Neregulární jazyky

Neregulární jazyky

Ne všechny jazyky jsou regulární.

Existují jazyky, pro které neexistuje žádný konečný automat, který by je rozpoznával.

Příklady neregulárních jazyků:

- $L_1 = \{a^n b^n \mid n \geq 0\}$
- $L_2 = \{ww \mid w \in \{a, b\}^*\}$
- $L_3 = \{ww^R \mid w \in \{a, b\}^*\}$

Poznámka: Existence neregulárních jazyků vyplývá již z faktu, že automatů pracujících nad nějakou abecedou Σ je jen spočetně mnoho, zatímco jazyků nad abecedou Σ je nespočetně mnoho.

Neregulární jazyky

Jak dokázat o nějakém jazyce L, že není regulární?

Jazyk není regulární, jestliže neexistuje (tj. není možné sestrojit) konečný automat, který by ho rozpoznával.

Jak ale dokázat, že něco neexistuje?

Neregulární jazyky

Jak dokázat o nějakém jazyce L, že není regulární?

Jazyk není regulární, jestliže neexistuje (tj. není možné sestrojit) konečný automat, který by ho rozpoznával.

Jak ale dokázat, že něco neexistuje?

Odpověď: Stačí ukázat, že jazyk L nemá nějakou vlastnost, kterou má každý jazyk rozpoznávaný nějakým konečným automatem.

Neregulární jazyky

Jak dokázat o nějakém jazyce L, že není regulární?

Jazyk není regulární, jestliže neexistuje (tj. není možné sestrojit) konečný automat, který by ho rozpoznával.

Jak ale dokázat, že něco neexistuje?

Odpověď: Stačí ukázat, že jazyk L nemá nějakou vlastnost, kterou má každý jazyk rozpoznávaný nějakým konečným automatem.

Dva základní postupy dokazování neregularity jazyků:

- využití tzv. pumping lemmatu
- využití Myhillovy-Nerodovy věty (nebudeme se jí dále zabývat)

Neregulární jazyky

Tvrzení

Každý konečný jazyk je regulární.

Důkaz: Konečný jazyk s n slovy můžeme popsat regulárním výrazem $w_1 + w_2 + \ldots + w_n$. Jazyk je tedy regulární.

Neregulární jazyky

Tvrzení

Každý konečný jazyk je regulární.

Důkaz: Konečný jazyk s n slovy můžeme popsat regulárním výrazem $w_1 + w_2 + \ldots + w_n$. Jazyk je tedy regulární.

Tvrzení

Je-li jazyk L nekonečný, obsahuje pro každou konstantu $k \in \mathbb{N}$ nějaké slovo w takové, že $|w| > k$

Důkaz: Uvažujeme jen konečnou abecedu. Pro každou konstantu k je tedy jen konečně mnoho slov kratších než k. Protože L je nekonečný, musí obsahovat nějaké slovo delší než k.

Pumping Lemma

Předpokládejme, že jazyk L je rozpoznáván nějakým konkrétním deterministickým konečným automatem A, tj. $L = L(A)$.

Vezměme nyní nějaké libovolné slovo $z \in L$, kde $z = a_1 a_2 \cdots a_k$.

Protože automat A slovo z přijímá, musí tomuto slovu odpovídat určitý přijímající výpočet automatu, tj. posloupnost stavů:

$$q_0,\ q_1,\ q_2,\ \ldots,\ q_{k-1},\ q_k$$

délky $k+1$, kde

- q_0 je počáteční stav
- $\delta(q_{i-1}, a_i) = q_i$ pro $\forall i \in \{1, 2, \ldots, k\}$
- q_k je přijímající stav

Pumping Lemma

Předpokládejme, že A má n stavů a že $|z| \geq n$.

Pak máme posloupnost délky minimálně $n+1$, ve které se může vyskytovat maximálně n různých stavů.

Z toho plyne, že musí existovat alespoň jeden stav q, který se v této posloupnosti vyskytuje alespoň dvakrát.

Jde o aplikaci tzv. **holubníkového principu (pigeonhole principle)**.

Holubníkový princip

Jestliže mám $n+1$ holubů rozmístěných do n klecí, pak jsou alespoň v jedné kleci minimálně dva holubi.

