Bezkontextové gramatiky

Bezkontextové gramatiky

Příklad:

$$
\begin{aligned}
\langle \text{STMT} \rangle &::= \langle \text{IF-STMT} \rangle \mid \langle \text{WHILE-STMT} \rangle \mid \langle \text{BEGIN-STMT} \rangle \mid \\
&\quad \langle \text{ASSG-STMT} \rangle \\
\langle \text{IF-STMT} \rangle &::= \textbf{if } \langle \text{BOOL-EXPR} \rangle \textbf{ then } \langle \text{STMT} \rangle \textbf{ else } \langle \text{STMT} \rangle \\
\langle \text{WHILE-STMT} \rangle &::= \textbf{while } \langle \text{BOOL-EXPR} \rangle \textbf{ do } \langle \text{STMT} \rangle \\
\langle \text{BEGIN-STMT} \rangle &::= \textbf{begin } \langle \text{STMT-LIST} \rangle \textbf{ end} \\
\langle \text{STMT-LIST} \rangle &::= \langle \text{STMT} \rangle \mid \langle \text{STMT} \rangle \, ; \, \langle \text{STMT-LIST} \rangle \\
\langle \text{ASSG-STMT} \rangle &::= \langle \text{VAR} \rangle := \langle \text{ARITH-EXPR} \rangle \\
\langle \text{BOOL-EXPR} \rangle &::= \langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle \\
\langle \text{COMPARE-OP} \rangle &::= < \mid > \mid \leq \mid \geq \mid = \mid \neq \\
\langle \text{ARITH-EXPR} \rangle &::= \langle \text{VAR} \rangle \mid \langle \text{CONST} \rangle \mid \\
&\quad (\langle \text{ARITH-EXPR} \rangle \langle \text{ARITH-OP} \rangle \langle \text{ARITH-EXPR} \rangle) \\
\langle \text{ARITH-OP} \rangle &::= + \mid - \mid * \mid / \\
\langle \text{CONST} \rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\
\langle \text{VAR} \rangle &::= a \mid b \mid c \mid \cdots \mid x \mid y \mid z
\end{aligned}
$$

Bezkontextové gramatiky

Symbolům, které mají v předchozím příkladě tvar $\langle \text{XXX} \rangle$, se říká **neterminální symboly** (**neterminály**).

Pravidla popisují, jaké řetězce může daný neterminál reprezentovat.

Z neterminálu $\langle \text{STMT} \rangle$ můžeme například dostat text

$$\textbf{while } x \leq y \textbf{ do begin } x := (x + 1)\,;\, y := (y - 1) \textbf{ end}$$

Poznámka: Výše použité notaci se říká Backus-Naurova forma (BNF).

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

⟨STMT⟩
⟨WHILE-STMT⟩
while ⟨BOOL-EXPR⟩ **do** ⟨STMT⟩
while ⟨ARITH-EXPR⟩⟨COMPARE-OP⟩⟨ARITH-EXPR⟩ **do** ⟨STMT⟩
while ⟨VAR⟩⟨COMPARE-OP⟩⟨ARITH-EXPR⟩ **do** ⟨STMT⟩

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq y$ **do** $\langle \text{STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle \text{STMT} \rangle$
$\langle \text{WHILE-STMT} \rangle$
while $\langle \text{BOOL-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{ARITH-EXPR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \langle \text{COMPARE-OP} \rangle \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{ARITH-EXPR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $\langle \text{VAR} \rangle \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq \langle \text{VAR} \rangle$ **do** $\langle \text{STMT} \rangle$
while $x \leq y$ **do** $\langle \text{STMT} \rangle$
while $x \leq y$ **do** $\langle \text{BEGIN-STMT} \rangle$

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

$\langle$STMT$\rangle$
$\langle$WHILE-STMT$\rangle$
while $\langle$BOOL-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$ARITH-EXPR$\rangle\langle$COMPARE-OP$\rangle\langle$ARITH-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$VAR$\rangle\langle$COMPARE-OP$\rangle\langle$ARITH-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$VAR$\rangle \leq \langle$ARITH-EXPR$\rangle$ **do** $\langle$STMT$\rangle$
while $\langle$VAR$\rangle \leq \langle$VAR$\rangle$ **do** $\langle$STMT$\rangle$
while $x \leq \langle$VAR$\rangle$ **do** $\langle$STMT$\rangle$
while $x \leq y$ **do** $\langle$STMT$\rangle$
while $x \leq y$ **do** $\langle$BEGIN-STMT$\rangle$
while $x \leq y$ **do begin** $\langle$STMT-LIST$\rangle$ **end**

Bezkontextové gramatiky

while $x \leq y$ **do begin** $x := (x + 1)\,;\; y := (y - 1)$ **end**

⟨STMT⟩
⟨WHILE-STMT⟩
while ⟨BOOL-EXPR⟩ **do** ⟨STMT⟩
while ⟨ARITH-EXPR⟩⟨COMPARE-OP⟩⟨ARITH-EXPR⟩ **do** ⟨STMT⟩
while ⟨VAR⟩⟨COMPARE-OP⟩⟨ARITH-EXPR⟩ **do** ⟨STMT⟩
while ⟨VAR⟩ $\leq$ ⟨ARITH-EXPR⟩ **do** ⟨STMT⟩
while ⟨VAR⟩ $\leq$ ⟨VAR⟩ **do** ⟨STMT⟩
while $x \leq$ ⟨VAR⟩ **do** ⟨STMT⟩
while $x \leq y$ **do** ⟨STMT⟩
while $x \leq y$ **do** ⟨BEGIN-STMT⟩
while $x \leq y$ **do begin** ⟨STMT-LIST⟩ **end**
...

Bezkontextové gramatiky

Formálně je **bezkontextová gramatika** definována jako čtveřice

$$G = (\Pi, \Sigma, S, P)$$

kde:

- Π je konečná množina **neterminálních symbolů** (**neterminálů**)
- Σ je konečná množina **terminálních symbolů** (**terminálů**), přičemž $\Pi \cap \Sigma = \emptyset$
- $S \in \Pi$ je **počáteční neterminál**
- $P \subseteq \Pi \times (\Pi \cup \Sigma)^*$ je konečná množina **přepisovacích pravidel**

Bezkontextové gramatiky

Poznámky:

- Pro označení neterminálních symbolů budeme používat velká písmena A, B, C, ...
- Pro označení terminálních symbolů budeme používat malá písmena a, b, c, ... nebo číslice 0, 1, 2, ...
- Pro označení řetězců z $(\Pi \cup \Sigma)^*$ budeme používat malá písmena řecké abecedy α, β, γ, ...
- Místo zápisu (A, α) budeme pro pravidla používat zápis

$$A \to \alpha$$

A – levá strana pravidla
α – pravá strana pravidla

Bezkontextové gramatiky

Příklad: Gramatika $G = (\Pi, \Sigma, S, P)$, kde

- $\Pi = \{A, B, C\}$
- $\Sigma = \{a, b\}$
- $S = A$
- P obsahuje pravidla

$$A \to aBBb$$
$$A \to AaA$$
$$B \to \varepsilon$$
$$B \to bCA$$
$$C \to AB$$
$$C \to a$$
$$C \to b$$

Bezkontextové gramatiky

Poznámka: Pokud máme více pravidel se stejnou levou stranou, jako třeba

$$A \to \alpha_1 \qquad A \to \alpha_2 \qquad A \to \alpha_3$$

můžeme je stručněji zapsat jako

$$A \to \alpha_1 \mid \alpha_2 \mid \alpha_3$$

Například pravidla dříve uvedené gramatiky můžeme zapsat jako

$$\begin{aligned} A &\to aBBb \mid AaA \\ B &\to \varepsilon \mid bCA \\ C &\to AB \mid a \mid b \end{aligned}$$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo *abbabb* je možné v gramatice G vygenerovat následujícím způsobem:

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

A

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$\begin{aligned} \underline{A} &\to aBBb \mid AaA \\ B &\to \varepsilon \mid bCA \\ C &\to AB \mid a \mid b \end{aligned}$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$\underline{A}$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$\underline{A} \longrightarrow \underline{aBBb} \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$\underline{A} \Rightarrow \underline{aBBb}$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow a\underline{B}Bb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow a\underline{B}Bb \Rightarrow a\underline{bCA}Bb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$\underline{A} \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$\underline{A} \longrightarrow \underline{aBBb} \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb \Rightarrow abC\underline{aBBb}Bb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$\underline{C} \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$\underline{C} \longrightarrow AB \mid a \mid \underline{b}$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb \Rightarrow ab\underline{b}aBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \longrightarrow aBBb \mid AaA$$
$$B \longrightarrow \varepsilon \mid bCA$$
$$C \longrightarrow AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb \Rightarrow abbabb$

Bezkontextové gramatiky

Gramatiky slouží ke generování slov.

