Bezkontextové jazyky

Jazyky typu 2

Definice 3.1 Gramatika $G = (N, \Sigma, P, S)$ si nazývá bezkontextovou gramatikou, jestliže všechna pravidla z P mají tvar

$$A \rightarrow \alpha, \quad A \in N, \quad \alpha \in (N \cup \Sigma)^*$$

Lemma 3.1 Každý regulární jazyk je jazykem bezkontextovým.

- Proč studujeme bezkontextové jazyky?

Příklad 3.1 Jazyk $L = \{a^n b^n \mid n \geq 0\}$, jak víme, není jazykem regulárním, je však jazykem bezkontextovým:
$L = L(G)$ kde
$G = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow \varepsilon\}, S)$

Proč mohou BG „počítat“

- Sebevkládání pomocí pravidel $A \to \alpha A \beta$ kde $\alpha, \beta \in (N \cup \Sigma)^*$

Zkonstruujte bezkontextovou gramatiku pro jazyk $L = \{a^{3n} b^{2n} \mid n \geq 0\}$

Proč mohou BG „počítat“

- Sebevkládání pomocí pravidel $A \rightarrow \alpha A \beta$ kde $\alpha, \beta \in (N \cup \Sigma)^*$

Zkonstruujte bezkontextovou gramatiku pro jazyk $L = \{a^{3n}b^{2n} \mid n \geq 0\}$

$G = (\{S\}, \{a, b\}, \{S \rightarrow aaaSbb, S \rightarrow \varepsilon\}, S)$

Proč mohou BG „počítat“

- Sebevkládání pomocí pravidel $A \to \alpha A \beta$ kde $\alpha, \beta \in (N \cup \Sigma)^*$

Zkonstruujte bezkontextovou gramatiku pro jazyk $L = \{a^{3n}b^{2n} \mid n \geq 0\}$

$G = (\{S\}, \{a, b\}, \{S \to aaaSbb, S \to \varepsilon\}, S)$

- Jak docílit libovolné pořadí symbolů?

Zkonstruujte bezkontextovou gramatiku pro jazyk $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$

Proč mohou BG „počítat“

- Sebevkládání pomocí pravidel $A \to \alpha A \beta$ kde $\alpha, \beta \in (N \cup \Sigma)^*$

Zkonstruujte bezkontextovou gramatiku pro jazyk $L = \{a^{3n}b^{2n} \mid n \geq 0\}$

$G = (\{S\}, \{a, b\}, \{S \to aaaSbb, S \to \varepsilon\}, S)$

- Jak docílit libovolné pořadí symbolů?

Zkonstruujte bezkontextovou gramatiku pro jazyk $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$

$G = (\{S\}, \{a, b\}, \{S \to aSb, S \to bSa, S \to SS, S \to \varepsilon\}, S)$

Příklad bezkontextové gramatiky

❖ Pro účely demonstrace vysvětlovaných pojmů budeme v následujících příkladech používat následující gramatiku.

Příklad 3.2 $G = (\{S, A, B\}, \{a, b, c\}, P, S)$, kde P obsahuje pravidla

$$\begin{aligned}
&S \rightarrow AB \\
&A \rightarrow aAb \mid ab \\
&B \rightarrow bBc \mid bc
\end{aligned}$$

Gramatika G generuje bezkontextový jazyk $L(G) = \{a^m b^{m+n} c^n \mid n \geq 1, m \geq 1\}$

Derivační strom

❖ Důležitým prostředkem pro grafické vyjádření struktury věty (její derivace) je strom, který se nazývá derivačním nebo syntaktickým stromem.

Definice 3.2 Nechť δ je věta nebo větná forma generovaná v gramatice $G = (N, \Sigma, P, S)$ a nechť $S = v_0 \Rightarrow v_1 \Rightarrow \ldots \Rightarrow v_k = \delta$ její derivace v G. Derivační strom příslušející této derivaci je vrcholově ohodnocený strom s těmito vlastnostmi:

1. Vrcholy derivačního stromu jsou ohodnoceny symboly z množiny $N \cup \Sigma \cup \{\varepsilon\}$; kořen stromu je označen výchozím symbolem S.
2. Přímé derivaci $v_{i-1} \Rightarrow v_i, i = 1, 2, \ldots, k$ kde
   - $v_{i-1} = \mu A \lambda,\ \mu, \lambda \in (N \cup \Sigma)^*,\ A \in N$
   - $v_i = \mu \alpha \lambda$
   - $A \rightarrow \alpha,\ \alpha = X_1 \ldots X_n$ je pravidlo z P,

   odpovídá právě n hran $(A, X_j), j = 1, \ldots, n$ vycházejících z uzlu A, jež jsou uspořádány zleva doprava v pořadí $(A, X_1), (A, X_2), \ldots, (A, X_n)$.
3. Ohodnocení koncových uzlů derivačního stromu vytváří zleva doprava větnou formu nebo větu δ (plyne z 1. a 2.).

