Rekapitulace

- Víme, co jsou to regulární jazyky.
- Umíme je vymezit pomocí konečných automatů nebo regulárních výrazů.

Regulární jazyky jsou “jednoduché”.
Postačují například pro lexikální analýzu (detekci klíčových slov, operátorů, konstant, atd.)

Na popis o něco složitějších jazyků zatím ale prostředky nemáme.

$$L = \{a^i b^i \mid i \in \mathbf{N}\}$$

Téma dnešní přednášky

Gramatiky – generativní formalismus.

- Na rozdíl od rozpoznávání slouží k postupnému vygenerování všech slov daného jazyka.

1. Zaměříme se na tzv. bezkontextové gramatiky, které generují bezkontextové jazyky.
   – vhodné pro popis syntaxe programovacích jazyků

2. Uvedeme i další důležité typy gramatik.
   – např. regulární gramatiky

Zápis syntaxe programovacího jazyka

- používá se tzv. rozšířená Backus-Naurova forma

Příklad – podmíněný příkaz (Java):

branching-statement ::= **if** *condition-clause*
single-statement | *block-statement*
[**else**
single-statement | *block-statement*]
condition-clause ::= **(** *Boolean-expression* **)**
single-statement ::= *Statement*
block-statement ::= **{** *Statement* [*Statements*] **}**

Odvození aritmetických výrazů

$$\langle EXPR\rangle \to \langle EXPR\rangle + \langle EXPR\rangle$$
$$\langle EXPR\rangle \to \langle EXPR\rangle * \langle EXPR\rangle$$
$$\langle EXPR\rangle \to (\langle EXPR\rangle)$$
$$\langle EXPR\rangle \to v$$

Zjednodušený zápis:

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

Terminální symboly: $(,),+,*,v$

Neterminální symbol (proměnná): E

Odvození aritmetických výrazů

Odvození (derivace) slova $v+v*v$:

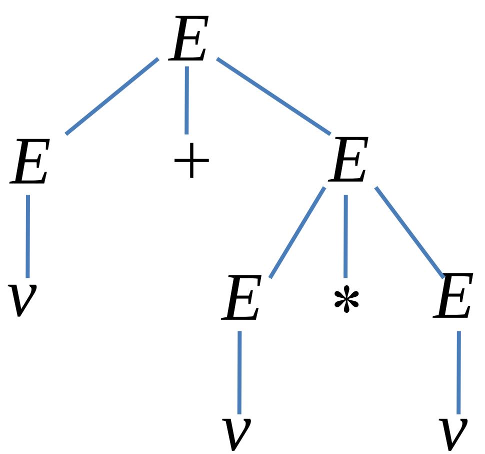
$E \Rightarrow E+E \Rightarrow v+E \Rightarrow v+E*E \Rightarrow v+v*E \Rightarrow v+v*v$

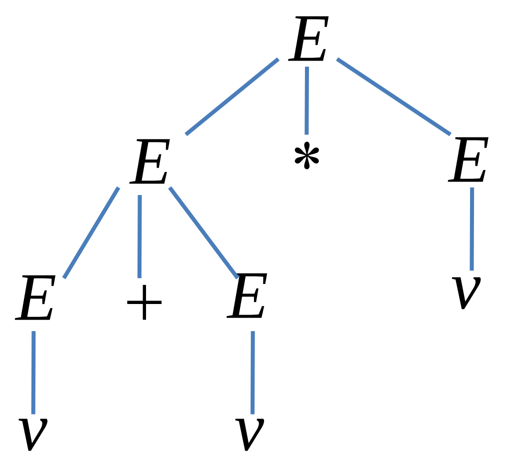
(levá derivace, nahrazujeme vždy neterminál nejvíce vlevo)

$E \Rightarrow E*E \Rightarrow E+E*E \Rightarrow v+E*E \Rightarrow v+v*E \Rightarrow v+v*v$

(též levá derivace)

Derivační strom:





Odstranění nejednoznačnosti

Pokud existují dvě různé levé derivace pro jedno slovo, říkáme, že gramatika je nejednoznačná.