Příklad: $G = (\Pi, \Sigma, A, P)$, kde $\Pi = \{A, B, C\}$, $\Sigma = \{a, b\}$ a P obsahuje pravidla

$$A \to aBBb \mid AaA$$
$$B \to \varepsilon \mid bCA$$
$$C \to AB \mid a \mid b$$

Například slovo $abbabb$ je možné v gramatice G vygenerovat následujícím způsobem:

$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb \Rightarrow abbabb$

Bezkontextové gramatiky

Řetězce z $(\Pi \cup \Sigma)^*$ nazýváme **větné formy**.

Na větných formách definujeme relaci $\Rightarrow \subseteq (\Pi \cup \Sigma)^* \times (\Pi \cup \Sigma)^*$ takovou, že

$$\alpha \Rightarrow \alpha'$$

právě když $\alpha = \beta_1 A \beta_2$ a $\alpha' = \beta_1 \gamma \beta_2$ pro nějaká $\beta_1, \beta_2, \gamma \in (\Pi \cup \Sigma)^*$ a $A \in \Pi$, kde $(A \to \gamma) \in P$.

Příklad: Jestliže $(B \to bCA) \in P$, pak

$$aCBbA \Rightarrow aCbCAbA$$

Poznámka: Neformálně řečeno zápis $\alpha \Rightarrow \alpha'$ znamená, že z větné formy α je možné jedním krokem odvodit větnou formu α, a to tak, že výskyt nějakého neterminálu A v α nahradíme pravou stranou nějakého pravidla $A \to \alpha$, kde se A vyskytuje na levé straně.

Bezkontextové gramatiky

Řetězce z $(\Pi \cup \Sigma)^*$ nazýváme **větné formy**.

Na větných formách definujeme relaci $\Rightarrow \subseteq (\Pi \cup \Sigma)^* \times (\Pi \cup \Sigma)^*$ takovou, že

$$\alpha \Rightarrow \alpha'$$

právě když $\alpha = \beta_1 A \beta_2$ a $\alpha' = \beta_1 \gamma \beta_2$ pro nějaká $\beta_1, \beta_2, \gamma \in (\Pi \cup \Sigma)^*$ a $A \in \Pi$, kde $(A \to \gamma) \in P$.

Příklad: Jestliže $(B \to bCA) \in P$, pak

$$aC\underline{B}bA \Rightarrow aC\underline{bCA}bA$$

Poznámka: Neformálně řečeno zápis $\alpha \Rightarrow \alpha'$ znamená, že z větné formy α je možné jedním krokem odvodit větnou formu α, a to tak, že výskyt nějakého neterminálu A v α nahradíme pravou stranou nějakého pravidla $A \to \alpha$, kde se A vyskytuje na levé straně.

Derivace délky n větné formy α' z větné formy α je posloupnost větných forem

$$\beta_0, \beta_1, \beta_2, \cdots, \beta_n$$

takových, že

- $\alpha = \beta_0$
- $\beta_{i-1} \Rightarrow \beta_i$ pro všechna $i \in \{1, 2, \ldots, n\}$
- $\alpha' = \beta_n$

což můžeme stručněji zapsat jako

$$\alpha = \beta_0 \Rightarrow \beta_1 \Rightarrow \beta_2 \Rightarrow \ldots \Rightarrow \beta_{n-1} \Rightarrow \beta_n = \alpha'$$

Bezkontextové gramatiky

Skutečnost, že pro dané n existuje nějaká derivace délky n větné formy α' z větné formy α, označujeme zápisem

$$\alpha \Rightarrow^n \alpha'$$

Skutečnost, že existuje nějaká derivace (nějaké délky n, kde $n \geq 0$) větné formy α' z větné formy α, označujeme zápisem

$$\alpha \Rightarrow^* \alpha'$$

Poznámka: Relace $\Rightarrow^*$ je reflexivním a tranzitivním uzávěrem relace $\Rightarrow$ (tj. nejmenší reflexivní a tranzitivní relací obsahující relaci $\Rightarrow$).

Bezkontextové gramatiky

Jazyk $L(G)$ generovaný gramatikou $G = (\Pi, \Sigma, S, P)$ je množina všech slov v abecedě Σ, která lze odvodit nějakou derivací z počátečního neterminálu S pomocí pravidel z P, tj.

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk

$$L = \{a^n b^n \mid n \geq 0\}$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk

$$L = \{a^n b^n \mid n \geq 0\}$$

Gramatika $G = (\Pi, \Sigma, S, P)$, kde $\Pi = \{S\}$, $\Sigma = \{a, b\}$ a P obsahuje

$$S \rightarrow aSb \mid \varepsilon$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk

$$L = \{a^n b^n \mid n \geq 0\}$$

Gramatika $G = (\Pi, \Sigma, S, P)$, kde $\Pi = \{S\}$, $\Sigma = \{a, b\}$ a P obsahuje

$$S \to aSb \mid \varepsilon$$

$S \Rightarrow \varepsilon$
$S \Rightarrow aSb \Rightarrow ab$
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaaSbbbb \Rightarrow aaaabbbb$
$\ldots$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk tvořený všemi palindromy nad abecedou $\{a, b\}$, tj.

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

Poznámka: w^R označuje tzv. **zrcadlový obraz** slova w, tj. slovo w zapsané pozpátku.

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk tvořený všemi palindromy nad abecedou $\{a, b\}$, tj.

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

Poznámka: w^R označuje tzv. **zrcadlový obraz** slova w, tj. slovo w zapsané pozpátku.

Řešení:

$$S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk tvořený všemi palindromy nad abecedou $\{a, b\}$, tj.

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

Poznámka: w^R označuje tzv. **zrcadlový obraz** slova w, tj. slovo w zapsané pozpátku.

Řešení:

$$S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$$

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abaSaba \Rightarrow abaaaba$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře uzávorkovanými sekvencemi symbolů '(' a ')'.

Například $(()())(()) \in L$, ale $)()) \notin L$.

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře uzávorkovanými sekvencemi symbolů '(' a ')'.

Například $(()())(()) \in L$, ale $)()) \notin L$.

Řešení:

$$S \to \varepsilon \mid (S) \mid SS$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře uzávorkovanými sekvencemi symbolů '(' a ')'.

Například $(()())(()) \in L$, ale $)()) \notin L$.

Řešení:

$$S \to \varepsilon \mid (S) \mid SS$$

$S \Rightarrow SS \Rightarrow (S)S \Rightarrow (S)(S) \Rightarrow (SS)(S) \Rightarrow ((S)S)(S) \Rightarrow$
$(()S)(S) \Rightarrow (()(S))(S) \Rightarrow (()())(S) \Rightarrow (()())((S)) \Rightarrow$
$(()())(())$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře vytvořenými aritmetickými výrazy, kde operandy jsou vždy tvaru 'a', a kde jako operátory můžeme používat symboly $+$ a $*$.