Pumping Lemma

Řekněme, že opakující stav se vyskytuje na pozicích i, j, tj. $q_i = q_j$, kde $i < j$.

$$q_0, \cdots, q_i, \cdots, q_j, \cdots, q_k$$

Poznámka: Zjevně můžeme najít taková i, j, že $i < j \leq n$

Slovo z můžeme rozdělit na tři části:

$$\underbrace{a_1 \cdots a_i}_{u} \quad \underbrace{a_{i+1} \cdots a_j}_{v} \quad \underbrace{a_{j+1} \cdots a_k}_{w}$$

- $\delta^*(q_0, u) = q_i$
- $\delta^*(q_i, v) = q_j = q_i$
- $\delta^*(q_j, w) = q_k$

Pumping Lemma

Vezměme nyní slova:

$$\underbrace{a_1 \cdots a_i}_{u} \quad \underbrace{a_{j+1} \cdots a_k}_{w}$$

$$\underbrace{a_1 \cdots a_i}_{u} \quad \underbrace{a_{i+1} \cdots a_j}_{v} \quad \underbrace{a_{i+1} \cdots a_j}_{v} \quad \underbrace{a_{j+1} \cdots a_k}_{w}$$

$$\underbrace{a_1 \cdots a_i}_{u} \quad \underbrace{a_{i+1} \cdots a_j}_{v} \quad \underbrace{a_{i+1} \cdots a_j}_{v} \quad \underbrace{a_{i+1} \cdots a_j}_{v} \quad \underbrace{a_{j+1} \cdots a_k}_{w}$$

$$\ldots$$

Je zřejmé, že A přijme každé z nich, vzhledem k tomu, že

- $\delta^*(q_0, u) = q_i$
- $\delta^*(q_i, v) = q_j = q_i$
- $\delta^*(q_j, w) = q_k$, $q_k \in F$

Pumping Lemma

Pumping Lemma

Jestliže jazyk L je regulární, pak existuje n takové, že každé slovo $z \in L$ takové, že $|z| \geq n$, je možné rozdělit na podslova u, v, w taková, že $z = uvw$, $|uv| \leq n$, $|v| \geq 1$ a pro všechna $i \geq 0$ platí $uv^i w \in L$.

Formálně zapsáno:

Jestliže L je regulární, pak

$(\exists n)(\forall z \text{ tž. } z \in L, |z| \geq n)(\exists u, v, w \text{ tž. } z = uvw, |uv| \leq n, |v| \geq 1)$
$(\forall i \geq 0) : uv^i w \in L$

Pumping Lemma

Tvrzení je možné obrátit. ($A \Rightarrow B$ je totéž, co $\neg B \Rightarrow \neg A$.)

Jestliže

$(\forall n)(\exists z \text{ tž. } z \in L, |z| \geq n)(\forall u, v, w \text{ tž. } z = uvw, |uv| \leq n, |v| \geq 1)$
$(\exists i \geq 0) : uv^i w \notin L,$

pak L není regulární.

Pokud tedy chceme ukázat, že jazyk L není regulární, stačí ukázat, že splňuje výše uvedenou podmínku.

Pumping Lemma

Příklad: Uvažujme jazyk $L = \{a^i b^i \mid i \geq 0\}$.

- Předpokládáme, že L je rozpoznáván nějakým deterministickým konečným automatem s n stavy.
- Zvolme slovo $z = a^n b^n$.
- Uvažujme všechny možnosti, jak může být z rozděleno na podslova u, v, w splňující podmínky $|uv| \leq n$ a $|v| \geq 1$.
  Zjevně slova u a v obsahují pouze symboly a. Pro každé konkrétní rozdělení existují nějaká j a k taková, že $j + k \leq n$, $k \geq 1$ a
  - $u = a^j$
  - $v = a^k$
  - $w = a^{n-(j+k)} b^n$
- Pokud nyní zvolíme $i = 0$, dostáváme $uv^i w = uw = a^{n-k} b^n$. Protože $n - k < n$, zjevně platí $uv^i w \notin L$.

Bezkontextové gramatiky

Bezkontextové gramatiky

Příklad:

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

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

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

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**

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

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** $\langle \text{BLOCK-STMT} \rangle$

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** $\langle$BLOCK-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**

$\langle$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** $\langle$BLOCK-STMT$\rangle$
while $x \leq y$ **do begin** $\langle$STMT-LIST$\rangle$ **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

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

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

$\underline{A}$

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

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

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

$A \Rightarrow aBBb$

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

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

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

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

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

$A \Rightarrow aBBb \Rightarrow abCABb$

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:

$A \Rightarrow aBBb \Rightarrow abCABb$

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

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

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

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

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

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

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

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

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

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

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

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

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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.