$$\langle EXPR \rangle \rightarrow \langle EXPR \rangle + \langle TERM \rangle \mid \langle TERM \rangle$$
$$\langle TERM \rangle \rightarrow \langle TERM \rangle * \langle FACTOR \rangle \mid \langle FACTOR \rangle$$
$$\langle FACTOR \rangle \rightarrow (\langle EXPR \rangle) \mid v$$

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid v$$

$$E \Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow$$
$$v + T \Rightarrow v + T * F \Rightarrow v + F * F \Rightarrow$$
$$v + v * F \Rightarrow v + v * v$$

Definice bezkontextové gramatiky

Bezkontextová gramatika G je čtveřice (Π, Σ, S, P), kde

- Π je konečná množina neterminálních symbolů
- Σ je konečná množina terminálních symbolů, $\Sigma \cap \Pi = \emptyset$
- $S \in \Pi$ je počáteční neterminál
- P je konečná množina pravidel typu $A \to \alpha$, kde
  - $A \in \Pi$
  - $\alpha \in (\Pi \cup \Sigma)^*$

Poznámka: Označení „bezkontextová“ vyjadřuje, že na levé straně pravidel je pouze jeden neterminál, bez jakýkoliv sousedů, tj. není v „kontextu“.

Odvození podle gramatiky

Nechť $G = (\Pi, \Sigma, S, P)$ je bezkontextová gramatika.
Pro $\gamma, \delta \in (\Pi \cup \Sigma)^*$ řekneme, že γ přímo odvodí δ, jestliže existují slova α_1, α_2 taková, že $\gamma = \alpha_1 A \alpha_2$, $\delta = \alpha_1 \beta \alpha_2$ a $A \to \beta$ je pravidlo v P.

Píšeme:

$$\gamma \Rightarrow_G \delta \qquad\qquad \alpha_1 A \alpha_2 \Rightarrow_G \alpha_1 \beta \alpha_2$$

Řekneme, že γ odvodí δ, pokud existuje posloupnost přímých odvození:

$$\gamma = \mu_0 \Rightarrow_G \mu_1 \Rightarrow_G \cdots \Rightarrow_G \mu_n = \delta$$

Píšeme: $\gamma \Rightarrow_G^* \delta$

Jazyk generovaný gramatikou

Jazyk generovaný bezkontextovou gramatikou $G = \left(\Pi, \Sigma, S, P\right)$:

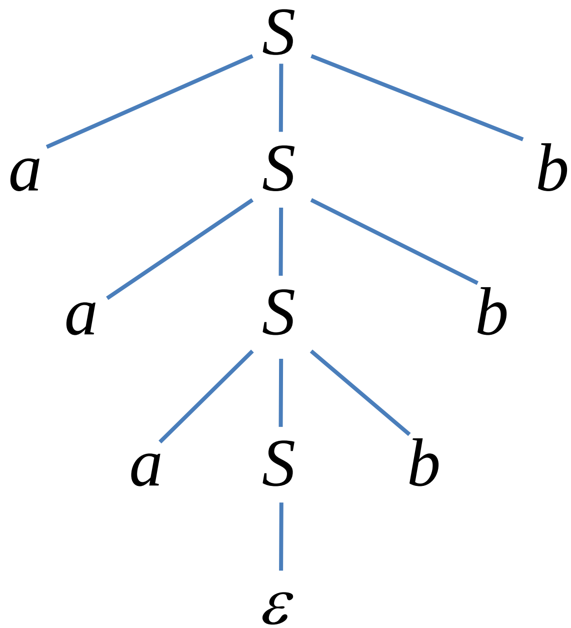
$$L(G) = \left\{ w \in \Sigma^* \mid S \Rightarrow_G^* w \right\}$$

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

Příklady bezkontextových gramatik

$$L = \left\{ a^i b^i \mid i \geq 0 \right\}$$

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



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

$$S \to \varepsilon$$
$$S \to aSa$$
$$S \to bSb$$

Jaké jazyky generují tyto gramatiky?

