_b \mid ab$
$X_bZ_b \to YbZ_b \mid bb$

Ukázka odvození:

$$S \Rightarrow XZ_aZ_a \Rightarrow aZ_aX_aZ_a \Rightarrow aZ_aYaZ_a \Rightarrow aXZ_aaZ_a \Rightarrow$$
$$\Rightarrow abZ_aX_baZ_a \Rightarrow abZ_aaX_bZ_a \Rightarrow abZ_aaYbZ_a \Rightarrow abZ_aYabZ_a \Rightarrow$$
$$\Rightarrow abXZ_aabZ_a \Rightarrow abbX_babZ_a \Rightarrow abbaX_bbZ_a \Rightarrow abbabX_bZ_a \Rightarrow$$
$$\Rightarrow abbabba$$

$L = \{ww \mid w \in \{a,b\}^*\}$

$S \to XZ_aZ_a \mid XZ_bZ_b \mid aa \mid bb \mid \varepsilon$

$XZ_a \to aZ_aX_a \mid bZ_aX_b \mid aaX_a \mid baX_b$

$X_aa \to aX_a$
$X_ab \to bX_a$
$X_ba \to aX_b$
$X_bb \to bX_b$

$X_aZ_a \to YaZ_a \mid aa$
$X_bZ_a \to YbZ_a \mid ba$

$aY \to Ya$
$bY \to Yb$

$Z_aY \to XZ_a$

Dále totéž pro Z_b místo Z_a:

$XZ_b \to aZ_bX_a \mid bZ_bX_b \mid abX_a \mid bbX_b$
$Z_bY \to XZ_b$

$X_aZ_b \to YaZ_b \mid ab$
$X_bZ_b \to YbZ_b \mid bb$

Ukázka odvození:

$S \Rightarrow XZ_aZ_a \Rightarrow aZ_aX_aZ_a \Rightarrow aZ_aYaZ_a \Rightarrow aXZ_aaZ_a \Rightarrow$
$\Rightarrow abZ_aX_baZ_a \Rightarrow abZ_aaX_bZ_a \Rightarrow abZ_aaYbZ_a \Rightarrow abZ_aYabZ_a \Rightarrow$
$\Rightarrow abXZ_aabZ_a \Rightarrow abbX_babZ_a \Rightarrow abbaX_bbZ_a \Rightarrow abbabX_bZ_a \Rightarrow$
$\Rightarrow abbabba$

$L = \{ww \mid w \in \{a,b\}^*\}$

$S \to XZ_aZ_a \mid XZ_bZ_b \mid aa \mid bb \mid \varepsilon$

$XZ_a \to aZ_aX_a \mid bZ_aX_b \mid aaX_a \mid baX_b$

$X_aa \to aX_a$
$X_ab \to bX_a$
$X_ba \to aX_b$
$X_bb \to bX_b$

$X_aZ_a \to YaZ_a \mid aa$
$X_bZ_a \to YbZ_a \mid ba$

$aY \to Ya$
$bY \to Yb$

$Z_aY \to XZ_a$

Dále totéž pro Z_b místo Z_a:

$XZ_b \to aZ_bX_a \mid bZ_bX_b \mid abX_a \mid bbX_b$
$Z_bY \to XZ_b$

$X_aZ_b \to YaZ_b \mid ab$
$X_bZ_b \to YbZ_b \mid bb$

Ukázka odvození:

$S \Rightarrow XZ_aZ_a \Rightarrow aZ_aX_aZ_a \Rightarrow aZ_aYaZ_a \Rightarrow aXZ_aaZ_a \Rightarrow$
$\Rightarrow abZ_aX_baZ_a \Rightarrow abZ_aaX_bZ_a \Rightarrow abZ_aaYbZ_a \Rightarrow abZ_aYabZ_a \Rightarrow$
$\Rightarrow abXZ_aabZ_a \Rightarrow abbX_babZ_a \Rightarrow abbaX_bbZ_a \Rightarrow abbabX_bZ_a \Rightarrow$
$\Rightarrow abbabba$

$L = \{ww \mid w \in \{a,b\}^*\}$

$S \to XZ_aZ_a \mid XZ_bZ_b \mid aa \mid bb \mid \varepsilon$

$XZ_a \to aZ_aX_a \mid bZ_aX_b \mid aaX_a \mid baX_b$

$X_aa \to aX_a$
$X_ab \to bX_a$
$X_ba \to aX_b$
$X_bb \to bX_b$

$X_aZ_a \to YaZ_a \mid aa$
$X_bZ_a \to YbZ_a \mid ba$

$aY \to Ya$
$bY \to Yb$

$Z_aY \to XZ_a$

Dále totéž pro Z_b místo Z_a:

$XZ_b \to aZ_bX_a \mid bZ_bX_b \mid abX_a \mid bbX_b$
$Z_bY \to XZ_b$

$X_aZ_b \to YaZ_b \mid ab$
$X_bZ_b \to YbZ_b \mid bb$

Ukázka odvození:

$S \Rightarrow XZ_aZ_a \Rightarrow aZ_aX_aZ_a \Rightarrow aZ_aYaZ_a \Rightarrow aXZ_aaZ_a \Rightarrow$
$\Rightarrow abZ_aX_baZ_a \Rightarrow abZ_aaX_bZ_a \Rightarrow abZ_aaYbZ_a \Rightarrow abZ_aYabZ_a \Rightarrow$
$\Rightarrow abXZ_aabZ_a \Rightarrow abbX_babZ_a \Rightarrow abbaX_bbZ_a \Rightarrow abbabX_bZ_a \Rightarrow$
$\Rightarrow abbabba$

$$L = \left\{ ww \;\middle|\; w \in \{a,b\}^* \right\}$$

$$S \to XZ_aZ_a \mid XZ_bZ_b \mid aa \mid bb \mid \varepsilon$$

$$XZ_a \to aZ_aX_a \mid bZ_aX_b \mid aaX_a \mid baX_b$$

$$X_aa \to aX_a$$
$$X_ab \to bX_a$$
$$X_ba \to aX_b$$
$$X_bb \to bX_b$$

$$X_aZ_a \to YaZ_a \mid aa$$
$$X_bZ_a \to YbZ_a \mid ba$$

$$aY \to Ya$$
$$bY \to Yb$$

$$Z_aY \to XZ_a$$

Dále totéž pro Z_b místo Z_a:

$$XZ_b \to aZ_bX_a \mid bZ_bX_b \mid abX_a \mid bbX_b$$
$$Z_bY \to XZ_b$$

$$X_aZ_b \to YaZ_b \mid ab$$
$$X_bZ_b \to YbZ_b \mid bb$$

Ukázka odvození:

$$S \Rightarrow XZ_aZ_a \Rightarrow aZ_aX_aZ_a \Rightarrow aZ_aYaZ_a \Rightarrow aXZ_aaZ_a \Rightarrow$$
$$\Rightarrow abZ_aX_baZ_a \Rightarrow abZ_aaX_bZ_a \Rightarrow abZ_aaYbZ_a \Rightarrow abZ_aYabZ_a \Rightarrow$$
$$\Rightarrow abXZ_aabZ_a \Rightarrow abbX_babZ_a \Rightarrow abbaX_bbZ_a \Rightarrow abbabX_bZ_a \Rightarrow$$
$$\Rightarrow abbabba$$

Kurodova normální forma

Definice (Kurodova normální forma pro gramatiky typu 1)

Gramatika je v KNF, jestliže všechna její pravidla jsou ve tvaru

$$\begin{array}{rcl} A & \to & BC \\ AB & \to & CD \\ A & \to & a \end{array}$$

$A, B, C, D \in N$, $a \in T$, *případně* $S \to \varepsilon$, S *není na pravé straně žádného pravidla.*

Převod do KNF

Věta

Ke každé nezkracující gramatice G lze sestrojit gramatiku G' v Kurodově normální formě takovou, že $L(G') = L(G)$.

Myšlenka důkazu

Použijeme postup, který známe pro gramatiky typu 0.
Vstup: nezkracující gramatika bez jednoduchých pravidel typu $X \to Y, \quad X, Y \in N$

- všechny terminály (na levé i pravé straně pravidla) $a \in T$ nahradíme „pomocnými" neterminály N_a,
- pro všechny neterminály vytvořené v předchozím bodu přidáme pravidlo $N_a \to a$,
- pravidla neodpovídající předpisu upravíme – rozdělíme.

Převod do KNF

Věta

Ke každé nezkracující gramatice G lze sestrojit gramatiku G' v Kurodově normální formě takovou, že $L(G') = L(G)$.

Myšlenka důkazu

Použijeme postup, který známe pro gramatiky typu 0.
Vstup: nezkracující gramatika bez jednoduchých pravidel typu $X \to Y, \quad X, Y \in N$

- všechny terminály (na levé i pravé straně pravidla) $a \in T$ nahradíme „pomocnými“ neterminály N_a,
- pro všechny neterminály vytvořené v předchozím bodu přidáme pravidlo $N_a \to a$,
- pravidla neodpovídající předpisu upravíme – rozdělíme.

Převod do KNF

Věta

Ke každé nezkracující gramatice G lze sestrojit gramatiku G' v Kurodově normální formě takovou, že $L(G') = L(G)$.

Myšlenka důkazu

Použijeme postup, který známe pro gramatiky typu 0.
Vstup: nezkracující gramatika bez jednoduchých pravidel typu $X \to Y, \quad X, Y \in N$

- všechny terminály (na levé i pravé straně pravidla) $a \in T$ nahradíme „pomocnými" neterminály N_a,
- pro všechny neterminály vytvořené v předchozím bodu přidáme pravidlo $N_a \to a$,
- pravidla neodpovídající předpisu upravíme – rozdělíme.

Převod do KNF

Věta

Ke každé nezkracující gramatice G lze sestrojit gramatiku G' v Kurodově normální formě takovou, že $L(G') = L(G)$.

Myšlenka důkazu

Použijeme postup, který známe pro gramatiky typu 0.
Vstup: nezkracující gramatika bez jednoduchých pravidel typu $X \to Y, \quad X, Y \in N$

- všechny terminály (na levé i pravé straně pravidla) $a \in T$ nahradíme „pomocnými" neterminály N_a,
- pro všechny neterminály vytvořené v předchozím bodu přidáme pravidlo $N_a \to a$,
- pravidla neodpovídající předpisu upravíme – rozdělíme.

Převod do KNF

Věta

Ke každé nezkracující gramatice G lze sestrojit gramatiku G' v Kurodově normální formě takovou, že $L(G') = L(G)$.