Bezkontextové gramatiky

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

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

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:

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

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

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

Derivační strom

A

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

A

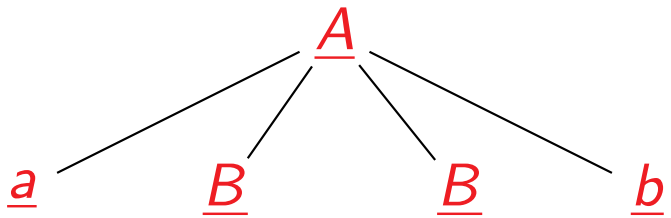
Derivační strom

$\underline{A}$

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

$\underline{A}$

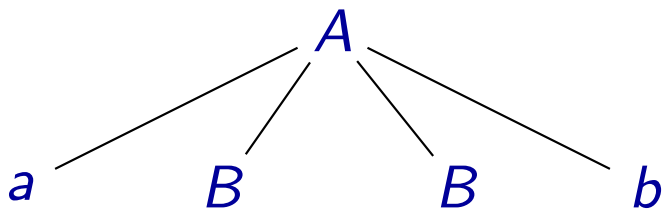
Derivační strom



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

$A \Rightarrow aBBb$

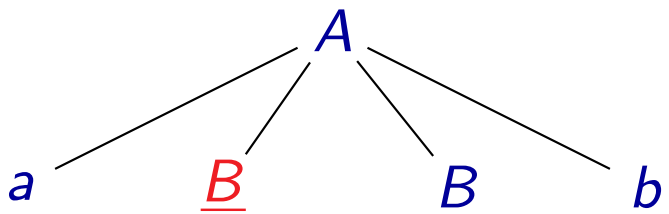
Derivační strom



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

$A \Rightarrow aBBb$

Derivační strom

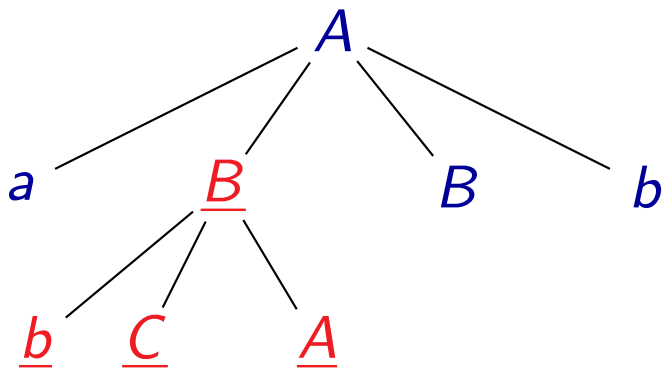


$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$

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

Derivační strom

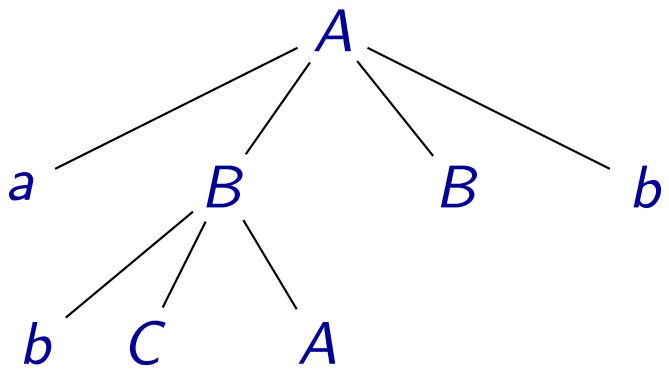
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb$

Derivační strom

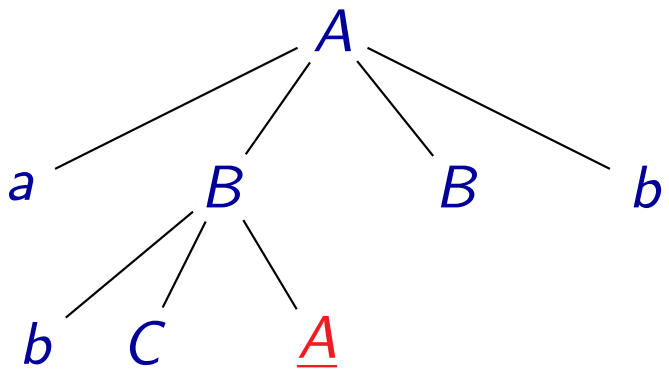
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