1. $S \rightarrow aSbS$

   $S \rightarrow \varepsilon$

2. $S \rightarrow bSbb \mid A \mid \varepsilon$

   $A \rightarrow aA \mid \varepsilon$

Redukovaná gramatika

Z technických důvodů je užitečné, že bezkontextové gramatiky lze převádět do různých normálních forem.

Redukovaná gramatika – eliminuje pravidla obsahující

1. Neterminály, ze kterých nelze odvodit žádné terminální slovo.
2. Neterminály, které nejsou dosažitelné z počátečního neterminálu.

Příklad:

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

$$A \to aAB \mid bB$$

$$B \to aAb \mid BB$$

$$C \to CC \mid cS$$

Konstrukce redukované gramatiky

$S \to aSb \mid aAbb \mid \varepsilon \qquad B \to aAb \mid BB$

$A \to aAB \mid bB \qquad C \to CC \mid cS$

Neterminály generující terminální slova:

1. $\mathbf{T}_1 = \{S\} \qquad S \to \varepsilon$
2. $\mathbf{T}_2 = \{S, C\} \qquad C \to cS$

$S \to aSb \mid \varepsilon$

$C \to CC \mid cS$

Dosažitelné neterminály:

1. $\mathbf{D}_1 = \{S\}$

$S \to aSb \mid \varepsilon$

Chomského normální forma

Chomského normální forma (CNF):

$N \rightarrow AB$ $\quad A, B, N$ … neterminály

$N \rightarrow a$ $\quad a$ … terminál

$S \rightarrow \varepsilon$ $\quad S$ … počáteční neterminál

- CNF obsahuje pouze jednoduché tvary pravidel, minimalizuje délku řetězců na pravé straně.

Věta: Každou bezkontextovou gramatiku lze převést na ekvivalentní bezkontextovou gramatiku v CNF.

Převod do CNF

Zadání: $\boxed{\begin{array}{l} S \rightarrow A \mid bSA \\ A \rightarrow aA \mid a \mid \varepsilon \end{array}}$

1. Nahradíme pravidla typu $N \rightarrow \varepsilon$, kde $N \in \Pi - \{S\}$

$$\boxed{\begin{array}{l} S \rightarrow A \mid bSA \mid \varepsilon \mid bS \\ A \rightarrow aA \mid a \end{array}}$$

2. Nahradíme pravidla typu $N \rightarrow M$, kde $N, M \in \Pi$

$$\boxed{\begin{array}{l} S \rightarrow aA \mid a \mid bSA \mid \varepsilon \mid bS \\ A \rightarrow aA \mid a \end{array}}$$

Převod do CNF

3. Pro každý terminál c zavedeme nový neterminál N_c a přidáme pravidlo $N_c \to c$. Pokud jsou na pravé straně nějakého pravidla aspoň 2 symboly, pak použijeme N_c místo c.

$$\boxed{\begin{aligned} &S \to N_a A \mid a \mid N_b SA \mid \varepsilon \mid N_b S \\ &A \to N_a A \mid a \\ &N_a \to a, \quad N_b \to b \end{aligned}}$$

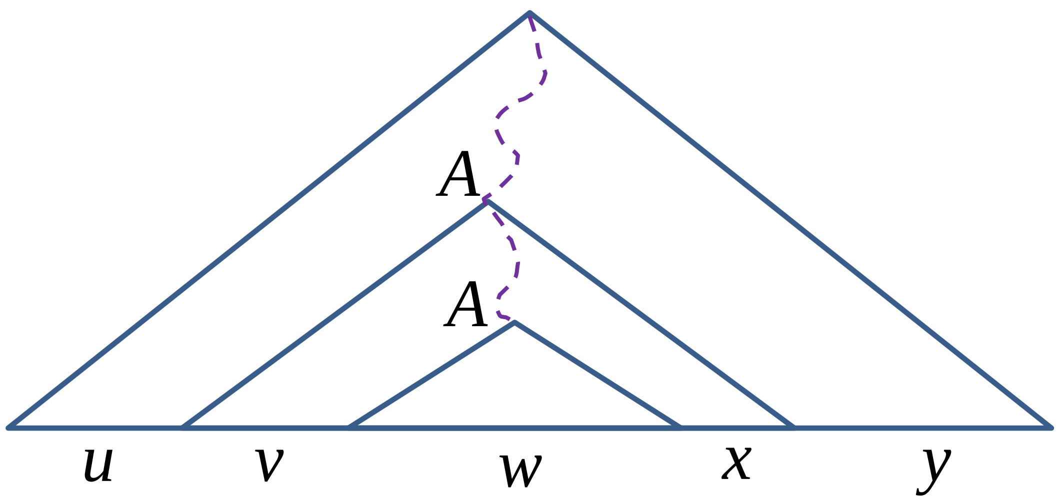
4. Upravíme řetězce neterminálů délky 3 a více.