Myšlenka důkazu

Použijeme postup, který známe pro gramatiky typu 0.
Vstup: nezkracující gramatika bez jednoduchých pravidel typu $X \to Y, \quad X, Y \in N$

- všechny terminály (na levé i pravé straně pravidla) $a \in T$ nahradíme „pomocnými“ neterminály N_a,
- pro všechny neterminály vytvořené v předchozím bodu přidáme pravidlo $N_a \to a$,
- pravidla neodpovídající předpisu upravíme – rozdělíme.

Převod do KNF

$AB \rightarrow CD$		OK
$A \rightarrow CD$		OK

$\boxed{A \rightarrow BCDE}$	$A \rightarrow BX_1$ $X_1 \rightarrow CX_2$ $X_2 \rightarrow DE$		OK
$\boxed{ABCD \rightarrow EFGHIJK}$	$AB \rightarrow EX_3$ $X_3C \rightarrow FX_4$ $X_4D \rightarrow GX_5$	$X_5 \rightarrow HX_6$ $X_6 \rightarrow IX_7$ $X_7 \rightarrow JK$	OK

Převod do KNF

$AB \to CD$ OK

$A \to CD$ OK

$\boxed{A \to BCDE}$

$$A \to BX_1$$
$$X_1 \to CX_2$$
$$X_2 \to DE$$

OK

$\boxed{ABCD \to EFGHIJK}$

$$AB \to EX_3 \qquad X_5 \to HX_6$$
$$X_3C \to FX_4 \qquad X_6 \to IX_7$$
$$X_4D \to GX_5 \qquad X_7 \to JK$$

OK

Převod do KNF

$AB \to CD$ OK
$A \to CD$ OK

$\boxed{A \to BCDE}$	$A \to BX_1$ $X_1 \to CX_2$ $X_2 \to DE$	OK

$\boxed{ABCD \to EFGHIJK}$	$AB \to EX_3$ $X_3C \to FX_4$ $X_4D \to GX_5$	$X_5 \to HX_6$ $X_6 \to IX_7$ $X_7 \to JK$	OK

Definice (Lineárně ohraničený automat)

je jednopáskový (nedeterministický) Turingův stroj

$$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \{-1, 0, 1\}),$$

ve kterém čtecí a zápisová hlava nesmí během výpočtu přepsat hraniční symboly $\$$ *pásky (tj. slovo nesmí být během výpočtu prodlužováno).*
Rozšíření: je dovoleno prodloužit pásku maximálně lineárně.

Definice (Konfigurace lineárně ohraničeného automatu)

Konfigurace lineárně ohraničeného automatu je

$$(\alpha, q, \beta),$$

kde q *je stav, na pásce je řetězec* $\alpha\beta$*, čtecí a zápisová hlava ukazuje na první symbol řetězce* β*.*

Další definice taktéž přejímáme.

Definice (Lineárně ohraničený automat)

je jednopáskový (nedeterministický) Turingův stroj

$$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \{-1, 0, 1\}),$$

ve kterém čtecí a zápisová hlava nesmí během výpočtu přepsat hraniční symboly $\$$ *pásky (tj. slovo nesmí být během výpočtu prodlužováno).*
Rozšíření: je dovoleno prodloužit pásku maximálně lineárně.

Definice (Konfigurace lineárně ohraničeného automatu)

Konfigurace lineárně ohraničeného automatu je

$$(\alpha, q, \beta),$$

kde q *je stav, na pásce je řetězec* $\alpha\beta$*, čtecí a zápisová hlava ukazuje na první symbol řetězce* β*.*

Další definice taktéž přejímáme.

Definice (Lineárně ohraničený automat)

je jednopáskový (nedeterministický) Turingův stroj

$$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \{-1, 0, 1\}),$$

ve kterém čtecí a zápisová hlava nesmí během výpočtu přepsat hraniční symboly \$ pásky (tj. slovo nesmí být během výpočtu prodlužováno).
Rozšíření: je dovoleno prodloužit pásku maximálně lineárně.

Definice (Konfigurace lineárně ohraničeného automatu)

Konfigurace lineárně ohraničeného automatu je

$$(\alpha, q, \beta),$$

kde q je stav, na pásce je řetězec $\alpha\beta$, čtecí a zápisová hlava ukazuje na první symbol řetězce β.

Další definice taktéž přejímáme.

Věta

Pro konkrétní lineárně ohraničený automat $\mathcal{M}$ a daný vstup w_0 existuje konečný počet různých konfigurací.

Důkaz:

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \{-1, 0, 1\})$

d … délka používané části pásky,

s … počet stavů v Q, tedy $|Q| = s$,

g … počet prvků páskové abecedy, $\Gamma = \{a_1, a_2, \dots, a_g\}$

$d = 1$ $\$\boxed{}\$$	$d = 2$ $\$\boxed{}\boxed{}\$$	$d = ?$
a_1	a_1a_1	všechny
a_2	a_1a_2	možné
…	…	d-tice
a_g	a_2a_2	symbolů
	a_2a_3	
	…	
	a_ga_g	
Počet možností:		
g	$g \cdot g = g^2$	g^d

Důkaz:

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \{-1, 0, 1\})$

d ... délka používané části pásky,

s ... počet stavů v Q, tedy $|Q| = s$,

g ... počet prvků páskové abecedy, $\Gamma = \{a_1, a_2, \ldots, a_g\}$

$d = 1$ $\$\square\$$	$d = 2$ $\$\square\square\$$	$d = ?$
a_1	a_1a_1	všechny
a_2	a_1a_2	možné
...	...	d-tice
a_g	a_2a_2	symbolů
	a_2a_3	
	...	
	a_ga_g	
Počet možností:		
g	$g \cdot g = g^2$	g^d

Důkaz:

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \{-1, 0, 1\})$

d …délka používané části pásky,
s …počet stavů v Q, tedy $|Q| = s$,
g …počet prvků páskové abecedy, $\Gamma = \{a_1, a_2, \dots, a_g\}$

Celý vzorec:

- g^d stavů pásky,
- d pozic čtecí a zápisové hlavy,
- s různých stavů na každé pozici hlavy:

$$\text{Počet konfigurací} = g^d \cdot d \cdot s$$

Důkaz:

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \{-1, 0, 1\})$

d . . . délka používané části pásky,
s . . . počet stavů v Q, tedy $|Q| = s$,
g . . . počet prvků páskové abecedy, $\Gamma = \{a_1, a_2, \ldots, a_g\}$

Celý vzorec:

- g^d stavů pásky,
- d pozic čtecí a zápisové hlavy,
- s různých stavů na každé pozici hlavy:

$$\text{Počet konfigurací} = g^d \cdot d \cdot s$$

Důkaz:

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F, \{-1, 0, 1\})$

d ... délka používané části pásky,
s ... počet stavů v Q, tedy $|Q| = s$,
g ... počet prvků páskové abecedy, $\Gamma = \{a_1, a_2, \ldots, a_g\}$

Celý vzorec:

- g^d stavů pásky,
- d pozic čtecí a zápisové hlavy,
- s různých stavů na každé pozici hlavy:

$$\text{Počet konfigurací} \quad = \quad g^d \cdot d \cdot s$$

Důsledek

Věta

LOA lze vždy navrhnout tak, aby výpočet skončil nad jakýmkoliv vstupem. Proto jazyky, které jsou přijímány nějakým LOA, jsou právě rekurzívní jazyky.

Vztah k rekurzivním jazykům

- Rekurzivní jazyky jsou jazyky *rozhodované* Turingovým strojem (TS se zastaví na jakýkoliv vstup, bez nekonečné smyčky).
- Lineárně ohraničený automat je Turingův stroj, který má ohraničenou pásku a proto ho lze sestrojit tak, aby se zastavil na každý vstup.
- $\Rightarrow$ Jazyky rozhodované lineárně ohraničenými automaty jsou právě rekurzivní jazyky.

Důsledek

Věta

LOA lze vždy navrhnout tak, aby výpočet skončil nad jakýmkoliv vstupem. Proto jazyky, které jsou přijímány nějakým LOA, jsou právě rekurzívní jazyky.

Vztah k rekurzivním jazykům

- Rekurzivní jazyky jsou jazyky *rozhodované* Turingovým strojem (TS se zastaví na jakýkoliv vstup, bez nekonečné smyčky).
- Lineárně ohraničený automat je Turingův stroj, který má ohraničenou pásku a proto ho lze sestrojit tak, aby se zastavil na každý vstup.
- $\Rightarrow$ Jazyky rozhodované lineárně ohraničenými automaty jsou právě rekurzivní jazyky.

Důsledek

Věta

LOA lze vždy navrhnout tak, aby výpočet skončil nad jakýmkoliv vstupem. Proto jazyky, které jsou přijímány nějakým LOA, jsou právě rekurzívní jazyky.

Vztah k rekurzivním jazykům

- Rekurzivní jazyky jsou jazyky *rozhodované* Turingovým strojem (TS se zastaví na jakýkoliv vstup, bez nekonečné smyčky).
- Lineárně ohraničený automat je Turingův stroj, který má ohraničenou pásku a proto ho lze sestrojit tak, aby se zastavil na každý vstup.
- $\Rightarrow$ Jazyky rozhodované lineárně ohraničenými automaty jsou právě rekurzivní jazyky.

Důsledek

Věta

LOA lze vždy navrhnout tak, aby výpočet skončil nad jakýmkoliv vstupem. Proto jazyky, které jsou přijímány nějakým LOA, jsou právě rekurzívní jazyky.

Vztah k rekurzivním jazykům

- Rekurzivní jazyky jsou jazyky *rozhodované* Turingovým strojem (TS se zastaví na jakýkoliv vstup, bez nekonečné smyčky).
- Lineárně ohraničený automat je Turingův stroj, který má ohraničenou pásku a proto ho lze sestrojit tak, aby se zastavil na každý vstup.
- $\Rightarrow$ Jazyky rozhodované lineárně ohraničenými automaty jsou právě rekurzivní jazyky.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Postup:

Sestrojíme LOA $\mathcal{M}'$, který se zastaví na každý vstup, k (obecnému) LOA $\mathcal{M}$.

- na vstupu bude slovo $\$w_0 \cdot \# \sqcup^k \$$, kde
  - w_0 je vstupní slovo stroje $\mathcal{M}$, $d = |w_0|$,
  - $k = (d+1) \cdot (g^d \cdot d \cdot s)$ (délka krát počet konfigurací), číslo 1 je buňka pro symbol oddělující následující konfiguraci – $\#$,
- v první fázi nahradíme na pásce řetězec $\sqcup^k$ posloupností všech možných konfigurací oddělených symbolem $\#$,
- dále simulujeme činnost původního stroje $\mathcal{M}$,
- po každém kroku původního stroje
  - označíme pozici hlavy,
  - na přidané části pásky najdeme konfiguraci, ve které se stroj právě nachází,
  - pokud není označená, označíme ji, vrátíme se k první části pásky a pokračujeme dalším krokem,
  - pokud je označená, znamená to, že v této konfiguraci už stroj jednou byl, ukončíme výsledek v q_{reject}.