$A \Rightarrow aBBb \Rightarrow abCABb$

Derivační strom

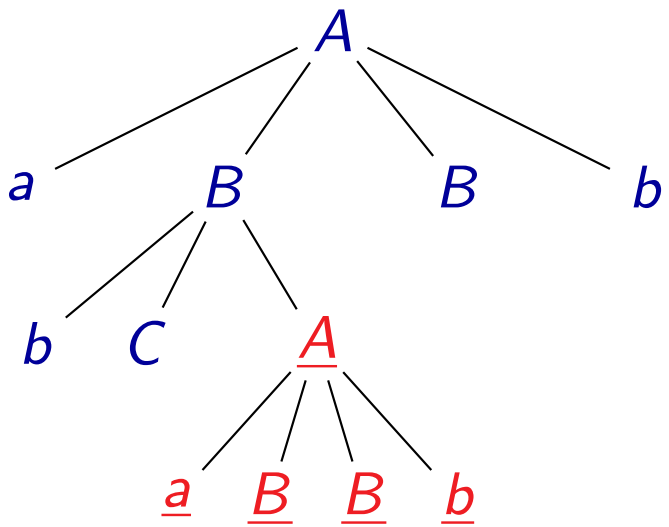
$\underline{A} \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

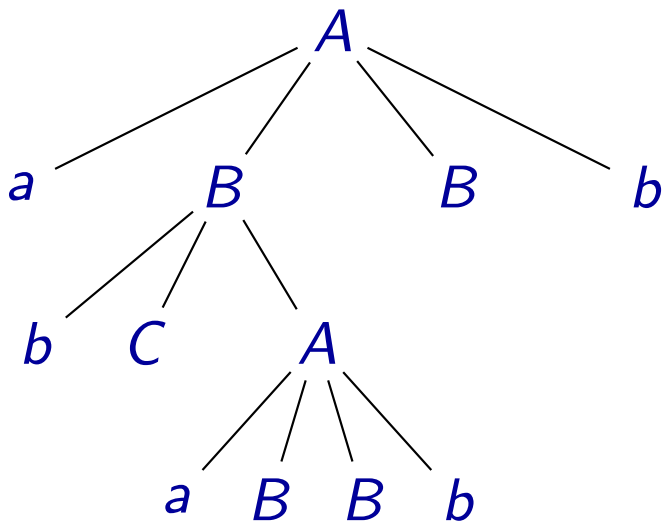
$\underline{A} \to \underline{aBBb} \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

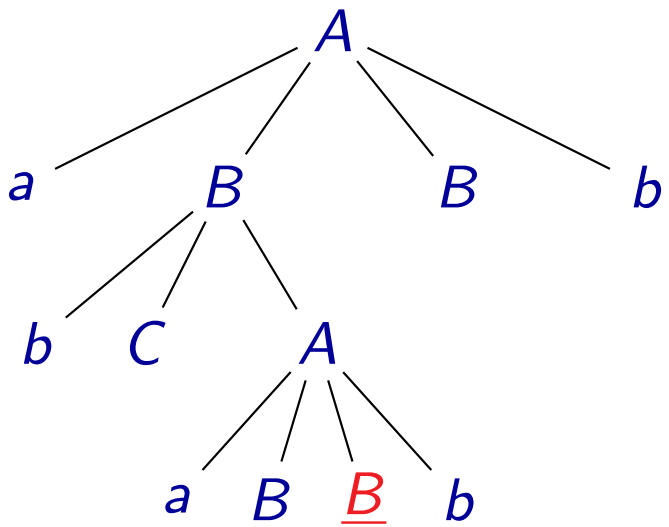
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

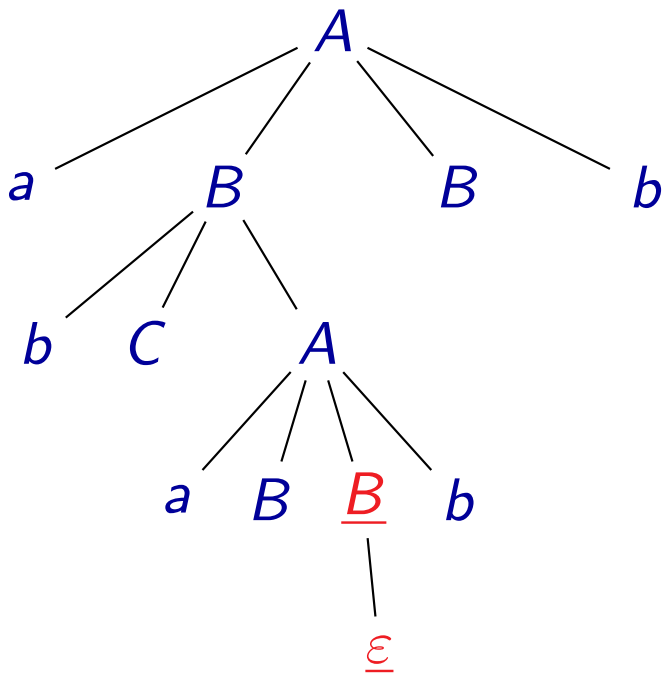
$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