$X \to Y_1 Y_2 \ldots Y_n$ nahradíme: $X \to Y_1 Z$ a $Z \to Y_2 \ldots Y_n$

$$\boxed{\begin{aligned} &S \to N_a A \mid a \mid N_b Z \mid \varepsilon \mid N_b S \\ &A \to N_a A \mid a \\ &N_a \to a, \quad N_b \to b, \quad Z \to SA \end{aligned}}$$

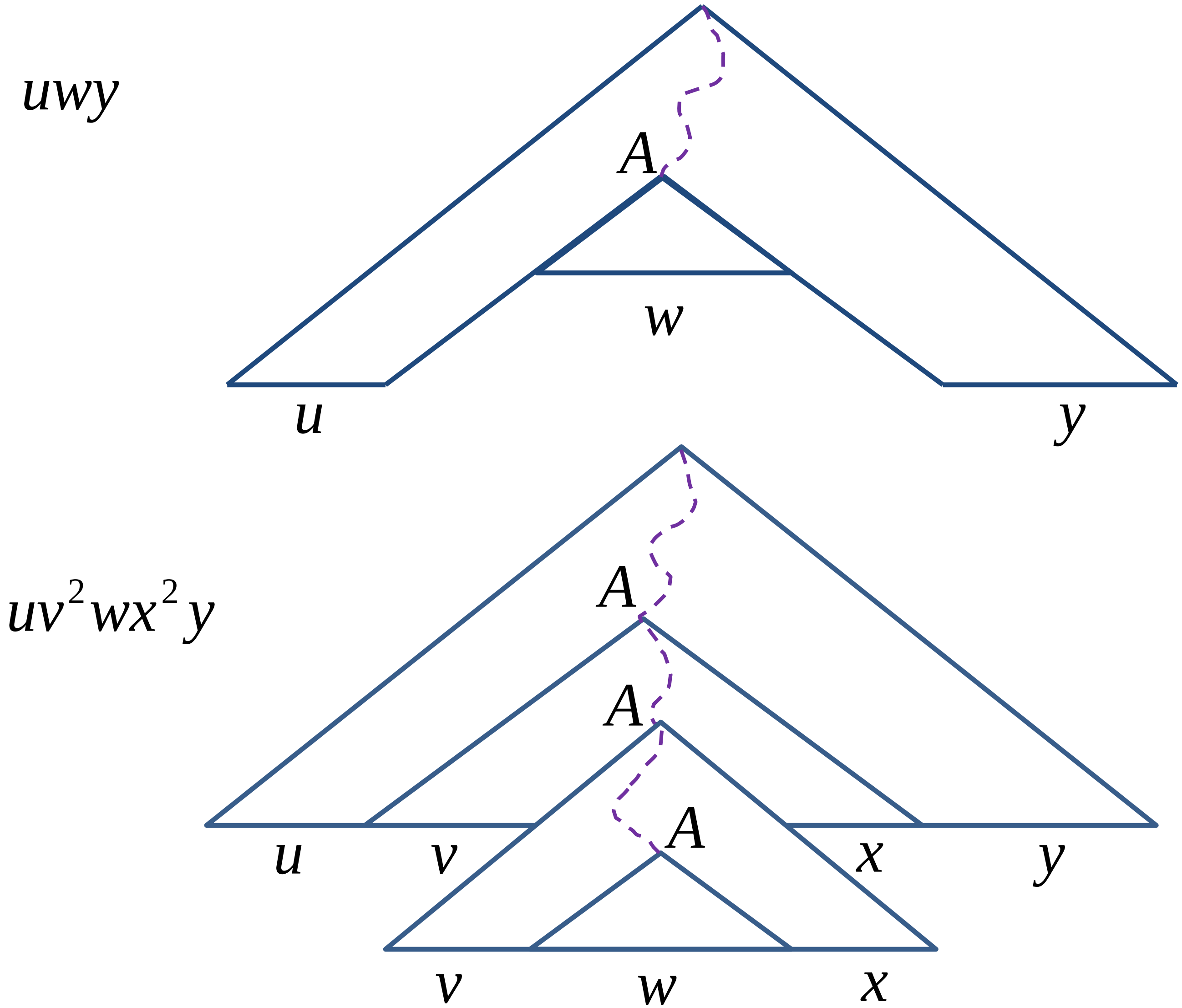
Pumping lemma – náhled k odvození

Uvažujme derivační strom popisující odvození slova z.
Pokud je $|z|$ dostatečně velké, musí nastat situace vyjádřená schématem:



Neterminálů je konečný počet, proto existuje cesta, na které se zopakuje stejný neterminál.

Pumping lemma



Pumping lemma

Věta: Nechť L je bezkontextový jazyk. Potom existuje konstanta n taková, že každé slovo $z \in L, |z| \geq n$, lze psát ve tvaru $z = uvwxy$, kde $|vx| \geq 1, |vwx| \leq n$ a pro všechna $i \in \mathbf{N}$ je $uv^i wx^i y \in L$.

Použití:

$L = \{a^i b^i c^i \mid i \in \mathbf{N}\}$ není bezkontextový jazyk

sporem, vezmeme slovo $a^n b^n c^n$

$a^n b^n c^n = uvwxy$, kde $|vwx| \leq n$

tzn. vwx nemůže obsahovat všechny tři znaky a, b, c

tzn. $uwy \notin L$ (zkrátili jsme délku některých ze tří úseků, ne ale všech)

Obecná (generativní) gramatika

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

- Zobecníme tvar pravidel:

  $\alpha \to \beta \qquad \alpha \in (\Pi \cup \Sigma)^* \Pi (\Pi \cup \Sigma)^*, \beta \in (\Pi \cup \Sigma)^*$

- Přímé odvození:

  $\gamma = \mu_1 \alpha \mu_2 \Rightarrow_G \mu_1 \beta \mu_2 = \delta$

- Opět zavedeme (obecné) odvození: $\gamma \Rightarrow_G^* \delta$

- A jazyk generovaný gramatikou:

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

Chomského hierarchie

Rozlišíme 4 důležité typy gramatik:

- Typ 0, neboli obecné gramatiky
- Typ 1, neboli kontextové gramatiky

$$\alpha X \beta \to \alpha \gamma \beta \qquad \alpha, \gamma, \beta \in (\Pi \cup \Sigma)^*, X \in \Pi, |\gamma| \geq 1$$

$$S \to \varepsilon$$

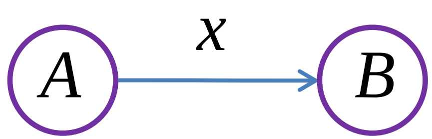
- Typ 2, neboli bezkontextové gramatiky
- Typ 3, neboli regulární gramatiky

$$X \to wY \qquad w \in \Sigma^*, \quad X, Y \in \Pi$$
$$X \to w$$

Regulární gramatiky generují regulární jazyky.

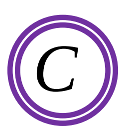
Konstrukce regulární gramatiky pro NKA

- terminály gramatiky = vstupní symboly automatu
- neterminály = stavy automatu
- počáteční neterminál = počáteční stav
- pravidla odpovídají přechodům v automatu



$$A \to xB$$

- přepis na prázdné slovo pro každý koncový stav



$$C \to \varepsilon$$