





## Nevypouštějící bezkontextové gramatiky

Bezkontextová gramatika  $G$  je **nevypouštějící** právě tehdy, když nemá pravidla ve tvaru  $X \rightarrow \lambda$ .

**Věta:** Ke každé bezkontextové gramatice  $G$  existuje nevypouštějící bezkontextová gramatika  $G_1$  taková, že  $L(G_1) = L(G) - \{\lambda\}$  (jazyky se liší maximálně o prázdné slovo).

Je-li  $\lambda \in L(G)$ , potom existuje BKG  $G_2$  tak, že  $L(G_2) = L(G)$  a jediné pravidlo s  $\lambda$  na pravé straně je  $S' \rightarrow \lambda$  a  $S'$  (počáteční neterminál) se nevyskytuje na pravé straně žádného pravidla  $G_2$  (tedy  $\mathcal{L}_1 \subseteq \mathcal{L}_2$ ).

**Příklad:**

$G: S \rightarrow 0S1 \mid \lambda$

$G_1: S \rightarrow 0S1 \mid 01$

$G_2: S' \rightarrow S \mid \lambda, S \rightarrow 0S1 \mid 01$



## Převod na nevypouštějící BKG

... aneb, jak se zbavit pravidel ve tvaru  $X \rightarrow \lambda$ ?

**Základní myšlenka:**

- pravidlo  $X \rightarrow \lambda$  se používá pro vyhození  $X$  ze slova
  - co když  $X$  do slova vůbec nezařadíme?
- ...,  $Y \rightarrow uXv, X \rightarrow \lambda, \dots \Rightarrow \dots, Y \rightarrow uXv, Y \rightarrow uv, \dots$

1) Nejprve zjistíme neterminály, které se přepisují na  $\lambda$ :

$U = \{X \mid X \in V_N \wedge X \Rightarrow^* \lambda\}$

Proč tak silně (nestačilo by  $X \rightarrow \lambda$  místo  $X \Rightarrow^* \lambda$ )?

Řešení derivací  $X \Rightarrow^{X \rightarrow Y} Y \Rightarrow^{Y \rightarrow Z} Z \Rightarrow^{Z \rightarrow \lambda} \lambda$

Iterační algoritmus pro získání  $U$ :

$U_1 = \{X \mid X \in V_N \wedge (X \rightarrow \lambda) \in P\}$

přímý přepis

$U_{i+1} = \{X \mid X \in V_N \wedge (X \rightarrow w) \in P, w \in U_i^*\}$

přepis po  $i+1$  krocích

$U_1 \subseteq U_2 \subseteq \dots \subseteq V_N + \text{stabilizace } (\exists k U_k = U_{k+1} = \dots) + U = U_k$

Automaty a gramatiky, Roman Barták

## Převod na nevypouštějící BKG - pokračování

2) Úprava pravidel

do  $P_1$  dáme pravidla tvaru  $X \rightarrow u$  taková, že:

- $u \neq \lambda$
- v  $P$  je pravidlo  $X \rightarrow v_1 Y_1 v_2 \dots v_m Y_m v_{m+1}$ ,  $Y_i \in U$ ,  $v_i \in (V_N - U) \cup V_T$ \* a  $u$  vzniká z  $(v_1 Y_1 v_2 \dots v_m Y_m v_{m+1})$  vypuštěním některých (všech, žádného) symbolů  $Y_i$ .

3) Ještě  $L(G_1) = L(G) - \{\lambda\}$

zřejmé:  $G_1$  je nevypouštějící BKG,  $L(G_1) \subseteq L(G)$ ,  $\lambda \notin L(G_1)$

necht'  $w \in L(G)$  a  $w \neq \lambda$ , tj.  $S \Rightarrow^* w$ ,

pokud se použilo pravidlo z  $P - P_1$ , pak má tvar  $X \rightarrow \lambda$

v derivaci před ním muselo být užito pravidlo  $Y \rightarrow uXv$

uděláme novou derivaci s  $Y \rightarrow uv$  a bez  $X \rightarrow \lambda$

4) Zbývá situace  $\lambda \in L(G)$

$G_2 = (V_N \cup \{S'\}, V_T, S', P_1 \cup \{S' \rightarrow \lambda, S' \rightarrow S\})$



## Příklad - nevypouštějící BKG

$S \rightarrow aSc \mid A$

$A \rightarrow bAc \mid \lambda$

1) Nejprve zjistíme neterminály, které se přepisují na  $\lambda$ :