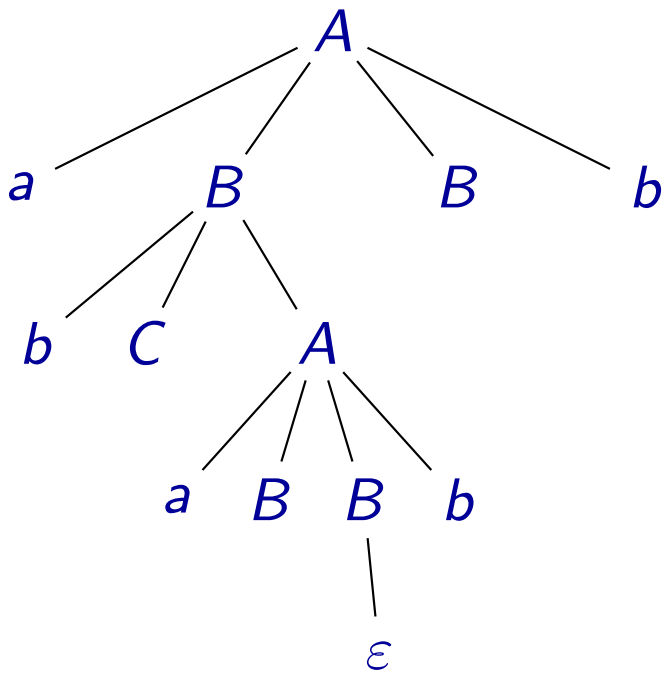
$A \to aBBb \mid AaA$
$\underline{B} \to \underline{\varepsilon} \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

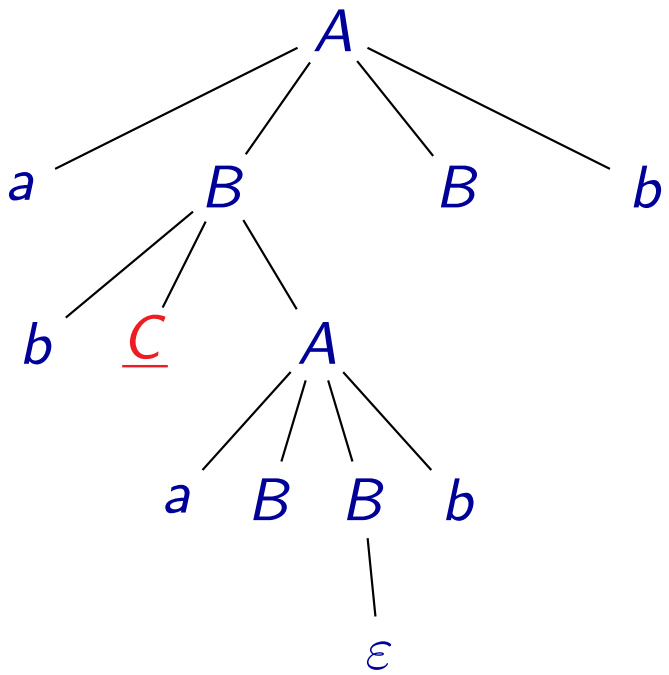
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

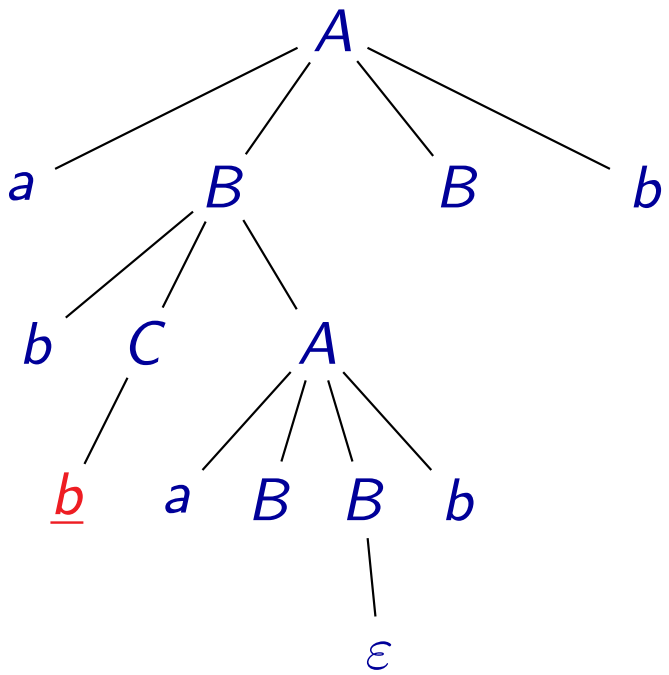
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$\underline{C} \to AB \mid a \mid b$