Příklad derivačního stromu

Příklad 3.3 V gramatice z příkladu 4.2 můžeme generovat řetězec $aabbbbcc$ např. derivací:

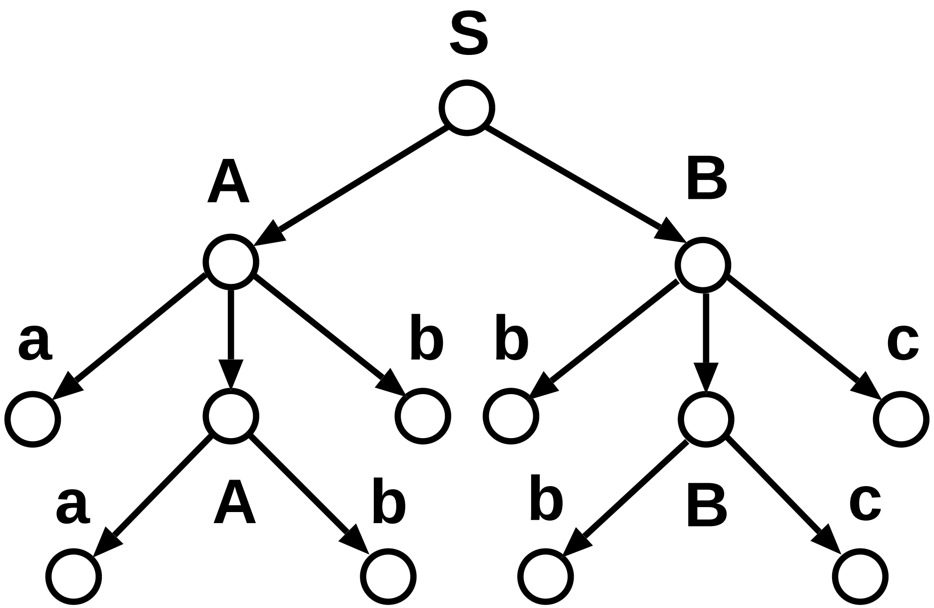
$$S \Rightarrow AB \Rightarrow aAbB \Rightarrow aAbbBc \Rightarrow aAbbbcc \Rightarrow aabbbbcc$$

Derivační strom odpovídající této derivaci vypadá takto (po stranách jsou uvedena použitá pravidla):

$S \rightarrow AB$

$A \rightarrow aAb$

$A \rightarrow ab$



$B \rightarrow bBc$

$B \rightarrow bc$

Levá a pravá derivace

❖ Ukažme si i jiné derivace věty $aabbbbcc$, které se liší v pořadí, v němž byly vybírány nonterminály pro přímé derivace.

1. $S \Rightarrow AB \Rightarrow aAbB \Rightarrow aabbB \Rightarrow aabbbBc \Rightarrow aabbbbcc$
2. $S \Rightarrow AB \Rightarrow AbBc \Rightarrow Abbcc \Rightarrow aAbbbc \Rightarrow aabbbbcc$

Definice 3.3 Nechť $S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_n = \alpha$ je derivace větné formy α. Jestliže byl v každém řetězci $\alpha_i, i = 1, \ldots, n-1$ přepsán nejlevější (nejpravější) nonterminál, pak tuto derivaci nazýváme levou (pravou) derivací větné formy α.

Výše uvedené příklady derivací představují levou (1.) a pravou (2.) derivaci.

Lemma 3.2 Je-li $S \equiv \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n \equiv w$ levá, resp. pravá derivace věty w, pak každá z větných forem α_i, $i = 1, 2, \ldots, n-1$ má tvar:

$x_i A_i \beta_i$ kde $x_i \in \Sigma^*, A_i \in N, \beta_i \in (N \cup \Sigma)^*$

resp. $\gamma_i B_i y_i$ kde $y_i \in \Sigma^*, B_i \in N, \gamma_i \in (N \cup \Sigma)^*$

t.j. větné formy levé, resp. pravé derivace mají terminální prefixy, resp. sufixy.

Víceznačnost gramatik

Definice 3.4 Nechť G je gramatika. Říkáme, že věta w generovaná gramatikou G je víceznačná, existují-li alespoň dva různé derivační stromy s koncovými uzly tvořícími větu w. Gramatika G je víceznačná, pokud generuje alespoň jednu víceznačnou větu. V opačném případě mluvíme o jednoznačné gramatice.
Jazyky, které lze generovat víceznačnou gramatikou, ale které nelze generovat jednoznačnou gramatikou, se nazývají jazyky s inherentní víceznačností.

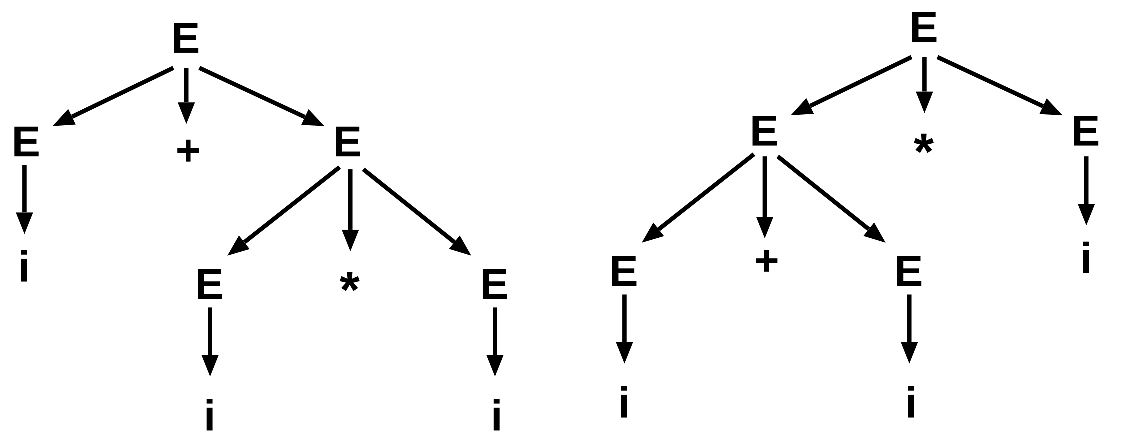
- Problém víceznačnosti gramatik je nerozhodnutelný, tj. neexistuje algoritmus, který by byl schopen v konečném čase rozhodnout, zda daná gramatika je nebo není víceznačná.
- Víceznačnost gramatiky je pokládána za negativní rys (vede k větám, které mají několik interpretací). Na druhé straně může být víceznačná gramatika jednodušší než odpovídající jednoznačná gramatika.