$U = \{A, S\}$

2) Upravíme pravidla:

$S \rightarrow aSc \mid A$

$S \rightarrow ac$

(vzniklo z  $S \rightarrow aSc$  vypuštěním  $S$ )

$A \rightarrow bAc$

(pravidlo  $A \rightarrow \lambda$  nepřevádíme)

$A \rightarrow bc$

(vzniklo z  $A \rightarrow bAc$  vypuštěním  $A$ )

Původní gramatika generuje jazyk  $\{a^i b^j c^k \mid i+j=k\}$ .

Převedená gramatika generuje jazyk  $\{a^i b^j c^k \mid i+j=k, k>0\}$ .

Automaty a gramatiky, Roman Barták

## Chomského hierarchie

**gramatiky typu 0 (rekurzivně spočetné jazyky  $\mathcal{L}_0$ )**

pravidla v obecné formě

**gramatiky typu 1 (kontextové jazyky  $\mathcal{L}_1$ )**

pouze pravidla ve tvaru  $\alpha X \beta \rightarrow \alpha w \beta$ ,

$$X \in V_N, \alpha, \beta \in (V_N \cup V_T)^*, w \in (V_N \cup V_T)^+$$

jedinou výjimkou je pravidlo  $S \rightarrow \lambda$ , potom se ale S nevyskytuje na pravé straně žádného pravidla

**gramatiky typu 2 (bezkontextové jazyky  $\mathcal{L}_2$ )**

pouze pravidla ve tvaru  $X \rightarrow w$ ,  $X \in V_N, w \in (V_N \cup V_T)^+$

**gramatiky typu 3 (regulární/pravé lineární jazyky  $\mathcal{L}_3$ )**

pouze pravidla ve tvaru  $X \rightarrow wY$ ,  $X \rightarrow w$ ,  $X, Y \in V_N, w \in V_T^*$

Automaty a gramatiky, Roman Barták

## Gramatiky typu 3 a regulární jazyky

pouze pravidla ve tvaru  $X \rightarrow wY$ ,  $X \rightarrow w$ ,  $X, Y \in V_N, w \in V_T^*$

Podívejme se na derivace generované gramatikami typu 3

$$P: S \rightarrow 0S \mid 1A \mid \lambda, \quad A \rightarrow 0A \mid 1B, \quad B \rightarrow 0B \mid 1S$$

$$S \Rightarrow 0S \Rightarrow 01A \Rightarrow 011B \Rightarrow 0110B \Rightarrow 01101S \Rightarrow 01101$$

**Pozorování:**

- každé slovo derivace obsahuje právě jeden neterminál
- tento neterminál je vždy umístěn zcela vpravo
- aplikací pravidla  $X \rightarrow w$  se derivace uzavírá
- krok derivace = generuje symbol(y) + změni neterminál

**Idea vztahu gramatiky a konečného automatu:**

neterminál = stav konečného automatu

pravidla = přechodová funkce

Automaty a gramatiky, Roman Barták

## Převod konečného automatu na gramatiku

$$L \in \mathcal{F} \Rightarrow L \in \mathcal{L}_3$$

**Důkaz:**

$L = L(A)$  pro nějaký konečný automat  $A = (Q, X, \delta, q_0, F)$

definujeme gramatiku  $G = (Q, X, q_0, P)$ , kde pravidla mají tvar

$$p \rightarrow aq, \quad \text{když } \delta(p, a) = q$$

$$p \rightarrow \lambda, \quad \text{když } p \in F$$

ještě  $L(A) = L(G)$ ?

$$1) \lambda \in L(A) \Leftrightarrow q_0 \in F \Leftrightarrow (q_0 \rightarrow \lambda) \in P \Leftrightarrow \lambda \in L(G)$$

$$2) a_1 \dots a_n \in L(A) \Leftrightarrow \exists q_0, \dots, q_n \in Q \text{ tž. } \delta(q_i, a_{i+1}) = q_{i+1}, q_n \in F$$

$$\Leftrightarrow (q_0 \Rightarrow a_1 q_1 \Rightarrow \dots \Rightarrow a_1 \dots a_n q_n \Rightarrow a_1 \dots a_n) \text{ je derivace pro } a_1 \dots a_n$$

$$\Leftrightarrow a_1 \dots a_n \in L(G)$$

QED

**A co naopak?**

- pravidla  $X \rightarrow aY$  kódujeme do přechodové funkce a  $X \rightarrow \lambda$  je konec
- ale co pravidla  $X \rightarrow a_1 \dots a_n Y$ ,  $X \rightarrow Y$ ,  $X \rightarrow a_1 \dots a_n$ ?