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

Derivační strom

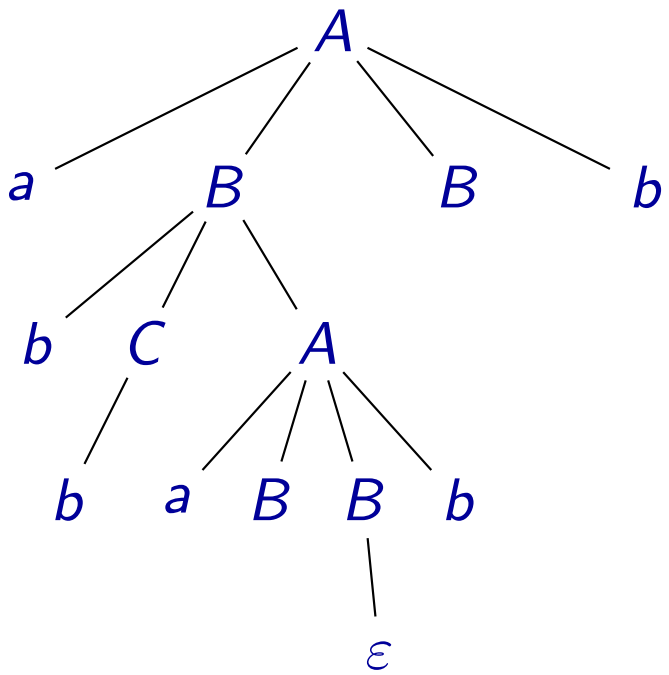
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$\underline{C} \to AB \mid a \mid \underline{b}$



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

Derivační strom

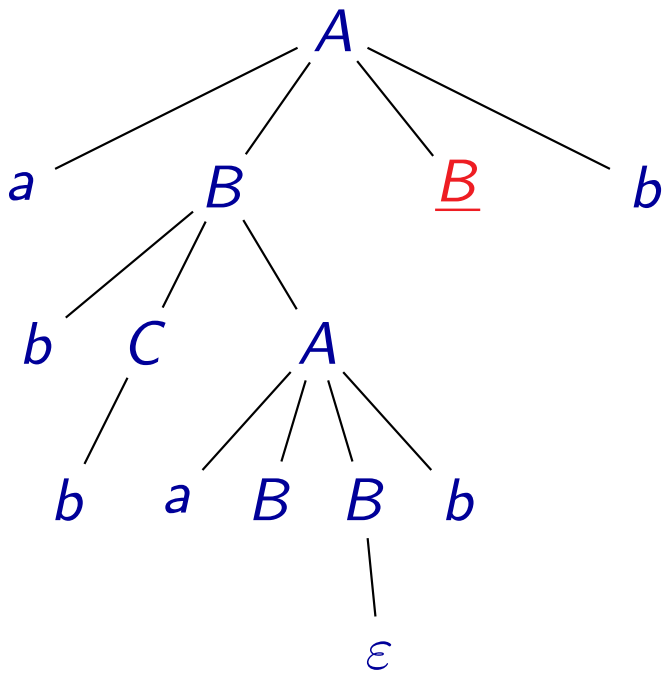
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

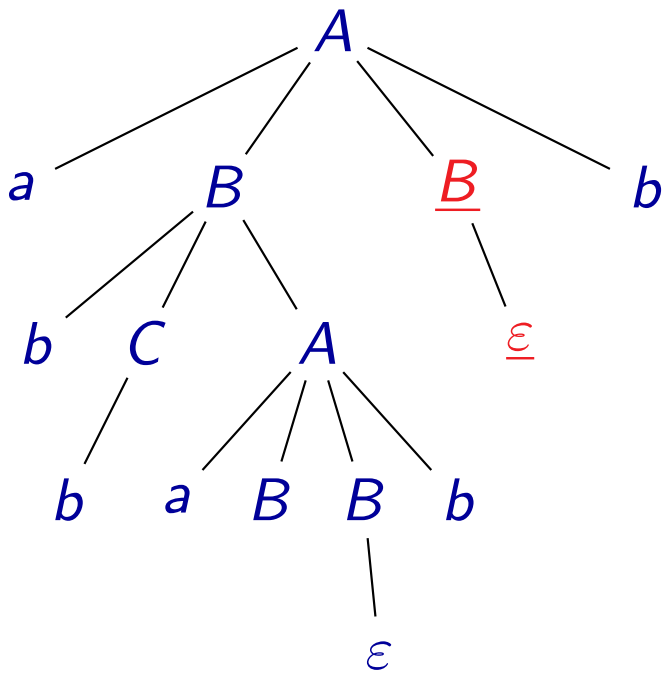
$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