Víceznačnost gramatik

Příklad 3.4 Uvažujme gramatiku $G = (\{E\}, \{+, -, *, /, (,)\}, P, E)$, kde P je množina pravidel

$$E \rightarrow E + E \mid E - E \mid E * E \mid E/E \mid (\ E\ ) \mid i$$

Jazyk $L(G)$ je tvořen aritmetickými výrazy s binárními operacemi. Gramatika G je na rozdíl od gramatiky z příkladu 4.2 víceznačná. Vezměme například větu $i + i * i$ a uvažujme všechny možné derivační stromy.



Není jasné, zda první operací bude násobení (derivační strom vlevo), nebo sčítání (derivační strom vpravo).

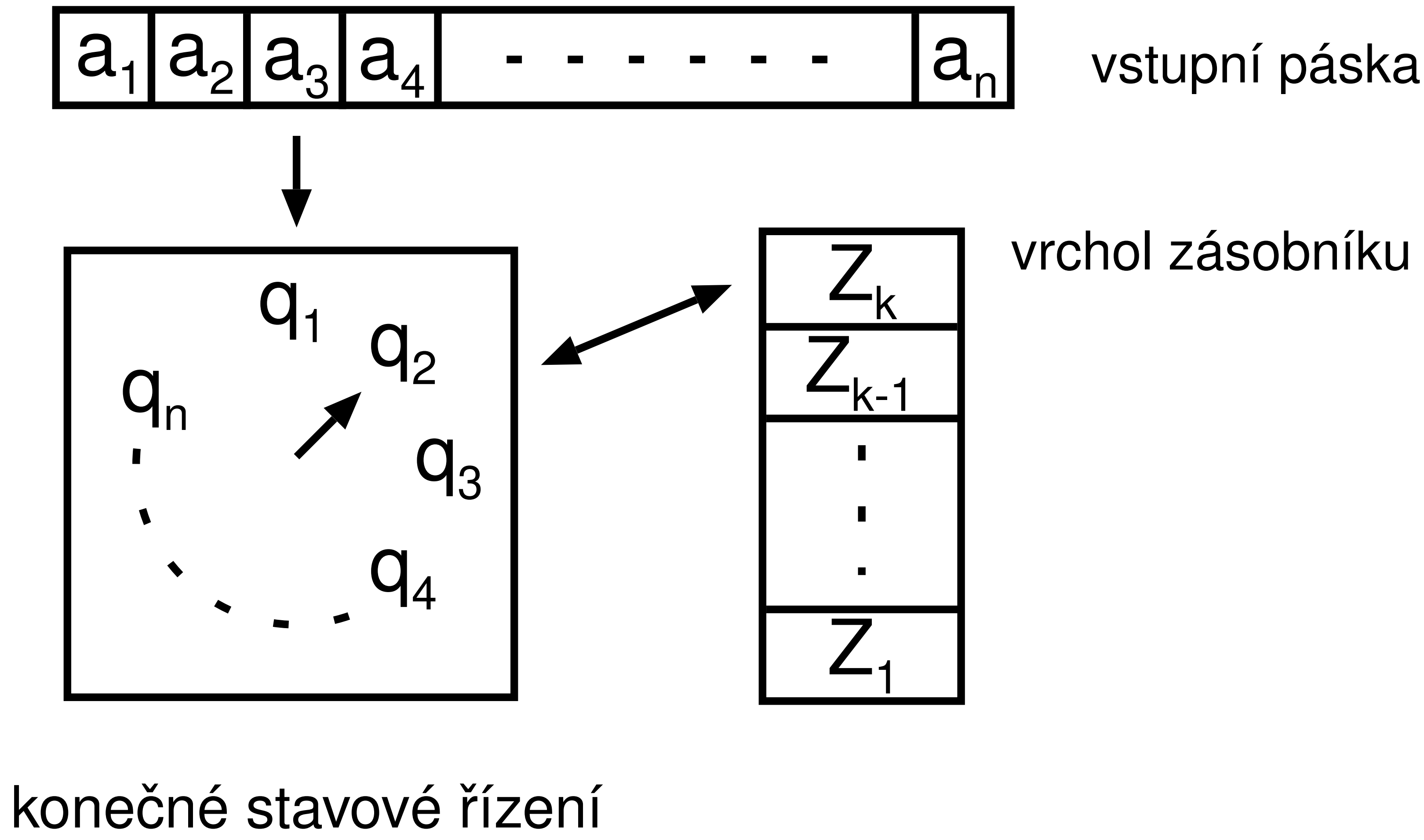
Příklad 3.5 Jednoznačnou gramatikou generující tentýž jazyk je gramatika $G = (\{E, T, F\}, \{+, -, *, /, (,), i\}, P, E)$ s množinou přepisovacích pravidel P definovanou následujícím způsobem:

$E \to T \mid E + T \mid E - T$
$T \to F \mid T * F \mid T/F$
$F \to (\ E\ ) \mid i$

Zásobníkové automaty

Základní schéma

Schéma zásobníkového automatu:



Základní definice

Definice 3.5 Zásobníkový automat P je n-tice $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

1. Q je konečná množina vnitřních stavů
2. Σ je konečná vstupní abeceda
3. Γ je konečná zásobníková abeceda
4. δ je přechodová funkce ve tvaru $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow Fin\left(2^{Q \times \Gamma^*}\right)$, kde $Fin\left(2^{Q \times \Gamma^*}\right)$ značí konečné množiny podmnožin množiny $Q \times \Gamma^*$
5. $q_0 \in Q$ je počáteční stav
6. $Z_0 \in \Gamma$ je startovací symbol zásobníku
7. $F \subseteq Q$ je množina koncových stavů