Automaty a gramatiky, Roman Barták

## Příklad převodu KA na gramatiku

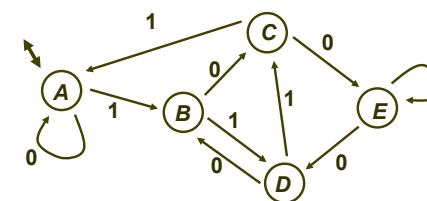
$$A \rightarrow 1B \mid 0A \mid \lambda$$

$$B \rightarrow 0C \mid 1D$$

$$C \rightarrow 0E \mid 1A$$

$$D \rightarrow 0B \mid 1C$$

$$E \rightarrow 0D \mid 1E$$



**Příklady derivací:**

$$A \Rightarrow 0A \Rightarrow 0 \quad (0)$$

$$A \Rightarrow 1B \Rightarrow 10C \Rightarrow 101A \Rightarrow 101 \quad (5)$$

$$A \Rightarrow 1B \Rightarrow 10C \Rightarrow 101A \Rightarrow 1010A \Rightarrow 1010 \quad (10)$$

$$A \Rightarrow 1B \Rightarrow 11D \Rightarrow 111C \Rightarrow 1111A \Rightarrow 1111 \quad (15)$$

$$L = \{ w \mid w \in \{0,1\}^* \wedge w \text{ je binární zápis čísla dělitelného } 5 \}$$

Automaty a gramatiky, Roman Barták

## Standardizace pravidel regulární gramatiky

Ke každé gramatice  $G=(V_N, V_T, S, P)$  typu 3 existuje ekvivalentní gramatika  $G'$ , která obsahuje pouze pravidla ve tvaru:  $X \rightarrow aY$  a  $X \rightarrow \lambda$ .

**Důkaz:**

definujeme  $G'=(V'_N, V_T, S, P')$ , kde pravidla  $P'$  získáme takto

P	P'
$X \rightarrow aY$	$X \rightarrow aY$
$X \rightarrow \lambda$	$X \rightarrow \lambda$
$X \rightarrow a_1 \dots a_n Y$	$X \rightarrow a_1 Y_2, Y_2 \rightarrow a_2 Y_3, \dots, Y_n \rightarrow a_n Y$
$Z \rightarrow a_1 \dots a_n$	$Z \rightarrow a_1 Z_1, Z_1 \rightarrow a_2 Z_2, \dots, Z_n \rightarrow \lambda$

( $Y_2, \dots, Y_n, Z_1, \dots, Z_n$  jsou nové neterminály - pro každé pravidlo jiná sada)

zbývá  $X \rightarrow Y$

definujeme  $U(X) = \{Y \mid Y \in V_N \wedge X \Rightarrow^* Y\}$

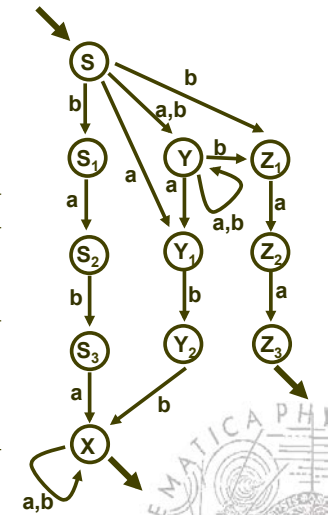
efektivní postup  $U_1 = \{Y \mid (X \rightarrow Y) \in P\}, U_{i+1} = U_i \cup \{Y \mid (Z \rightarrow Y) \in P, Z \in U_i\}$

$X \rightarrow Y$	$X \rightarrow w$ pro všechna $Z \rightarrow w$ z $P'$ a $Z \in U(X)$