Například $(a + a) * a + (a * a) \in L$.

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře vytvořenými aritmetickými výrazy, kde operandy jsou vždy tvaru 'a', a kde jako operátory můžeme používat symboly $+$ a $*$.

Například $(a + a) * a + (a * a) \in L$.

Řešení:

$$E \longrightarrow a \mid E + E \mid E * E \mid (E)$$

Bezkontextové gramatiky

Příklad: Chceme vytvořit gramatiku generující jazyk L tvořený všemi dobře vytvořenými aritmetickými výrazy, kde operandy jsou vždy tvaru 'a', a kde jako operátory můžeme používat symboly $+$ a $*$.

Například $(a + a) * a + (a * a) \in L$.

Řešení:

$$E \rightarrow a \mid E + E \mid E * E \mid (E)$$

$$E \Rightarrow E+E \Rightarrow E*E+E \Rightarrow (E)*E+E \Rightarrow (a+a)*E+E \Rightarrow (a+a)*a+E \Rightarrow$$
$$(a+a)*a+(E) \Rightarrow (a+a)*a+(E*E) \Rightarrow (a+a)*a+(a*E) \Rightarrow (a+a)*a+(a*a)$$

Derivační strom

$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$

Derivační strom

A

$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$

A

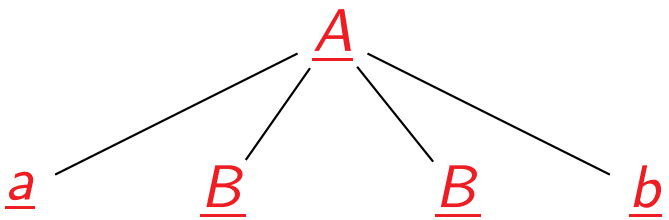
Derivační strom

$\underline{A}$

$\underline{A} \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$

$\underline{A}$

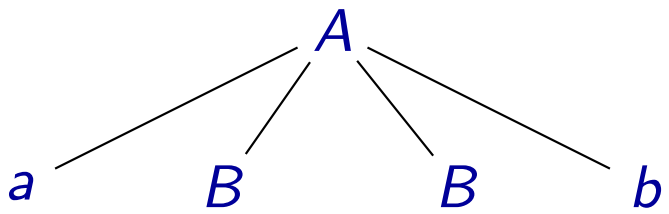
Derivační strom



$\underline{A} \longrightarrow \underline{aBBb} \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$

$\underline{A} \Rightarrow \underline{aBBb}$

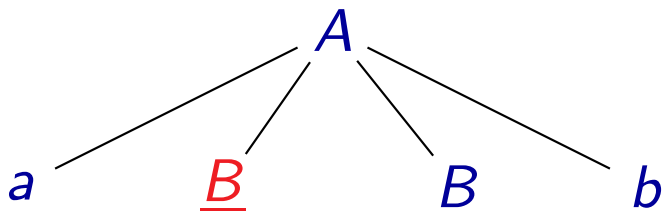
Derivační strom



$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$

$A \Rightarrow aBBb$

Derivační strom

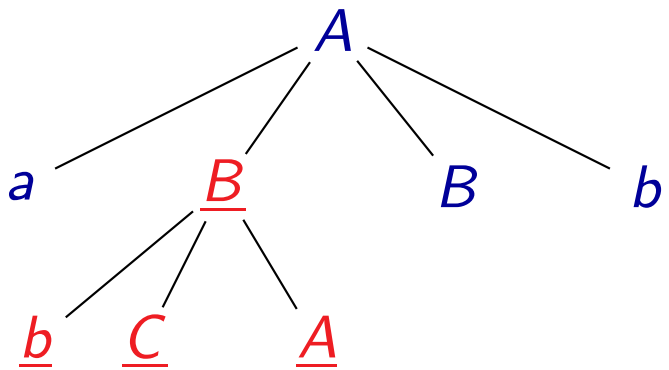


$A \longrightarrow aBBb \mid AaA$
$\underline{B} \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$

$A \Rightarrow a\underline{B}Bb$

Derivační strom

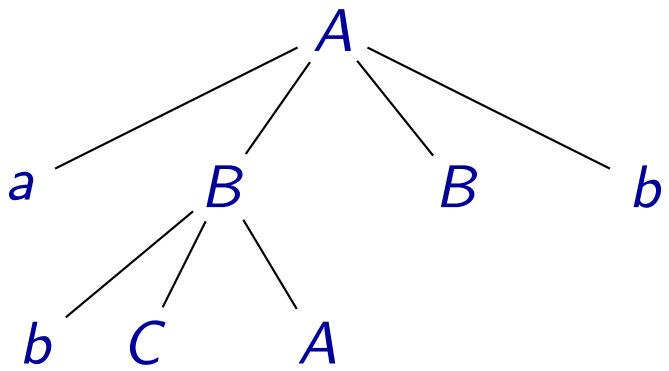
$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb$

Derivační strom

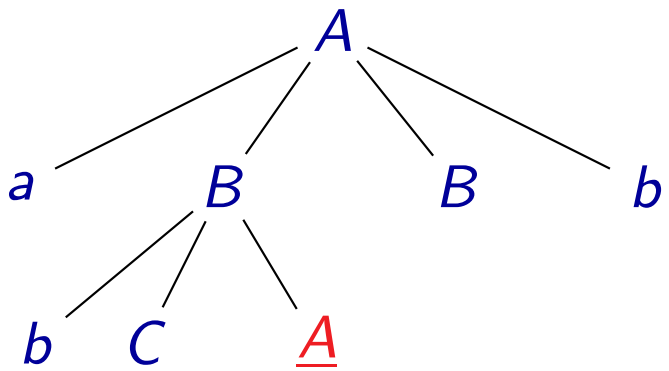
$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb$

Derivační strom

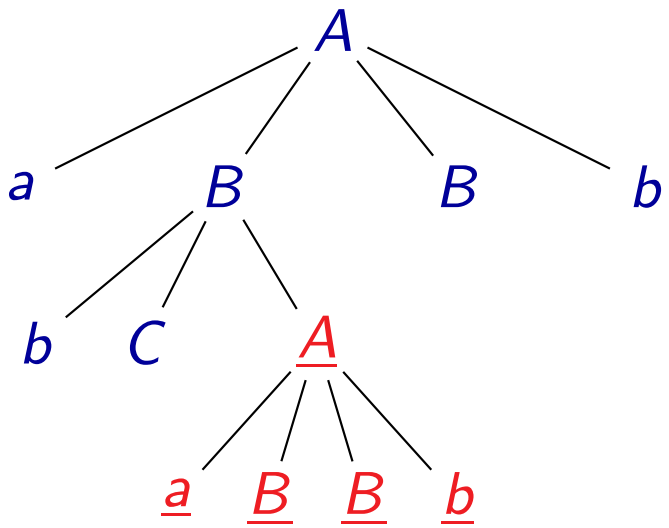
$\underline{A} \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb$

Derivační strom

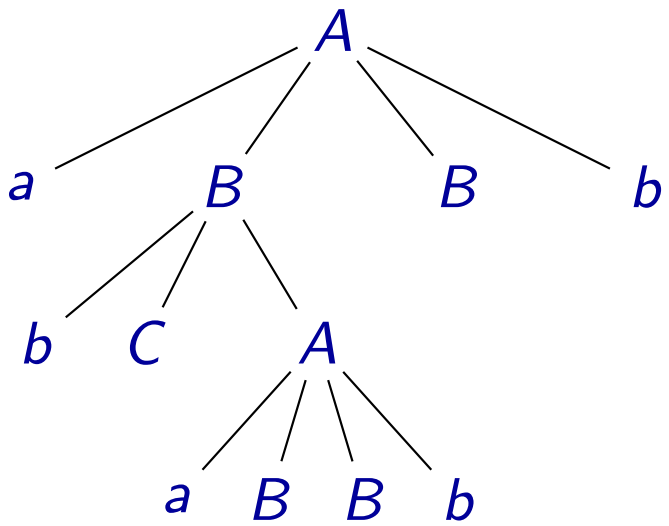
$\underline{A} \longrightarrow \underline{aBBb} \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abC\underline{A}Bb \Rightarrow abC\underline{aBBb}Bb$