Vztah mezi jazyky typu 1 a LOA

Věta

Jazyky typu 1 jsou právě jazyky přijímané lineárně ohraničenými automaty.

Důkaz „⇒“:

Nezkracující gramatika G generuje zadaný jazyk typu 1 L.
Derivace v této gramatice je ve tvaru
$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \ldots w_n$, kde $|w_i| \leq |w_j|$ pro $i < j$.

Důkaz „⇒“ – návrh LOA $\mathcal{M}$:

1. Na vstupu je slovo w_0, o kterém chceme zjistit, zda je generováno G.
2. Na w_0 budeme uplatňovat pravidla gramatiky:
   1. nedeterministicky zvolíme pravidlo gramatiky $(\alpha \to \beta)$,
   2. najdeme v řetězci na pásce některý výskyt β – pokud je jich více, nedeterministicky mezi nimi jeden zvolíme,
   3. výskyt β přepíšeme řetězcem α, pokud je α kratší, posuneme to, co následuje za β, doleva, aby zbytek pracovního slova navazoval na α.
3. Výpočet ukončíme:
   - ve stavu akceptování (q_{accept}), pokud na pásce bude pouze startovací symbol gramatiky a nic jiného,
   - ve stavu odmítnutí slova (q_{reject}, q_{error}), pokud je na pásce něco jiného a již nelze použít žádné pravidlo gramatiky.

Důkaz „$\Rightarrow$“ – návrh LOA $\mathcal{M}$:

1. Na vstupu je slovo w_0, o kterém chceme zjistit, zda je generováno G.
2. Na w_0 budeme uplatňovat pravidla gramatiky:
   1. nedeterministicky zvolíme pravidlo gramatiky $(\alpha \to \beta)$,
   2. najdeme v řetězci na pásce některý výskyt β – pokud je jich více, nedeterministicky mezi nimi jeden zvolíme,
   3. výskyt β přepíšeme řetězcem α, pokud je α kratší, posuneme to, co následuje za β, doleva, aby zbytek pracovního slova navazoval na α.
3. Výpočet ukončíme:
   - ve stavu akceptování (q_{accept}), pokud na pásce bude pouze startovací symbol gramatiky a nic jiného,
   - ve stavu odmítnutí slova $(q_{reject},\ q_{error})$, pokud je na pásce něco jiného a již nelze použít žádné pravidlo gramatiky.

Důkaz „⇒“ – návrh LOA $\mathcal{M}$:

1. Na vstupu je slovo w_0, o kterém chceme zjistit, zda je generováno G.
2. Na w_0 budeme uplatňovat pravidla gramatiky:
   1. nedeterministicky zvolíme pravidlo gramatiky $(\alpha \to \beta)$,
   2. najdeme v řetězci na pásce některý výskyt β – pokud je jich více, nedeterministicky mezi nimi jeden zvolíme,
   3. výskyt β přepíšeme řetězcem α, pokud je α kratší, posuneme to, co následuje za β, doleva, aby zbytek pracovního slova navazoval na α.
3. Výpočet ukončíme:
   - ve stavu akceptování (q_{accept}), pokud na pásce bude pouze startovací symbol gramatiky a nic jiného,
   - ve stavu odmítnutí slova (q_{reject}, q_{error}), pokud je na pásce něco jiného a již nelze použít žádné pravidlo gramatiky.

Důkaz „⇒“ – návrh LOA $\mathcal{M}$:

1. Na vstupu je slovo w_0, o kterém chceme zjistit, zda je generováno G.
2. Na w_0 budeme uplatňovat pravidla gramatiky:
   1. nedeterministicky zvolíme pravidlo gramatiky $(\alpha \rightarrow \beta)$,
   2. najdeme v řetězci na pásce některý výskyt β – pokud je jich více, nedeterministicky mezi nimi jeden zvolíme,
   3. výskyt β přepíšeme řetězcem α, pokud je α kratší, posuneme to, co následuje za β, doleva, aby zbytek pracovního slova navazoval na α.
3. Výpočet ukončíme:
   - ve stavu akceptování (q_{accept}), pokud na pásce bude pouze startovací symbol gramatiky a nic jiného,
   - ve stavu odmítnutí slova (q_{reject}, q_{error}), pokud je na pásce něco jiného a již nelze použít žádné pravidlo gramatiky.

Důkaz „$\Rightarrow$“ – návrh LOA $\mathcal{M}$:

1. Na vstupu je slovo w_0, o kterém chceme zjistit, zda je generováno G.
2. Na w_0 budeme uplatňovat pravidla gramatiky:
    1. nedeterministicky zvolíme pravidlo gramatiky $(\alpha \to \beta)$,
    2. najdeme v řetězci na pásce některý výskyt β – pokud je jich více, nedeterministicky mezi nimi jeden zvolíme,
    3. výskyt β přepíšeme řetězcem α, pokud je α kratší, posuneme to, co následuje za β, doleva, aby zbytek pracovního slova navazoval na α.
3. Výpočet ukončíme:
    - ve stavu akceptování (q_{accept}), pokud na pásce bude pouze startovací symbol gramatiky a nic jiného,
    - ve stavu odmítnutí slova (q_{reject}, q_{error}), pokud je na pásce něco jiného a již nelze použít žádné pravidlo gramatiky.