Konfigurace a přechod ZA

Definice 3.6 Nechť $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je zásobníkový automat. Konfigurací automatu P nazveme trojici $(q, w, \alpha) \in Q \times \Sigma^* \times \Gamma^*$, kde

1. q je přítomný stav vnitřního řízení
2. w je dosud nezpracovaná část vstupního řetězce
3. α je obsah zásobníku ($\alpha = Z_{i_1} Z_{i_2} \ldots Z_{i_k}$, Z_{i_1} je vrchol)

Přechod ZA P je binární relace $\vdash_P$ definovaná na množině konfigurací:

$$(q, w, \beta) \vdash_P (q', w', \beta') \overset{def}{\iff} w = aw' \wedge \beta = Z\alpha \wedge \beta' = \gamma\alpha \wedge (q', \gamma) \in \delta(q, a, Z),$$

kde $q, q' \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $w, w' \in \Sigma^*$, $Z \in \Gamma$ a $\alpha, \beta, \beta', \gamma \in \Gamma^*$.

- Je-li $a = \varepsilon$, pak odpovídající přechod nazýváme ε-přechodem.
- Relace $\vdash^i_P$, $\vdash^*_P$, $\vdash^+_P$ jsou definovány obvyklým způsobem.
- Platí-li pro řetězec $w \in \Sigma^*$ relace $(q_0, w, Z_0) \vdash^*_P (q, \varepsilon, \gamma)$, kde $q \in F$ a $\gamma \in \Gamma^*$, pak říkáme, že w je přijímán zásobníkovým automatem P (q_0, w, Z_0), resp. (q, ε, γ) je počáteční, resp. koncová konfigurace.
- Definujeme jazyk přijímaný zásobníkovým automatem P:
  $L(P) = \{w | (q_0, w, Z_0) \vdash^*_P (q, \varepsilon, \gamma) \wedge q \in F\}$.

Příklad zásobníkového automatu

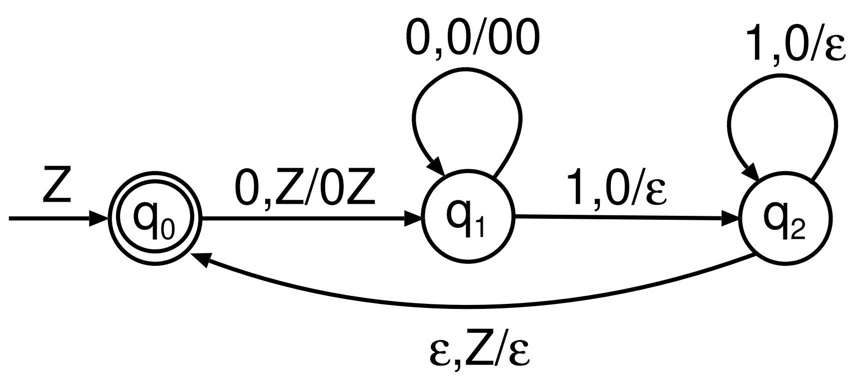
Příklad 3.6 Sestrojme zásobníkový automat, který přijímá jazyk $L = \{0^n 1^n \mid n \geq 0\}$.
– Řešením je $P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{Z, 0\}, \delta, q_0, Z, \{q_0\})$, kde

$$\delta(q_0, 0, Z) = \{(q_1, 0Z)\}$$
$$\delta(q_1, 0, 0) = \{(q_1, 00)\}$$
$$\delta(q_1, 1, 0) = \{(q_2, \varepsilon)\}$$
$$\delta(q_2, 1, 0) = \{(q_2, \varepsilon)\}$$
$$\delta(q_2, \varepsilon, Z) = \{(q_0, \varepsilon)\}$$

– Při přijetí řetězce 0011 projde P těmito konfiguracemi:

$$(q_0, 0011, Z) \vdash (q_1, 011, 0Z) \vdash (q_1, 11, 00Z) \vdash (q_2, 1, 0Z) \vdash (q_2, \varepsilon, Z) \vdash (q_0, \varepsilon, \varepsilon)$$

– Zásobníkové automaty lze také popsat přechodovým diagramem, jak je ilustrováno níže na právě sestrojeném automatu P:



Návrh složitějších automatů

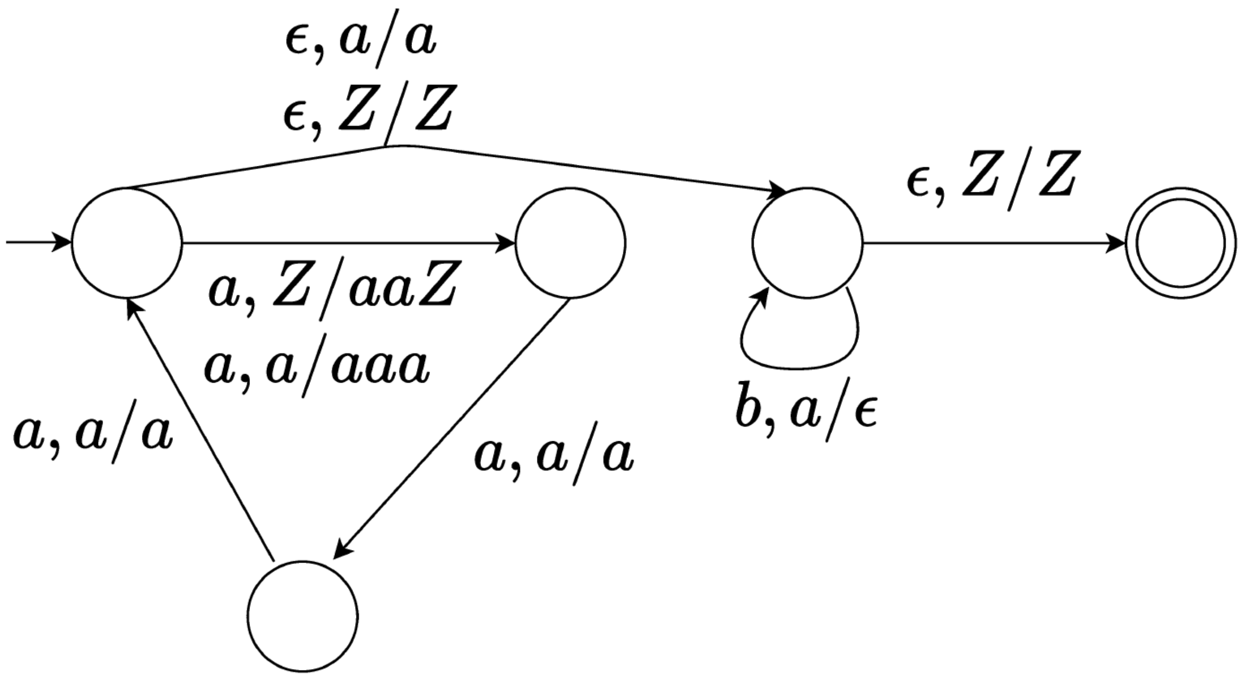
- Pokročilejší práce se zásobníkem.

Zkonstruujte zásobníkový automat pro jazyk $L = \{a^{3n}b^{2n} \mid n \geq 0\}$

Návrh složitějších automatů

❖ Pokročilejší práce se zásobníkem.

Zkonstruujte zásobníkový automat pro jazyk $L = \{a^{3n}b^{2n} \mid n \geq 0\}$



Varianty zásobníkových automatů

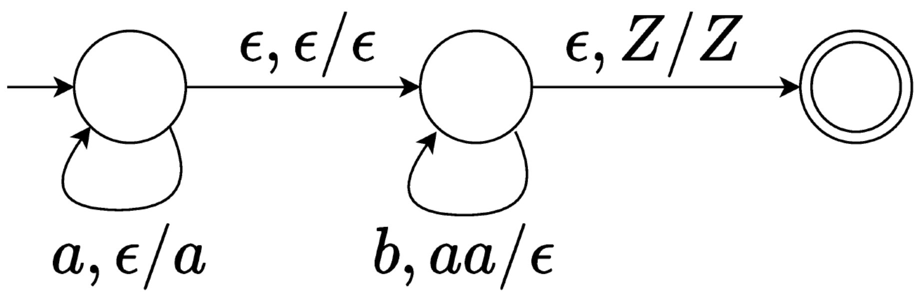
Rozšířený zásobníkový automat

Definice 3.7 Rozšířený zásobníkový automat P je sedmice $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde δ je přechodová funkce definovaná takto:

$$\delta : Fin\,(Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*) \to Fin\left(2^{Q\times\Gamma^*}\right), \text{ kde}$$

$Fin\,(Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*)$ značí konečné podmnožiny množiny $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*$. Ostatní složky mají stejný význam jako v definici 3.5.

Příklad 3.7 Zkonstruujte RZA pro jazyk $L = \{a^{2n}b^n \mid n \geq 0\}$



Ekvivalence RZA a ZA

Věta 3.1 Nechť $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je rozšířený zásobníkový automat. Pak existuje zásobníkový automat P_1 takový, že $L(P_1) = L(P)$.

Důkaz. Položme $m = max\{|\alpha| \mid \delta(q, a, \alpha) \neq \emptyset$ pro nějaké $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ a $\alpha \in \Gamma^*\}$.

Zásobníkový automat P_1 budeme konstruovat tak, aby simuloval automat P.

Protože automat P neurčuje přechody podle vrcholu zásobníku, ale podle vrcholového řetězce zásobníku, bude automat P_1 ukládat m vrcholových symbolů v jakési vyrovnávací paměti řídící jednotky tak, aby na počátku každého přechodu věděl, jakých m vrcholových symbolů je v zásobníku automatu P.

Nahrazuje-li automat P k vrcholových symbolů řetězcem délky l, pak se totéž provede ve vyrovnávací paměti automatu P_1.

Jestliže $l < k$, pak P_1 realizuje $k - l$ ε-přechodů, které přesouvají $k - l$ symbolů z vrcholu zásobníku do vyrovnávací paměti. Automat P_1 pak může simulovat další přechod automatu P.

Je-li $l \geq k$ pak se symboly přesouvají z vyrovnávací paměti do zásobníku.

Formálně můžeme konstrukci zásobníkového automatu P_1 popsat takto:
$P_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, Z_1, F_1)$, kde

1. $Q_1 = \{[q, \alpha] | q \in Q, \alpha \in \Gamma_1^* \wedge 0 \leq |\alpha| \leq m\}$
2. $\Gamma_1 = \Gamma \cup \{Z_1\}$
3. Zobrazení δ_1 je definováno takto:
   (a) Předpokládejme, že $\delta(q, a, X_1 \dots X_k)$ obsahuje $(r, Y_1 \dots Y_l)$.
      i. Jestliže $l \geq k$, pak pro všechna $Z \in \Gamma_1$ a $\alpha \in \Gamma_1^*$ taková, že $|\alpha| = m - k$, pak $\delta_1([q, X_1 \dots X_k\alpha], a, Z)$ obsahuje $([r, \beta], \gamma Z)$, kde $\beta\gamma = Y_1 \dots Y_l\alpha$ a $|\beta| = m$.
      ii. Je-li $l < k$, pak pro všechna $Z \in \Gamma_1$ a $\alpha \in \Gamma_1^*$ taková, že $|\alpha| = m - k$, pak $\delta_1([q, X_1 \dots X_k\alpha], a, Z)$ obsahuje $([r, Y_1 \dots Y_l\alpha Z], \varepsilon)$.