Derivační strom

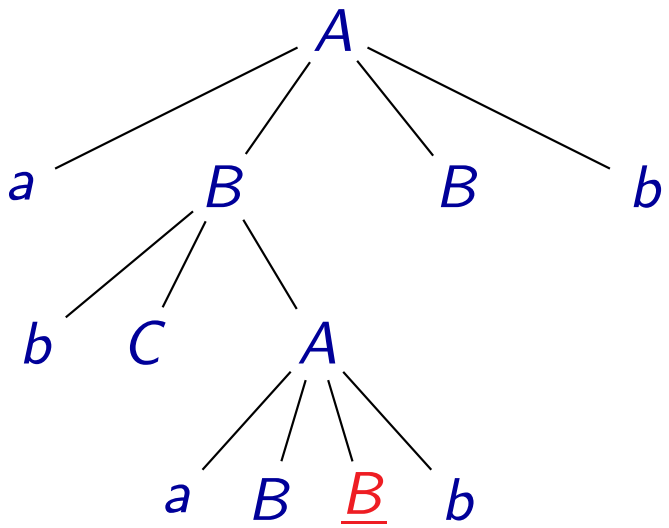
$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb$

Derivační strom

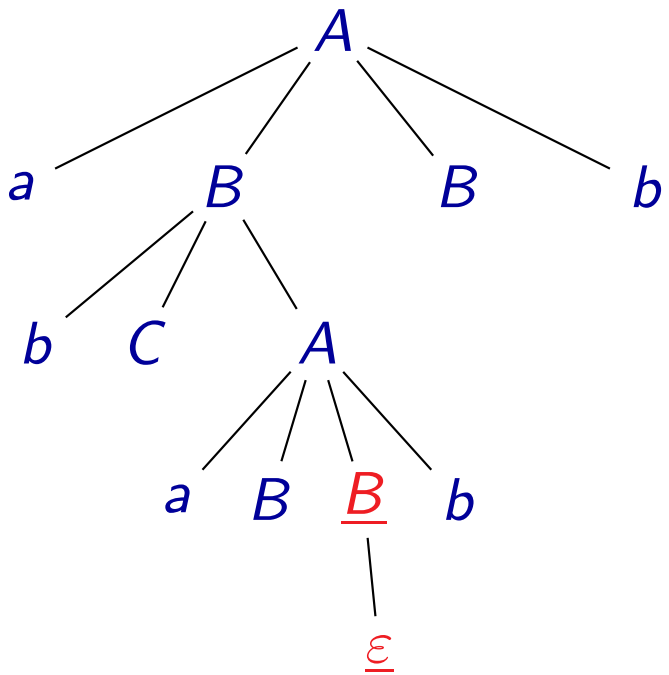
$A \longrightarrow aBBb \mid AaA$
$\underline{B} \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb$

Derivační strom

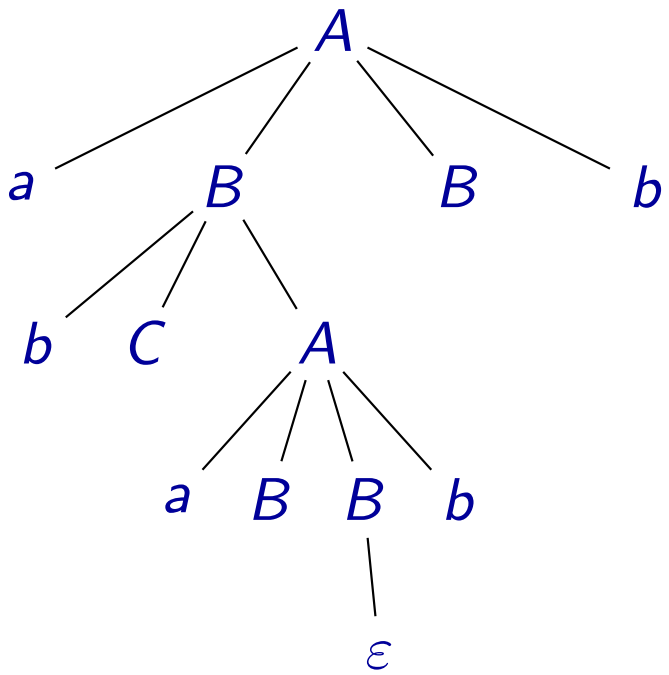
$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaB\underline{B}bBb \Rightarrow abCaBbBb$

Derivační strom

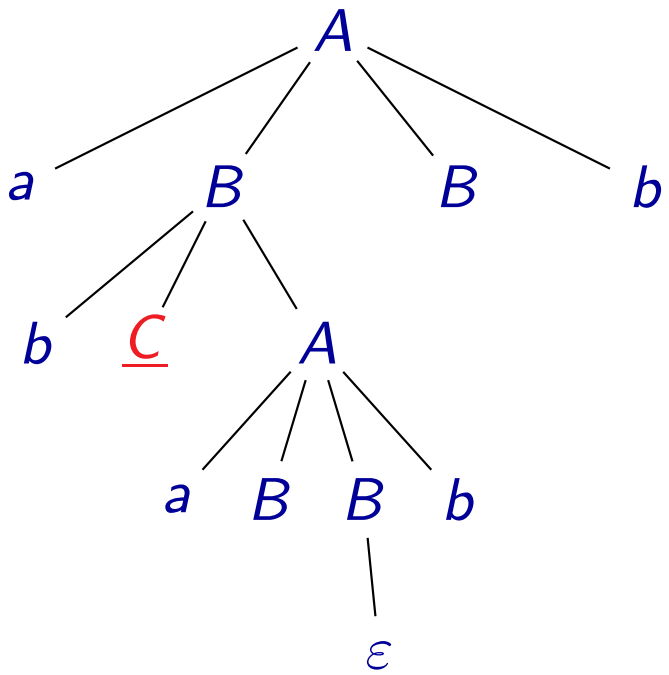
$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb$

Derivační strom

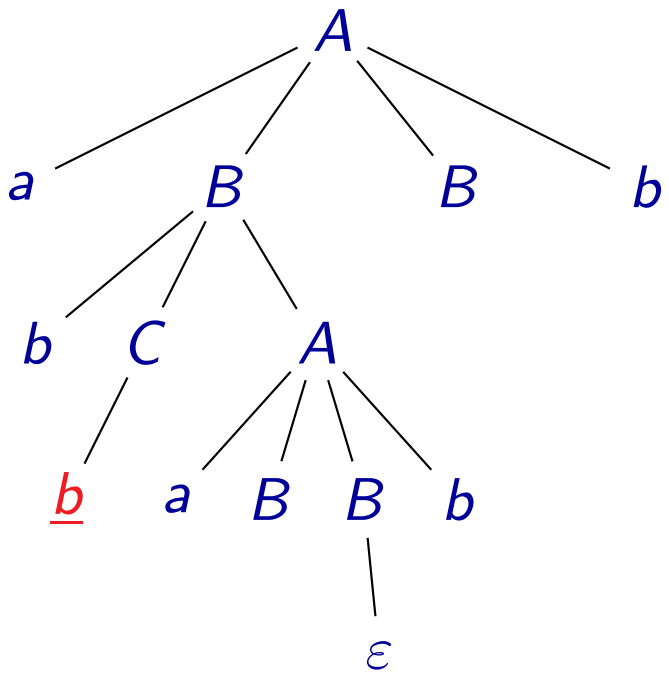
$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$\underline{C} \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb$