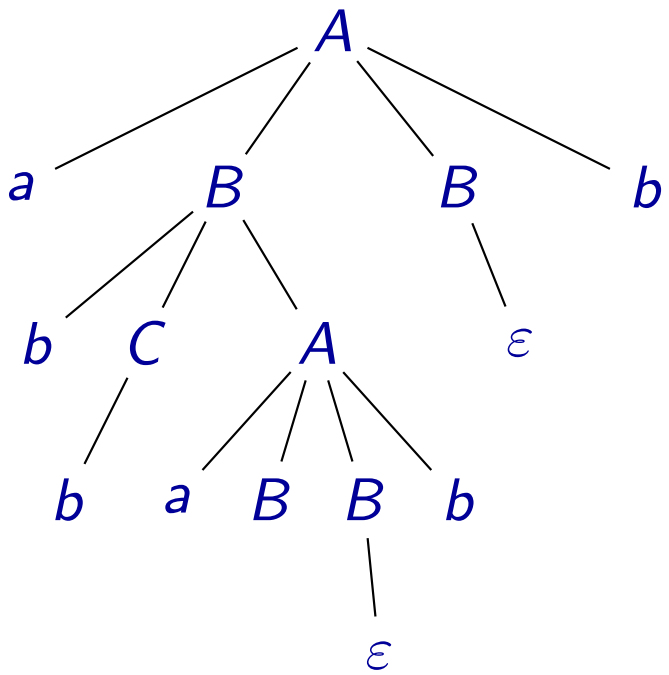
$A \to aBBb \mid AaA$
$\underline{B} \to \underline{\varepsilon} \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

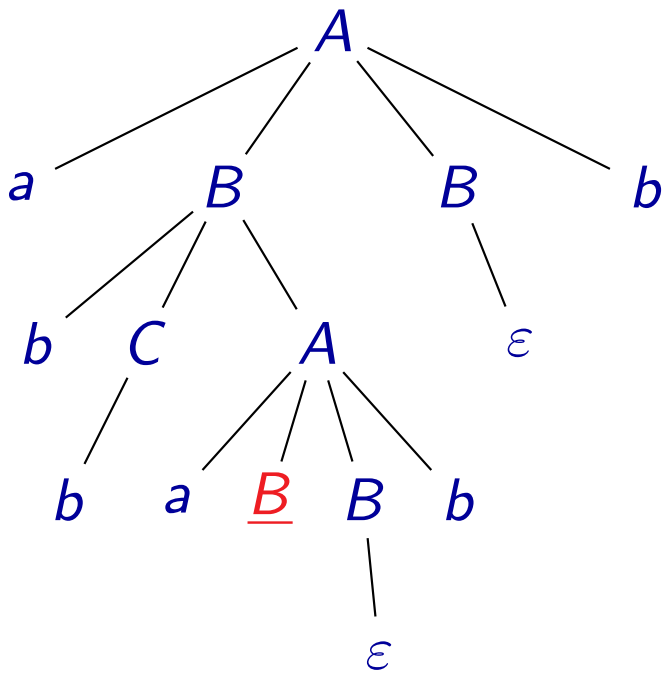
$A \to aBBb \mid AaA$
$B \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

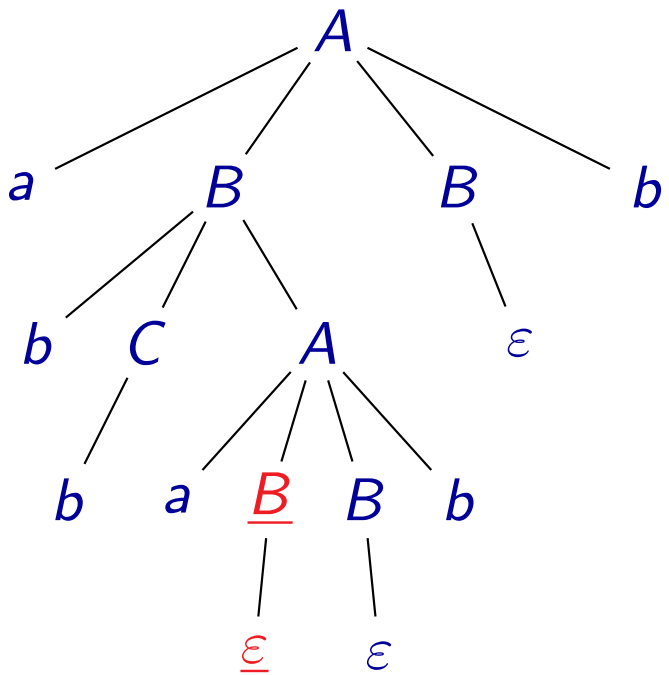
$A \to aBBb \mid AaA$
$\underline{B} \to \varepsilon \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