   (b) Pro všechna $q \in Q, Z \in \Gamma_1$ a $\alpha \in \Gamma_1^*$ taková, že $|\alpha| < m$, platí $\delta_1([q, \alpha], \varepsilon, Z) = \{([q, \alpha Z], \varepsilon)\}$. Tato pravidla vedou k naplnění vyrovnávací paměti.

4. $q_1 = [q_0, Z_0, Z_1^{m-1}]$. Vyrovnávací paměť obsahuje na počátku symbol Z_0 na vrcholu a $m-1$ symbolů Z_1 na dalších místech. Symboly Z_1 jsou speciální znaky pro označení dna zásobníku.
5. $F_1 = \{[q, \alpha] \mid q \in F, \alpha \in \Gamma_1^*\}$
   Lze ukázat, že $(a, aw, X_1 \dots X_k X_{k+1} \dots X_n) \vdash_P (r, w, Y_1 \dots Y_l X_{k+1} \dots X_n)$ platí, právě když $([q, \alpha], aw, \beta) \vdash_{P_1}^+ ([r, \alpha'], w, \beta')$ kde
   $\alpha\beta = X_1 \dots X_n Z_1^m$
   $\alpha'\beta' = Y_1 \dots Y_l X_{k+1} \dots X_n Z_1^m$
   $|\alpha| = |\alpha'| = m$
   a mezi těmito dvěma konfiguracemi automatu P_1 není žádná konfigurace, ve které by druhý člen stavu (vyrovnávací paměť) měl délku m.

Tedy relace $(q_0, w, Z_0) \vdash_P (q, \varepsilon, \alpha)$ pro $q \in F, \alpha \in \Gamma^*$ platí, právě když $([q_0, Z_0, Z_1^{m-1}], w, Z_1) \vdash_{P_1}^* ([q, \beta], \varepsilon, \gamma)$, kde $|\beta| = m$ a $\beta\gamma = \alpha Z_1^m$. Tedy $L(P) = L(P_1)$. $\square$

ZA přijímající s vyprázdněním zás.

Definice 3.8 Zásobníkový automat nebo rozšířený zásobníkový automat $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, \emptyset)$ přijímá s vyprázdněním zásobníku, pokud

$$L(P) = \{w \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon), q \in Q\}$$

Věta 3.2 Ke každému ZA (resp. RZA) P existuje ZA (resp. RZA) P', který přijímá s vyprázdněním zásobníku, takový, že $L(P) = L(P')$.

Důkaz. (Hlavní myšlenka) Opět budeme konstruovat automat P' tak, aby simuloval automat P. Kdykoli automat P dospěje do koncového stavu, přejde automat P' do speciálního stavu q_ε, který způsobí vyprázdnění zásobníku. Musíme však uvážit situaci, kdy automat P je v konfiguraci s prázdným zásobníkem, nikoli však v koncovém stavu. Abychom zabránili případům, že automat P' přijímá řetězec, který nemá být přijat, přidáme k zásobníkové abecedě automatu P' znak, jenž bude označovat dno zásobníku a může být vybrán pouze tehdy, je-li automat P' ve stavu q_ε. $\square$

Ekvivalence BJ a jazyků přijímaných ZA

Označme třídu všech jazyků přijímaných zásobníkovými automaty symbolem $\mathcal{L}_P$. Dokážeme, že $\mathcal{L}_2 = \mathcal{L}_P$ postupem analogickým s důkazem tvrzení $\mathcal{L}_3 = \mathcal{L}_M$. Ukážeme tedy, že

- ke každé bezkontextové gramatice existuje ekvivalentní zásobníkový automat, tj. $\mathcal{L}_2 \subseteq \mathcal{L}_P$ – využijeme konstrukci pro nedeterministickou syntaktickou analýzu
- a ke každému zásobníkovému automatu existuje ekvivalentní gramatika typu 2, tj. $\mathcal{L}_P \subseteq \mathcal{L}_2$

Poznámka: Pokročilé konstrukce pro syntaktickou analýzu BG se již v TINu neučí a nezkouší (více informací viz opora a kurzy IFJ a VYPa).

Nedeterministická analýza shora dolů

Věta 3.3 Nechť $G = (N, \Sigma, P, S)$ je bezkontextová gramatika. Pak existuje zásobníkový automat P, který přijímá s vyprázdněním zásobníku takový, že $L(G) = L(P)$.

Důkaz. Zásobníkový automat P vytvoříme tak, aby vytvářel levou derivaci vstupního řetězce v gramatice G (modeloval syntaktickou analýzu shora dolů). Nechť P je ZA:

$P = (\{q\}, \Sigma, N \cup \Sigma, \delta, q, S, \emptyset)$, kde δ je určena takto:

- Je-li $A \to \alpha$ pravidlo z P, pak $(q, \alpha) \in \delta(q, \varepsilon, A)$
- $\delta(q, a, a) = \{(q, \varepsilon)\}$ pro všechna $a \in \Sigma$

Indukcí lze dokázat ekvivalenci

$$A \Rightarrow^m w \;\Leftrightarrow\; (q, w, A) \vdash^n (q, \varepsilon, \varepsilon), m, n \geq 1, w \in \Sigma^*$$

což pro případ $A = S$ znamená $L(G) = L(P)$.