Důkaz „$\Rightarrow$“ – návrh LOA $\mathcal{M}$:

1. Na vstupu je slovo w_0, o kterém chceme zjistit, zda je generováno G.
2. Na w_0 budeme uplatňovat pravidla gramatiky:
   1. nedeterministicky zvolíme pravidlo gramatiky $(\alpha \to \beta)$,
   2. najdeme v řetězci na pásce některý výskyt β – pokud je jich více, nedeterministicky mezi nimi jeden zvolíme,
   3. výskyt β přepíšeme řetězcem α, pokud je α kratší, posuneme to, co následuje za β, doleva, aby zbytek pracovního slova navazoval na α.
3. Výpočet ukončíme:
   - ve stavu akceptování (q_{accept}), pokud na pásce bude pouze startovací symbol gramatiky a nic jiného,
   - ve stavu odmítnutí slova $(q_{reject},\ q_{error})$, pokud je na pásce něco jiného a již nelze použít žádné pravidlo gramatiky.

Důkaz „⇒“ – návrh LOA $\mathcal{M}$:

1. Na vstupu je slovo w_0, o kterém chceme zjistit, zda je generováno G.
2. Na w_0 budeme uplatňovat pravidla gramatiky:
   1. nedeterministicky zvolíme pravidlo gramatiky $(\alpha \to \beta)$,
   2. najdeme v řetězci na pásce některý výskyt β – pokud je jich více, nedeterministicky mezi nimi jeden zvolíme,
   3. výskyt β přepíšeme řetězcem α, pokud je α kratší, posuneme to, co následuje za β, doleva, aby zbytek pracovního slova navazoval na α.
3. Výpočet ukončíme:
   - ve stavu akceptování (q_{accept}), pokud na pásce bude pouze startovací symbol gramatiky a nic jiného,
   - ve stavu odmítnutí slova $(q_{reject},\ q_{error})$, pokud je na pásce něco jiného a již nelze použít žádné pravidlo gramatiky.

Důkaz „⇒“ – návrh LOA $\mathcal{M}$:

1. Na vstupu je slovo w_0, o kterém chceme zjistit, zda je generováno G.
2. Na w_0 budeme uplatňovat pravidla gramatiky:
   1. nedeterministicky zvolíme pravidlo gramatiky $(\alpha \to \beta)$,
   2. najdeme v řetězci na pásce některý výskyt β – pokud je jich více, nedeterministicky mezi nimi jeden zvolíme,
   3. výskyt β přepíšeme řetězcem α, pokud je α kratší, posuneme to, co následuje za β, doleva, aby zbytek pracovního slova navazoval na α.
3. Výpočet ukončíme:
   - ve stavu akceptování (q_{accept}), pokud na pásce bude pouze startovací symbol gramatiky a nic jiného,
   - ve stavu odmítnutí slova (q_{reject}, q_{error}), pokud je na pásce něco jiného a již nelze použít žádné pravidlo gramatiky.

Důkaz „$\Leftarrow$“:

Je dán LOA $\mathcal{M}$. Vytvoříme nezkracující gramatiku G.

1. V gramatice G nejdřív vygenerujeme řetězec neterminálů, který představuje dvojici slov na vstupu simulovaného LOA. Oproti gramatice typu 0 musíme symboly pro začátek a konec pracovní části pásky a také symbol pro stav automatu umístit dovnitř neterminálů pro symboly slova, abychom nebyli nuceni na konci výpočtu použít epsilonová pravidla.

$$S \to \begin{bmatrix} a \\ \$, q_0, a \end{bmatrix} S' \;\Big|\; \begin{bmatrix} \varepsilon \\ \$, q_0, \$ \end{bmatrix}$$

$$S' \to \begin{bmatrix} a \\ a \end{bmatrix} S' \;\Big|\; \begin{bmatrix} a \\ a, \$ \end{bmatrix} \text{ pro každé } a \in \Sigma$$

Dostaneme řetězec neterminálů

$$\begin{bmatrix} a_1 \\ \$, q_0, a_1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_2 \end{bmatrix} \begin{bmatrix} a_3 \\ a_3 \end{bmatrix} \dots \begin{bmatrix} a_k \\ a_k, \$ \end{bmatrix}$$

Důkaz „⇐“:

Je dán LOA $\mathcal{M}$. Vytvoříme nezkracující gramatiku G.