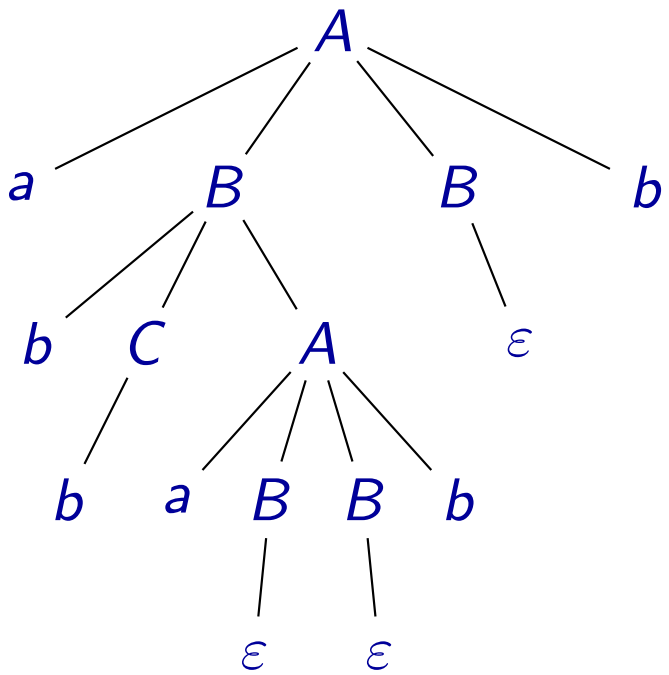
$A \to aBBb \mid AaA$
$\underline{B} \to \underline{\varepsilon} \mid bCA$
$C \to AB \mid a \mid b$



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

Derivační strom

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



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

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.

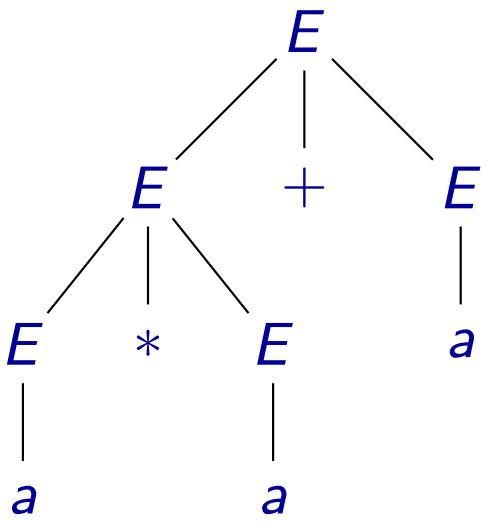
Nejednoznačné gramatiky

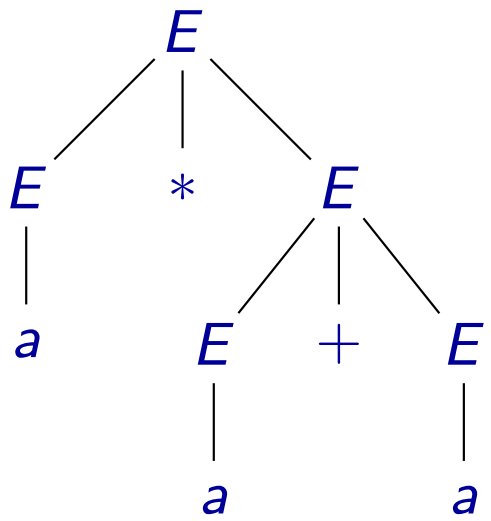
Gramatika G je **nejednoznačná**, jestliže existuje nějaké slovo $w \in L(G)$, kterému přísluší dva různé derivační stromy, resp. dvě různé levé či dvě různé pravé derivace.

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 $L(G) = L(G_1)^*$:

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

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 $X \in \Pi$, $w \in \Sigma^*$.

Příklad:

$$S \to abbA \mid bB \mid A \mid \varepsilon$$
$$A \to aA \mid babS \mid aaB$$
$$B \to bbB \mid abA \mid b$$

Regulární gramatiky generují právě třídu regulárních jazyků, tj.:

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