Derivační strom

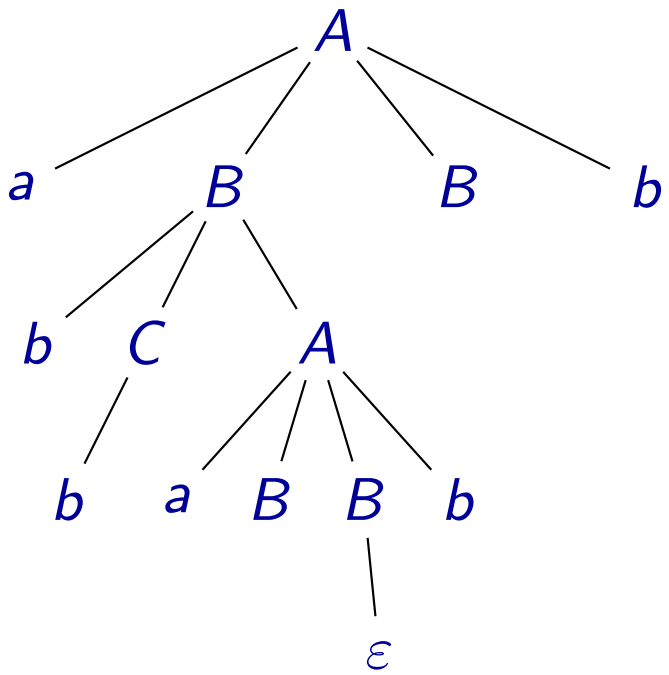
$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$\underline{C} \longrightarrow AB \mid a \mid \underline{b}$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow ab\underline{C}aBbBb \Rightarrow ab\underline{b}aBbBb$

Derivační strom

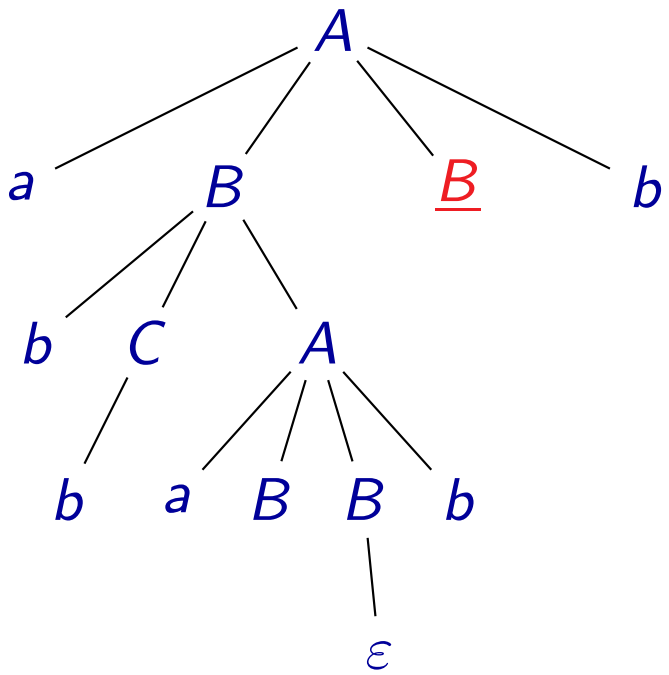
$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb$

Derivační strom

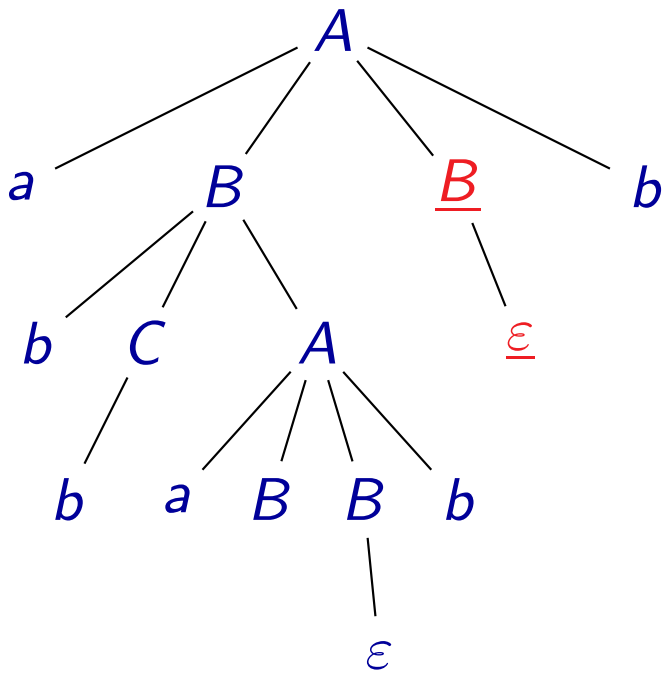
$A \longrightarrow aBBb \mid AaA$
$\underline{B} \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b$

Derivační strom

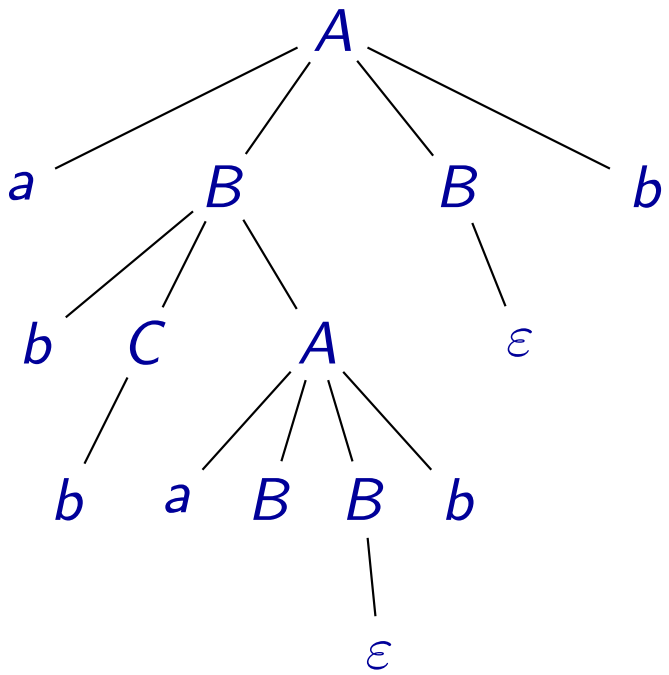
$A \longrightarrow aBBb \mid AaA$
$\underline{B} \longrightarrow \underline{\varepsilon} \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBb\underline{B}b \Rightarrow abbaBbb$

Derivační strom

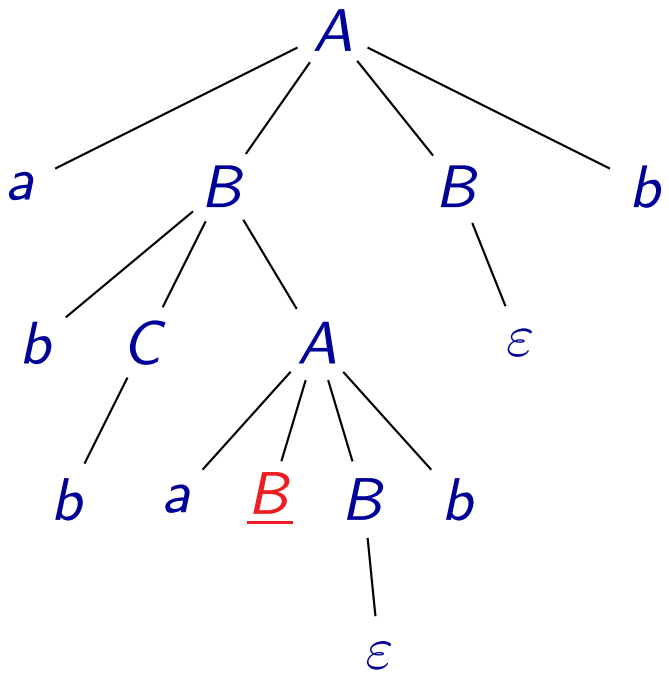
$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb$