$\square$

Příklad 3.8 Ke gramatice

$$G = (\{S\}, \{0, 1\}, \{S \to 0S1, S \to 01\}, S),$$

sestrojíme zásobníkový automat P, který modeluje syntaktickou analýzu shora dolů:

$$P = (\{q\}, \{0, 1\}, \{S, 0, 1\}, \delta, q, S, 0), \text{ kde}$$

$$\begin{aligned}
&\delta(q, \varepsilon, S) = \{(q, 0S1), (q, 01)\}\\
&\delta(q, 0, 0) = \{(q, \varepsilon)\}\\
&\delta(q, 1, 1) = \{(q, \varepsilon)\}
\end{aligned}$$

Skutečně, např. derivaci

$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000111$$

odpovídá posloupnost přechodů automatu P:

$(q, 000111, S) \vdash (q, 000111, 0S1) \vdash (q, 00111, S1) \vdash (q, 00111, 0S11) \vdash (q, 0111, S11) \vdash$
$(q, 0111, 0111) \vdash (q, 111, 111) \vdash (q, 11, 11) \vdash (q, 1, 1) \vdash (q, \varepsilon, \varepsilon)$

$*\mathcal{L}_P \subseteq \mathcal{L}_2*$

Věta 3.4 Nechť $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$ je zásobníkový automat přijímající s vyprázdněním zásobníku. Pak existuje gramatika $G = (N, \Sigma, P, S)$ taková, že

$$L(P) = L(G).$$

Důkaz. Gramatiku G budeme definovat formálně takto:

- $N = \{[qZr] \mid q, r \in Q, Z \in \Gamma\} \cup \{S\}$
- Jestliže $(r, X_1X_2 \ldots X_k) \in \delta(q, a, Z)$, $k \geq 1$, pak k P přidej pravidla tvaru
$$[qZs_k] \rightarrow a[rX_1s_1][s_1X_2s_2] \ldots [s_{k-1}X_ks_k]$$
pro každou posloupnost stavů $s_1, s_2, \ldots, s_k$ z množiny Q
- Jestliže $(r, \varepsilon) \in \delta(q, a, Z)$, pak k P přidej pravidlo $[qZr] \rightarrow a$ (pro $a \in \Sigma \cup \{\varepsilon\}$)
- Pro každý stav $q \in Q$ přidej k P pravidlo $S \rightarrow [q_0Z_0q]$

Indukcí lze dokázat $S \Rightarrow [q_0Z_0q] \Rightarrow^+ w$ právě když $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$

□

Ukázka fixpoint algoritmů pro BG

- Fixpoint algoritmy se využívají pro různé transformace BG (viz opora)
  - jednotlivé transformace se již neučí a nezkouší
  - na zkouškách se ale může objevit konstrukce nějakých fixpoint algoritmů (viz cvičení)

Algoritmus 3.1 Výpočet množiny nonterminálů generujících terminální řetězce
Vstup: Gramatika $G = (N, \Sigma, P, S)$.
Výstup: Množina $N_t = \{A \in N \mid A \Rightarrow^+ w, w \in \Sigma^*\}$.
Metoda: Počítáme množiny $N_0, N_1, N_2, \ldots$ rekurentně takto:

1. N_0 := $\emptyset, i = 1$
2. N_i := $\{A \mid A \rightarrow \alpha$ je v P a $\alpha \in (N_{i-1} \cup \Sigma)^*\}$
3. Je-li $N_i \neq N_{i-1}$, i := $i + 1$ a vrať se k (2). Je-li $N_i = N_{i-1}$, polož $N_t = N_i$ a skonči.

Příklad 3.9 Uvažujme gramatiku
$G = (\{S, A, B\}, \{a, b\}, \{S \rightarrow a, S \rightarrow A, A \rightarrow AB, B \rightarrow b\}, S)$.

1. $N_0 = \emptyset$
2. $N_1 = \{S, B\}$
3. $N_2 = N_1 = N_t = \{S, B\}$

Ukázka fixpoint algoritmů pro BG

Algoritmus 3.2 Výpočet množiny dostupných symbolů
Vstup: Gramatika $G = (N, \Sigma, P, S)$.
Výstup: Množina $V = \{X \in N \cup \Sigma \mid S \Rightarrow^* \alpha X \beta, \alpha, \beta \in (N \cup \Sigma)^*\}$.
Metoda:

1. V_0 := $\{S\}, i = 1$
2. V_i := $\{X \mid A \rightarrow \alpha X \beta$ je v P a $A \in V_{i-1}\} \cup V_{i-1}$
3. Je-li $V_i \neq V_{i-1}$, i := $i + 1$ a vrať se k (2). Je-li $V_i = V_{i-1}$, polož $V = V_i$ a skonči.

Příklad 3.10 Uvažujme gramatiku
$G = (\{S, A, B\}, \{a, b\}, \{S \rightarrow a, S \rightarrow A, A \rightarrow AB, B \rightarrow b\}, S)$.

1. $V_0 = S$
2. $V_1 = \{S, a, A\}$
3. $V_2 = \{S, a, A, B\}$
4. $V_3 = \{S, a, A, B, b\}$
5. $V_4 = V_2 = V = \{S, a, A, B, b\}$

Chomského normální forma (CNF)

- Důležitá (ekvivalentní) varianta BG využívaná v důkazu Pumping teorému pro BJ.