2. Simulujeme průběh výpočtu LOA:

   Pro každou část δ funkce typu $\delta(q_i, a) \ni (q_j, b, 1)$

   $$\begin{bmatrix} x \\ q_i, a \end{bmatrix} \begin{bmatrix} y \\ c \end{bmatrix} \rightarrow \begin{bmatrix} x \\ b \end{bmatrix} \begin{bmatrix} y \\ q_j, c \end{bmatrix} \text{ pro každé } c \in \Gamma - \{\$\}$$

   Pro každou část δ funkce typu $\delta(q_i, a) \ni (q_j, b, 0)$

   $$\begin{bmatrix} x \\ q_i, a \end{bmatrix} \rightarrow \begin{bmatrix} x \\ q_j, b \end{bmatrix}$$

   Pro každou část δ funkce typu $\delta(q_i, a) \ni (q_j, b, -1)$

   $$\begin{bmatrix} y \\ c \end{bmatrix} \begin{bmatrix} x \\ q_i, a \end{bmatrix} \rightarrow \begin{bmatrix} y \\ q_j, c \end{bmatrix} \begin{bmatrix} x \\ b \end{bmatrix} \text{ pro každé } c \in \Gamma - \{\$\}$$

Důkaz „$\Leftarrow$“:

Je dán LOA $\mathcal{M}$. Vytvoříme nezkracující gramatiku G.

2. Simulujeme průběh výpočtu LOA:

   Je nutné ošetřit \$ (okraje pásek):

   Pro každou dvojici předpisů
   $\delta(q_k, a) \ni (q_i, b, -1), \quad \delta(q_i, \$) \ni (q_j, \$, 1)$

   $$\left[\begin{array}{c} x \\ \$, q_k, a \end{array}\right] \rightarrow \left[\begin{array}{c} x \\ \$, q_j, b \end{array}\right]$$

Důkaz „⇐“:

Je dán LOA $\mathcal{M}$. Vytvoříme nezkracující gramatiku G.

3. Ukončení výpočtu:

$$\begin{bmatrix} x \\ a \end{bmatrix} \begin{bmatrix} y \\ q_{acc}, b \end{bmatrix} \rightarrow \begin{bmatrix} x \\ q_{acc}, a \end{bmatrix} \begin{bmatrix} y \\ b \end{bmatrix}$$

$$\begin{bmatrix} x \\ \$, a \end{bmatrix} \begin{bmatrix} y \\ q_{acc}, b \end{bmatrix} \rightarrow x \begin{bmatrix} y \\ K, b \end{bmatrix}$$

$$\begin{bmatrix} x \\ K, a \end{bmatrix} \begin{bmatrix} y \\ b \end{bmatrix} \rightarrow x \begin{bmatrix} y \\ K, b \end{bmatrix}$$

$$\begin{bmatrix} x \\ K, a \end{bmatrix} \begin{bmatrix} y \\ b, \$ \end{bmatrix} \rightarrow xy$$

Uzávěrové vlastnosti – $\cup, \circ, *, +, h, s$

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operacím

- *sjednocení,*
- *zřetězení,*
- *iterace,*
- *pozitivní iterace,*
- *homomorfismu,*
- *substituce.*

Uzávěrové vlastnosti – $\cup, \circ, *, +, h, s$

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operacím

- *sjednocení,*
- *zřetězení,*
- *iterace,*
- *pozitivní iterace,*
- *homomorfismu,*
- *substituce.*

Uzávěrové vlastnosti – $\cup, \circ, *, +, h, s$

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operacím

- *sjednocení,*
- *zřetězení,*
- *iterace,*
- *pozitivní iterace,*
- *homomorfismu,*
- *substituce.*

Uzávěrové vlastnosti – $\cup, \circ, *, +, h, s$

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operacím

- *sjednocení,*
- *zřetězení,*
- *iterace,*
- *pozitivní iterace,*
- *homomorfismu,*
- *substituce.*

Uzávěrové vlastnosti – $\cup, \circ, *, +, h, s$

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operacím

- *sjednocení,*
- *zřetězení,*
- *iterace,*
- *pozitivní iterace,*
- *homomorfismu,*
- *substituce.*

Uzávěrové vlastnosti – $\cup, \circ, *, +, h, s$

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operacím

- *sjednocení,*
- *zřetězení,*
- *iterace,*
- *pozitivní iterace,*
- *homomorfismu,*
- *substituce.*

Uzávěrové vlastnosti – průnik

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operaci průniku.

Postup

Když vytváříme LOA $\mathcal{M}$ rozpoznávající průnik jazyků rozpoznávaných $\mathcal{M}_1$ a $\mathcal{M}_2$, budeme postupovat takto:

- předpokládejme, že stroje $\mathcal{M}_1$ a $\mathcal{M}_2$ mají zpracovat slovo w – vstupem stroje $\mathcal{M}$ bude $\$w\#w\$$,
- na část pásky $\$w\#$ spustíme simulaci stroje $\mathcal{M}_1$,
- když skončí ve stavu q_{reject}, rovnou ukončíme celý výpočet,
- když skončí ve stavu q_{accept}, na druhou část pásky $\#w\$$ spustíme simulaci stroje $\mathcal{M}_2$,
- podle výsledku této simulace ukončíme výpočet ve stavu q_{accept} nebo q_{reject}.

Uzávěrové vlastnosti – průnik

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operaci průniku.

Postup

Když vytváříme LOA $\mathcal{M}$ rozpoznávající průnik jazyků rozpoznávaných $\mathcal{M}_1$ a $\mathcal{M}_2$, budeme postupovat takto:

- předpokládejme, že stroje $\mathcal{M}_1$ a $\mathcal{M}_2$ mají zpracovat slovo w – vstupem stroje $\mathcal{M}$ bude $\$w\#w\$$,
- na část pásky $\$w\#$ spustíme simulaci stroje $\mathcal{M}_1$,
- když skončí ve stavu q_{reject}, rovnou ukončíme celý výpočet,
- když skončí ve stavu q_{accept}, na druhou část pásky $\#w\$$ spustíme simulaci stroje $\mathcal{M}_2$,
- podle výsledku této simulace ukončíme výpočet ve stavu q_{accept} nebo q_{reject}.