Derivační strom

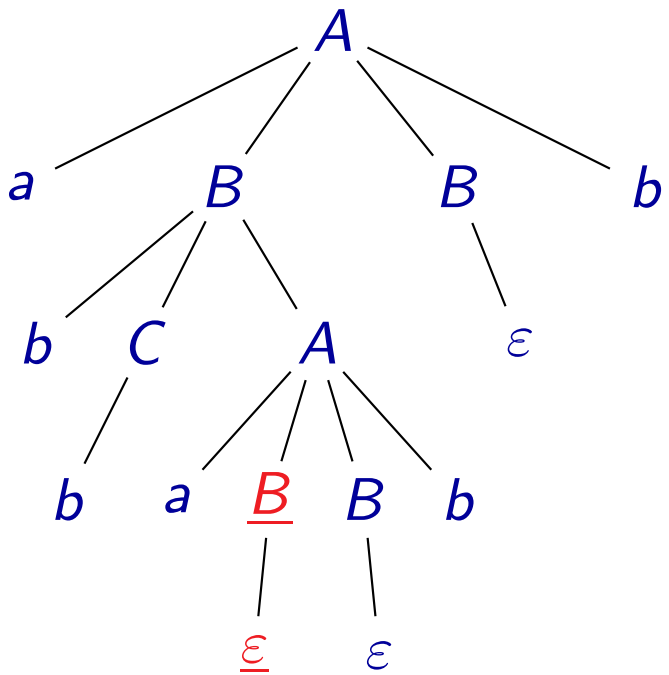
$A \longrightarrow aBBb \mid AaA$
$\underline{B} \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abba\underline{B}bb$

Derivační strom

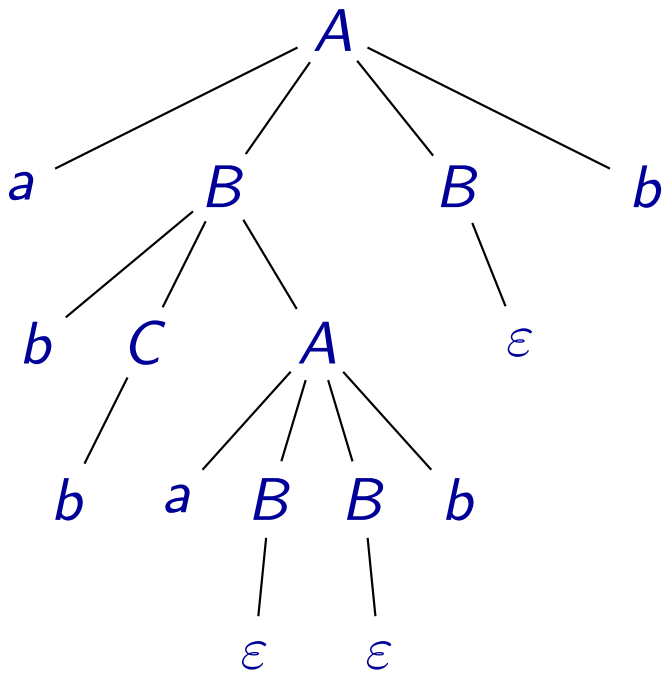
$A \longrightarrow aBBb \mid AaA$
$\underline{B} \longrightarrow \underline{\varepsilon} \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abba\underline{B}bb \Rightarrow abbabb$

Derivační strom

$A \longrightarrow aBBb \mid AaA$
$B \longrightarrow \varepsilon \mid bCA$
$C \longrightarrow AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow abbaBbb \Rightarrow abbabb$

Derivační strom

Každé derivaci odpovídá nějaký **derivační strom:**

- Vrcholy stromu jsou ohodnoceny terminály a neterminály.
- Kořen stromu je ohodnocen počátečním neterminálem.
- Listy stromu jsou ohodnoceny terminály nebo symboly ε.
- Ostatní vrcholy stromu jsou ohodnoceny neterminály.
- Pokud je vrchol ohodnocen neterminálem A, pak jeho potomci jsou ohodnoceni symboly pravé strany nějakého přepisovacího pravidla $A \to \alpha$.

Levá a pravá derivace

$$E \rightarrow E + E \mid E * E \mid (E) \mid a$$

Levá derivace je derivace, ve které v každém kroku nahrazujeme vždy nejlevější neterminál.

$$\underline{E} \Rightarrow \underline{E} + E \Rightarrow \underline{E} * E + E \Rightarrow a * \underline{E} + E \Rightarrow a * a + \underline{E} \Rightarrow a * a + a$$

Pravá derivace je derivace, ve které v každém kroku nahrazujeme vždy nejpravější neterminál.

$$\underline{E} \Rightarrow E + \underline{E} \Rightarrow \underline{E} + a \Rightarrow E * \underline{E} + a \Rightarrow \underline{E} * a + a \Rightarrow a * a + a$$

Derivace však nemusí být ani levá ani pravá:

$$\underline{E} \Rightarrow \underline{E} + E \Rightarrow E * \underline{E} + E \Rightarrow E * a + \underline{E} \Rightarrow \underline{E} * a + a \Rightarrow a * a + a$$

Levá a pravá derivace

- Jednomu derivačnímu stromu může odpovídat více různých derivací.
- Každému derivačnímu stromu odpovídá právě jedna levá a právě jedna pravá derivace.

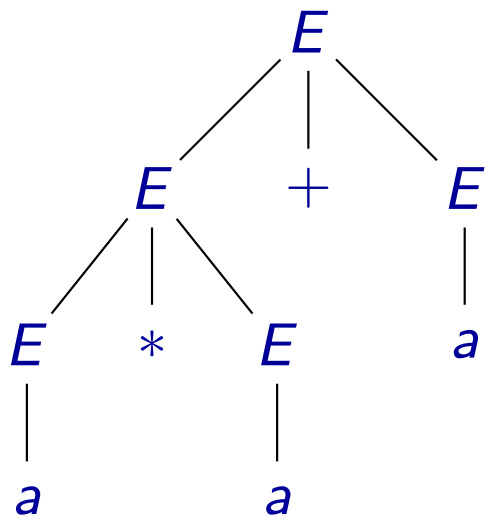
Nejednoznačné gramatiky

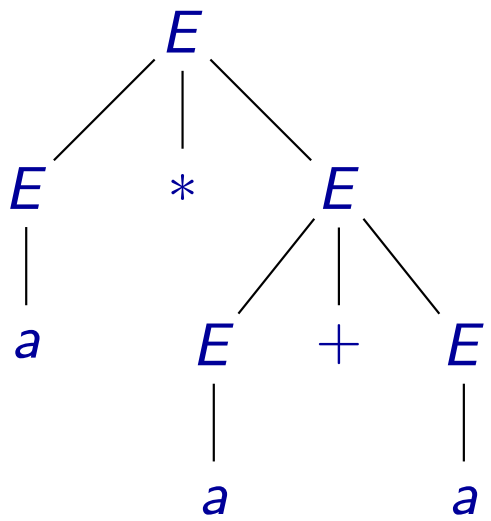
Gramatika G je **nejednoznačná**, jestliže existuje nějaké slovo $w \in L(G)$, kterému přísluší dva různé derivační stromy, resp. dvě různé levé či dvě různé pravé derivace.