Automaty a gramatiky, Roman Barták

## Příklad standardizace regulární gramatiky

Originální	Převedená
$S \rightarrow babaX$	$S \rightarrow bS_1$ $S_1 \rightarrow aS_2$ $S_2 \rightarrow bS_3$ $S_3 \rightarrow aX$
$S \rightarrow Y$	$S \rightarrow aY \mid bY \mid aY_1 \mid bZ_1$
$Y \rightarrow aY \mid bY$	$Y \rightarrow aY \mid bY$
$Y \rightarrow abbX$	$Y \rightarrow aY_1$ $Y_1 \rightarrow bY_2$ $Y_2 \rightarrow bX$
$Y \rightarrow baa$	$Y \rightarrow bZ_1$ $Z_1 \rightarrow aZ_2$ $Z_2 \rightarrow aZ_3$ $Z_3 \rightarrow \lambda$
$X \rightarrow aX \mid bX \mid \lambda$	$X \rightarrow aX \mid bX \mid \lambda$



$L = \{ w \mid w = babau \vee w = uabbb \vee w = ubaa, u, v \in \{a, b\}^* \}$

Automaty a gramatiky, Roman Barták

## Převod gramatiky na konečný automat

$L \in \mathcal{L}_3 \Rightarrow L \in \mathcal{F}$

**Důkaz:**

$L = L(G)$  pro nějakou gramatiku  $G = (V_N, V_T, S, P)$  typu 3 obsahující pouze pravidla ve tvaru:  $X \rightarrow aY$  a  $X \rightarrow \lambda$

definujeme nedeterministický konečný automat

$A = (V_N, V_T, \delta, \{S\}, F)$ , kde:

$F = \{ X \mid (X \rightarrow \lambda) \in P \}$

$\delta(X, a) = \{ Y \mid (X \rightarrow aY) \in P \}$

ještě  $L(G) = L(A)$

1)  $\lambda \in L(G) \Leftrightarrow (S \rightarrow \lambda) \in P \Leftrightarrow S \in F \Leftrightarrow \lambda \in L(A)$

2)  $a_1 \dots a_n \in L(G)$

$\Leftrightarrow$  existuje derivace  $(S \Rightarrow a_1 X_1 \Rightarrow \dots \Rightarrow a_1 \dots a_n X_n \Rightarrow a_1 \dots a_n)$

$\Leftrightarrow \exists X_0, \dots, X_n \in V_N$  tž.  $\delta(X_i, a_{i+1}) \ni X_{i+1}, X_0 = S, X_n \in F \Leftrightarrow a_1 \dots a_n \in L(A)$

Automaty a gramatiky, Roman Barták

## Levé (a pravé) lineární gramatiky

Gramatiky typu 3 nazýváme také *pravé lineární* (neterminál je vždy vpravo).

Obdobně - gramatika  $G$  je *levá lineární*, jestliže má pouze pravidla tvaru  $X \rightarrow Yw, X \rightarrow w, X, Y \in V_N, w \in V_T^*$  (neterminál je vždy vlevo).

**Věta:** Jazyky generované levou lineární gramatikou jsou právě regulární jazyky.

**Důkaz:**

- „otočením“ pravidel dostaneme pravou lineární gramatiku

$X \rightarrow Yw, X \rightarrow w$  převedeme na  $X \rightarrow w^R Y, X \rightarrow w^R$

- získaná gramatika generuje jazyk  $L^R$

- víme, že regulární jazyky jsou uzavřené na reverzi tudíž protože  $L^R$  je regulární, je i  $L = (L^R)^R$  regulární

- takto lze získat všechny regulární jazyky

Automaty a gramatiky, Roman Barták

## Lineární gramatiky (a jazyky)

Můžeme levě a právě lineární pravidla používat najednou?

Další zobecnění - **gramatika je lineární**, jestliže má pouze pravidla tvaru  $X \rightarrow uYv$ ,  $X \rightarrow w$ ,  $X, Y \in V_N$ ,  $u, v, w \in V_T^*$  (na pravé straně vždy maximálně jeden neterminál).

**Lineární jazyky** jsou právě jazyky generované lineárními gramatikami.

Zřejmě: regulární jazyky  $\subseteq$  lineární jazyky

Platí také: regulární jazyky  $\subseteq$  lineární jazyky?

**NE!**

$\{0^n 1^n \mid n \geq 1\}$  není regulární jazyk, ale je lineární ( $S \rightarrow 0S1 \mid 01$ )

**Pozorování:**

lineární pravidla lze rozložit na levě a právě lineární pravidla

$S \rightarrow 0A$ ,  $A \rightarrow S1$

Automaty a gramatiky, Roman Barták