Uzávěrové vlastnosti – průnik

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operaci průniku.

Postup

Když vytváříme LOA $\mathcal{M}$ rozpoznávající průnik jazyků rozpoznávaných $\mathcal{M}_1$ a $\mathcal{M}_2$, budeme postupovat takto:

- předpokládejme, že stroje $\mathcal{M}_1$ a $\mathcal{M}_2$ mají zpracovat slovo w – vstupem stroje $\mathcal{M}$ bude $\$w\#w\$$,
- na část pásky $\$w\#$ spustíme simulaci stroje $\mathcal{M}_1$,
- když skončí ve stavu q_{reject}, rovnou ukončíme celý výpočet,
- když skončí ve stavu q_{accept}, na druhou část pásky $\#w\$$ spustíme simulaci stroje $\mathcal{M}_2$,
- podle výsledku této simulace ukončíme výpočet ve stavu q_{accept} nebo q_{reject}.

Uzávěrové vlastnosti – průnik

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operaci průniku.

Postup

Když vytváříme LOA $\mathcal{M}$ rozpoznávající průnik jazyků rozpoznávaných $\mathcal{M}_1$ a $\mathcal{M}_2$, budeme postupovat takto:

- předpokládejme, že stroje $\mathcal{M}_1$ a $\mathcal{M}_2$ mají zpracovat slovo w – vstupem stroje $\mathcal{M}$ bude $\$w\#w\$$,
- na část pásky $\$w\#$ spustíme simulaci stroje $\mathcal{M}_1$,
- když skončí ve stavu q_{reject}, rovnou ukončíme celý výpočet,
- když skončí ve stavu q_{accept}, na druhou část pásky $\#w\$$ spustíme simulaci stroje $\mathcal{M}_2$,
- podle výsledku této simulace ukončíme výpočet ve stavu q_{accept} nebo q_{reject}.

Uzávěrové vlastnosti – průnik

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operaci průniku.

Postup

Když vytváříme LOA $\mathcal{M}$ rozpoznávající průnik jazyků rozpoznávaných $\mathcal{M}_1$ a $\mathcal{M}_2$, budeme postupovat takto:

- předpokládejme, že stroje $\mathcal{M}_1$ a $\mathcal{M}_2$ mají zpracovat slovo w – vstupem stroje $\mathcal{M}$ bude $\$w\#w\$$,
- na část pásky $\$w\#$ spustíme simulaci stroje $\mathcal{M}_1$,
- když skončí ve stavu q_{reject}, rovnou ukončíme celý výpočet,
- když skončí ve stavu q_{accept}, na druhou část pásky $\#w\$$ spustíme simulaci stroje $\mathcal{M}_2$,
- podle výsledku této simulace ukončíme výpočet ve stavu q_{accept} nebo q_{reject}.

Uzávěrové vlastnosti – průnik

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operaci průniku.

Postup

Když vytváříme LOA $\mathcal{M}$ rozpoznávající průnik jazyků rozpoznávaných $\mathcal{M}_1$ a $\mathcal{M}_2$, budeme postupovat takto:

- předpokládejme, že stroje $\mathcal{M}_1$ a $\mathcal{M}_2$ mají zpracovat slovo w – vstupem stroje $\mathcal{M}$ bude $\$w\#w\$$,
- na část pásky $\$w\#$ spustíme simulaci stroje $\mathcal{M}_1$,
- když skončí ve stavu q_{reject}, rovnou ukončíme celý výpočet,
- když skončí ve stavu q_{accept}, na druhou část pásky $\#w\$$ spustíme simulaci stroje $\mathcal{M}_2$,
- podle výsledku této simulace ukončíme výpočet ve stavu q_{accept} nebo q_{reject}.

Uzávěrové vlastnosti – doplněk

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operaci doplňku.

Konstrukční důkaz:

- k původnímu LOA $\mathcal{M}_1$ sestrojíme ekvivalentní LOA $\mathcal{M}_1'$, který pro každý vstup ukončí výpočet po konečně mnoha krocích,
- výsledný LOA $\mathcal{M}$ sestrojíme z $\mathcal{M}_1'$ tak, že zaměníme akceptující a odmítající stavy.

Uzávěrové vlastnosti – doplněk

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operaci doplňku.

Konstrukční důkaz:

- k původnímu LOA $\mathcal{M}_1$ sestrojíme ekvivalentní LOA $\mathcal{M}_1'$, který pro každý vstup ukončí výpočet po konečně mnoha krocích,
- výsledný LOA $\mathcal{M}$ sestrojíme z $\mathcal{M}_1'$ tak, že zaměníme akceptující a odmítající stavy.

Uzávěrové vlastnosti – doplněk

Věta

Třída jazyků typu 1 je uzavřena vzhledem k operaci doplňku.

Konstrukční důkaz:

- k původnímu LOA $\mathcal{M}_1$ sestrojíme ekvivalentní LOA $\mathcal{M}'_1$, který pro každý vstup ukončí výpočet po konečně mnoha krocích,
- výsledný LOA $\mathcal{M}$ sestrojíme z $\mathcal{M}'_1$ tak, že zaměníme akceptující a odmítající stavy.