Příklad:

$E \Rightarrow E + E \Rightarrow E * E + E \Rightarrow a * E + E \Rightarrow a * a + E \Rightarrow a * a + a$

$E \Rightarrow E * E \Rightarrow E * E + E \Rightarrow a * E + E \Rightarrow a * a + E \Rightarrow a * a + a$





Nejednoznačné gramatiky

Někdy je možné nejednoznačnou gramatiku nahradit gramatikou, která generuje tentýž jazyk, ale není nejednoznačná.

Příklad: Gramatiku

$$E \to E + E \mid E * E \mid (E) \mid a$$

můžeme nahradit ekvivalentní gramatikou

$$\begin{aligned} E &\to T \mid T + E \\ T &\to F \mid F * T \\ F &\to a \mid (E) \end{aligned}$$

Poznámka: Pokud se nejednoznačná gramatika žádnou ekvivalentní jednoznačnou gramatikou nahradit nedá, říkáme, že je **podstatně nejednoznačná**.

Ekvivalence gramatik

Gramatiky G_1 a G_2 jsou **ekvivalentní**, jestliže generují tentýž jazyk, tj. jestliže $L(G_1) = L(G_2)$.

Poznámka: Problém ekvivalence bezkontextových gramatik je algoritmicky nerozhodnutelný. Dá se dokázat, že není možné vytvořit algoritmus, který by pro libovolné dvě bezkontextové gramatiky rozhodl, zda jsou ekvivalentní či ne.

Dokonce je algoritmicky nerozhodnutelný i problém, zda gramatika generuje jazyk Σ^*.

Bezkontextové jazyky

Definice

Jazyk L je **bezkontextový**, jestliže existuje bezkontextová gramatika G taková, že $L = L(G)$.

Třída bezkontextových jazyků je uzavřená vůči:

- zřetězení
- sjednocení
- iteraci

Třída bezkontextových jazyků však není uzavřená vůči:

- doplňku
- průniku

Bezkontextové jazyky

Máme dány gramatiky $G_1 = (\Pi_1, \Sigma, S_1, P_1)$ a $G_2 = (\Pi_2, \Sigma, S_2, P_2)$, přičemž můžeme předpokládat, že $\Pi_1 \cap \Pi_2 = \emptyset$ a $S \notin \Pi_1 \cup \Pi_2$.

- Gramatika G taková, že $L(G) = L(G_1)L(G_2)$:

$$G = (\Pi_1 \cup \Pi_2 \cup \{S\},\ \Sigma,\ S,\ P_1 \cup P_2 \cup \{S \to S_1 S_2\})$$

- Gramatika G taková, že $L(G) = L(G_1) \cup L(G_2)$:

$$G = (\Pi_1 \cup \Pi_2 \cup \{S\},\ \Sigma,\ S,\ P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\})$$

- Gramatika G taková, že $L(G) = L(G_1)^*$:

$$G = (\Pi_1 \cup \{S\},\ \Sigma,\ S,\ P_1 \cup \{S \to \varepsilon, S \to S_1 S\})$$

Redukované gramatiky

Definice

Bezkontextová gramatika $G = (\Pi, \Sigma, S, P)$ je **redukovaná**, jestliže:

- Každý neterminál $X \in \Pi$ je možné přepsat na sekvenci terminálů, tj. existuje $w \in \Sigma^*$ takové, že $X \Rightarrow^* w$.
- Každý neterminál $X \in \Pi$ je dosažitelný z počátečního neterminálu S, tj. existují $\alpha, \beta \in (\Pi \cup \Sigma)^*$ takové, že $S \Rightarrow^* \alpha X \beta$.

Ke každé bezkontextové gramatice je možné sestrojit ekvivalentní redukovanou gramatiku.

Redukované gramatiky

Algoritmus, který zkonstruuje množinu

$$\mathcal{T} = \{X \in \Pi \mid \exists w \in \Sigma^* : X \Rightarrow^* w\}$$

```
1  T ← ∅
2  change ← TRUE
3  while change
4        do change ← FALSE
5           for each (X → α) ∈ P
6                do if X ∉ T and α ∈ (T ∪ Σ)*
7                   then T ← T ∪ {X}
8                        change ← TRUE
9  return T
```

Z gramatiky G pak vytvoříme gramatiku G' tak, že z G odstraníme všechny neterminály nepatřící do $\mathcal{T}$ a pravidla, kde se vyskytují.

Redukované gramatiky

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

$$
\begin{aligned}
S &\to A \mid B \\
A &\to aB \mid bS \mid b \\
B &\to AB \mid Ba \\
C &\to AS \mid b
\end{aligned}
$$

Redukované gramatiky

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

$$
\begin{aligned}
S &\to A \mid B \\
A &\to aB \mid bS \mid b \\
B &\to AB \mid Ba \\
C &\to AS \mid b
\end{aligned}
$$

$$\mathcal{T} = \{$$

Redukované gramatiky

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

$$
\begin{aligned}
S &\to A \mid B \\
A &\to aB \mid bS \mid b \\
B &\to AB \mid Ba \\
C &\to AS \mid b
\end{aligned}
$$

$$\mathcal{T} = \{A,$$

Redukované gramatiky

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

$$
\begin{aligned}
S &\to A \mid B \\
A &\to aB \mid bS \mid b \\
B &\to AB \mid Ba \\
C &\to AS \mid b
\end{aligned}
$$

$$\mathcal{T} = \{A, C,$$

Redukované gramatiky

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

$$
\begin{aligned}
S &\to A \mid B \\
A &\to aB \mid bS \mid b \\
B &\to AB \mid Ba \\
C &\to AS \mid b
\end{aligned}
$$

$$\mathcal{T} = \{A, C, S$$

Redukované gramatiky

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

$$
\begin{aligned}
S &\to A \mid B \\
A &\to aB \mid bS \mid b \\
B &\to AB \mid Ba \\
C &\to AS \mid b
\end{aligned}
$$

$$\mathcal{T} = \{A, C, S\}$$

Redukované gramatiky

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

$$
\begin{aligned}
S &\to A \mid B \\
A &\to aB \mid bS \mid b \\
B &\to AB \mid Ba \\
C &\to AS \mid b
\end{aligned}
$$

$$\mathcal{T} = \{A, C, S\}$$

Gramatika $G' = (\{S, A, C\}, \{a, b\}, S, P')$

$$
\begin{aligned}
S &\to A \\
A &\to bS \mid b \\
C &\to AS \mid b
\end{aligned}
$$

Redukované gramatiky

Algoritmus, který zkonstruuje množinu

$$\mathcal{D} = \{X \in \Pi' \mid \exists \alpha, \beta \in (\Pi' \cup \Sigma)^* : S \Rightarrow^* \alpha X \beta\}$$

```
1  D ← {S}
2  change ← TRUE
3  while change
4        do change ← FALSE
5           for each (X → α) ∈ P' where X ∈ D
6                do for i ← 1 to |α|
7                         do if α[i] ∈ Π' and α[i] ∉ D
8                               then D ← D ∪ {α[i]}
9                                    change ← TRUE
10 return D
```

Z gramatiky G' pak vytvoříme gramatiku G'' tak, že z G' odstraníme všechny neterminály nepatřící do $\mathcal{D}$ a pravidla, kde se vyskytují.

Redukované gramatiky

Příklad: Gramatika $G' = (\{S, A, C\}, \{a, b\}, S, P')$

$$
\begin{aligned}
S &\to A \\
A &\to bS \mid b \\
C &\to AS \mid b
\end{aligned}
$$

Redukované gramatiky

Příklad: Gramatika $G' = (\{S, A, C\}, \{a, b\}, S, P')$

$$S \to A$$
$$A \to bS \mid b$$
$$C \to AS \mid b$$

$$\mathcal{D} = \{$$

Redukované gramatiky

Příklad: Gramatika $G' = (\{S, A, C\}, \{a, b\}, S, P')$

$$S \to A$$
$$A \to bS \mid b$$
$$C \to AS \mid b$$

$$\mathcal{D} = \{S,$$

Redukované gramatiky

Příklad: Gramatika $G' = (\{S, A, C\}, \{a, b\}, S, P')$

$$S \to A$$
$$A \to bS \mid b$$
$$C \to AS \mid b$$

$$\mathcal{D} = \{S, A$$

Redukované gramatiky

Příklad: Gramatika $G' = (\{S, A, C\}, \{a, b\}, S, P')$

$$S \to A$$
$$A \to bS \mid b$$
$$C \to AS \mid b$$

$$\mathcal{D} = \{S, A\}$$

Redukované gramatiky

Příklad: Gramatika $G' = (\{S, A, C\}, \{a, b\}, S, P')$

$$S \to A$$
$$A \to bS \mid b$$
$$C \to AS \mid b$$

$$\mathcal{D} = \{S, A\}$$

Gramatika $G'' = (\{S, A\}, \{a, b\}, S, P'')$

$$S \to A$$
$$A \to bS \mid b$$

Redukované gramatiky

Pořadí obou kroků je třeba dodržet. Pokud bychom je provedli v opačném pořadí, můžeme dostat gramatiku, která není redukovaná.

Příklad:

$$S \to a \mid A$$
$$A \to AB$$
$$B \to b$$

Pokud provedeme oba kroky algoritmu ve správném pořadí, dostaneme

$$S \to a$$

Pokud provedeme kroky algoritmu v opačném pořadí, dostaneme

$$S \to a$$
$$B \to b$$

Redukované gramatiky

Předchozí algoritmus lze použít ke zjištění, zda $L(G) \neq \emptyset$.

Stačí ověřit, zda $S \in \mathcal{T}$:

- Pokud ano, je $L(G) \neq \emptyset$.
- Pokud ne, je $L(G) = \emptyset$.

Nevypouštějící gramatiky

Definice

Gramatika G je **nevypouštějící**, jestliže neobsahuje žádná ε-pravidla, tj. pravidla tvaru $X \rightarrow \varepsilon$, kde $X \in \Pi$.

Ke každé bezkontextové gramatice je možné sestrojit nevypouštějící gramatiku G' takovou, že $L(G') = L(G) - \{\varepsilon\}$.

Nevypouštějící gramatiky

Algoritmus, který zkonstruuje množinu

$$\mathcal{E} = \{X \in \Pi \mid X \Rightarrow^* \varepsilon\}$$

```
1  E ← ∅
2  change ← TRUE
3  while change
4        do change ← FALSE
5           for each (X → α) ∈ P
6                 do if X ∉ E and α ∈ E*
7                       then E ← E ∪ {X}
8                            change ← TRUE
9  return E
```

Nevypouštějící gramatiky

Nevypouštějící gramatiku G' pak z G vytvoříme tak, že pro každé pravidlo $X \to \alpha$ z G přidáme do G' všechna možná pravidla

$$X \to \alpha'$$

kde α' vznikne z α vypuštěním libovolného počtu symbolů z $\mathcal{E}$, přičemž ale $\alpha' \neq \varepsilon$.

Příklad: Pokud například $\mathcal{E} = \{A, C, D\}$, pak místo pravidla

$$A \to aASCbA$$

přidáme pravidla

$$A \to aASCbA \mid aSCbA \mid aASbA \mid aASCb \mid aSbA \mid aSCb \mid aASb \mid aSb$$

Nevypouštějící gramatiky

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

$$
\begin{aligned}
S &\to AB \mid \varepsilon \\
A &\to aAAb \mid BS \mid CA \\
B &\to BbA \mid CaC \mid \varepsilon \\
C &\to aBB \mid bS
\end{aligned}
$$

Nevypouštějící gramatiky

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

$$
\begin{aligned}
S &\to AB \mid \varepsilon \\
A &\to aAAb \mid BS \mid CA \\
B &\to BbA \mid CaC \mid \varepsilon \\
C &\to aBB \mid bS
\end{aligned}
$$

$$\mathcal{E} = \{$$

Nevypouštějící gramatiky

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

$$
\begin{aligned}
S &\to AB \mid \varepsilon \\
A &\to aAAb \mid BS \mid CA \\
B &\to BbA \mid CaC \mid \varepsilon \\
C &\to aBB \mid bS
\end{aligned}
$$

$$\mathcal{E} = \{S,$$

Nevypouštějící gramatiky

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

$$
\begin{aligned}
S &\to AB \mid \varepsilon \\
A &\to aAAb \mid BS \mid CA \\
B &\to BbA \mid CaC \mid \varepsilon \\
C &\to aBB \mid bS
\end{aligned}
$$

$$\mathcal{E} = \{S, B,$$

Nevypouštějící gramatiky

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

$$
\begin{aligned}
S &\to AB \mid \varepsilon \\
A &\to aAAb \mid BS \mid CA \\
B &\to BbA \mid CaC \mid \varepsilon \\
C &\to aBB \mid bS
\end{aligned}
$$

$$\mathcal{E} = \{S, B, A$$

Nevypouštějící gramatiky

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

$$
\begin{aligned}
S &\to AB \mid \varepsilon \\
A &\to aAAb \mid BS \mid CA \\
B &\to BbA \mid CaC \mid \varepsilon \\
C &\to aBB \mid bS
\end{aligned}
$$

$$\mathcal{E} = \{S, B, A\}$$

Nevypouštějící gramatiky

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

$$
\begin{aligned}
S &\to AB \mid \varepsilon \\
A &\to aAAb \mid BS \mid CA \\
B &\to BbA \mid CaC \mid \varepsilon \\
C &\to aBB \mid bS
\end{aligned}
$$

$$\mathcal{E} = \{S, B, A\}$$

Gramatika $G' = (\{S, A, B, C\}, \{a, b\}, S, P')$

$$
\begin{aligned}
S &\to AB \mid A \mid B \\
A &\to aAAb \mid aAb \mid ab \mid BS \mid B \mid S \mid CA \mid C \\
B &\to BbA \mid bA \mid Bb \mid b \mid CaC \\
C &\to aBB \mid aB \mid a \mid bS \mid b
\end{aligned}
$$

Chomského normální forma

Definice

Gramatika G je v **Chomského normální formě**, jestliže všechna její pravidla jsou pouze následujícího tvaru:

- $X \to YZ$, kde $X, Y, Z \in \Pi$, nebo
- $X \to a$, kde $X \in \Pi$ a $a \in \Sigma$.

Ke každé bezkontextové gramatice je možné sestrojit gramatiku G' v Chomského normální formě takovou, že $L(G') = L(G) - \{\varepsilon\}$.

Greibachové normální forma

Definice

Gramatika G je v **Greibachové normální formě**, jestliže všechna její pravidla jsou pouze následujícího tvaru:

- $X \to a\alpha$, kde $a \in \Sigma$, $X \in \Pi$ a $\alpha \in \Pi^*$.

Ke každé bezkontextové gramatice je možné sestrojit gramatiku G' v Greibachové normální formě takovou, že $L(G') = L(G) - \{\varepsilon\}$.

Regulární gramatiky

Definice

Gramatika G je **regulární**, jestliže všechna její pravidla jsou tvaru:

- $X \to wY$, kde $X, Y \in \Pi$, $w \in \Sigma^*$, nebo
- $X \to w$, kde $X \in \Pi$, $w \in \Sigma^*$.

Regulární gramatiky generují právě třídu regulárních jazyků, tj.:

- Jazyk generovaný regulární gramatikou je vždy regulární.
- Každý regulární jazyk je generován nějakou regulární gramatikou.