Definice 3.9 Bezkontextová gramatika $G = (N, \Sigma, P, S)$ je v Chomského normální formě, má-li každé pravidlo z P jeden z těchto tvarů:

1. $A \to BC$, kde $A, B, C \in N$
2. $A \to a$, kde $a \in \Sigma$
3. je-li $\varepsilon \in L(G)$, pak $S \to \varepsilon$ je jediné ε-pravidlo a S se nevyskytuje na pravé straně žádného přepisovacího pravidla.

Problém: Nechť $G = (N, \Sigma, P, S)$ je bezkontextová gramatika v CNF a nechť $w \in L(G)$ a $S \Rightarrow_G^p w$. Jaká je délka řetězce w?

Řešení: Označme $|w| = n$. Zřejmě platí

$$p = n + (n - 1) = 2n - 1$$

$$|w| = \frac{p + 1}{2}$$

Věta 3.5 Nechť G je bezkontextová gramatika. Pak existuje gramatika G' v Chomského normální formě taková, že $L(G') = L(G)$.

Důkaz. (Hlavní myšlenka) Gramatiku G převedeme na ekvivalentní vlastní gramatiku bez jednoduchých pravidel (viz opora).

1. Pravidla tvaru (1), (2) a (3) ponecháme.
2. Pravidla tvaru $A \to X_1 X_2 \dots X_n$, kde $X_i \in (N \cup \Sigma)$ pro $i = 1, \dots, n$, $n > 2$, transformujeme na $A \to X_1' \langle X_2 X_3 \dots X_n \rangle$, kde $\langle X_2 X_3 \dots X_n \rangle$ je nový nonterminál a X_1' je nový nonterminál pokud $X_1 \in \Sigma$, nebo $X_1' = X_1$ v opačném případě.
3. Pravidla tvaru $A \to X_1 X_2$ transformujeme na pravidla $A \to X_1' X_2'$, kde X_i' je nový nonterminál pokud $X_i \in \Sigma$, nebo $X_i' = X_i$ v opačném případě pro $i \in \{1, 2\}$
4. Pro nové nonterminály tvaru $\langle X_1 X_2 \dots X_n \rangle$, $n \geq 2$, zavedeme pravidla $\langle X_1 X_2 \dots X_n \rangle \to X_1' \langle X_2 \dots X_n \rangle$ pro $n > 2$ a $\langle X_1 X_2 \rangle \to X_1' X_2'$ pro $n = 2$, kde $\langle X_2 \dots X_n \rangle$ je nový nonterminál a X_i' je nový nonterminál pokud $X_i \in \Sigma$, nebo $X_i' = X_i$ v opačném případě pro $i \in \{1, 2\}$.
5. Pro nové nonterminály tvaru X_i', kde $X_i \in \Sigma$ přidáme pravidla tvaru $X_i' \to X_i$.

$\square$

Příklad 3.11 Uvažujme gramatiku $G = (\{A, B\}, \{a, b, c\}, P, A)$ s pravidly:

$A \rightarrow BAB \mid Ba \mid bc$
$B \rightarrow AB \mid a \mid BBB$

Po aplikaci transformací (1.)-(4.) získáme CNF ve tvaru:

$A \rightarrow B\langle AB\rangle | Ba' | b'c'$
$B \rightarrow AB \mid a \mid B\langle BB\rangle$
$\langle AB\rangle \rightarrow AB$
$\langle BB\rangle \rightarrow BB$
$a' \rightarrow a$
$b' \rightarrow b$
$c' \rightarrow c$

Vlastnosti bezkontextových jazyků

Pumping teorém pro BJ

Věta 3.6 Nechť L je bezkontextový jazyk. Pak existuje konstanta $k > 0$ taková, že je-li $z \in L$ a $|z| \geq k$, pak lze z napsat ve tvaru:

$$z = uvwxy, vx \neq \varepsilon, |vwx| \leq k$$

a pro všechna $i \geq 0$ je $uv^i wx^i y \in L$.

❖ Ekvivalentní formulace Pumping lemmatu (použití explicitní alternace kvantifikátorů) :

$$\begin{aligned} L \in \mathcal{L}_2 \Rightarrow \quad & \exists k > 0 : \\ & \forall z \in \Sigma^* : z \in L \wedge \ |z| \geq k \Rightarrow \\ & (\exists uvwxy \in \Sigma^* : z = uvwxy \wedge \ vx \neq \epsilon \ \wedge \ |vwx| \leq k \wedge \forall i \geq 0 : uv^i wx^i y \in L) \end{aligned}$$

Důkaz. Nechť $L = L(G)$ a nechť $G = (N, \Sigma, P, S)$ je gramatika v CNF.

1. Nejprve dokážeme implikaci:
   Jestliže $A \Rightarrow^+ w$ pro nějaké $A \in N$, $w \in \Sigma^*$, pak $|w| \leq 2^{m-2}$, kde m je počet vrcholů nejdelší cesty v odpovídajícím derivačním stromu.

   Tato implikace platí, protože $|w|$ je rovno počtu přímých předchůdců listů příslušného derivačního stromu, který je maximálně roven počtu listů plného binárního stromu, jehož všechny větve obsahují $m - 1$ uzlů, což je právě 2^{m-2}.

   Skutečně:
   - Plný binární strom s větvemi o n uzlech, má 2^{n-1} listů, což se snadno ukáže indukcí:
     - Plný binární strom s (jedinou) větví o $n = 1$ uzlu, má $1 = 2^0 = 2^{n-1}$ listů.
     - Plný binární strom s větvemi délky $n = n' + 1$ uzlů, $n' \geq 1$, má $2^{n'-1} + 2^{n'-1} = 2.2^{n'-1} = 2^{1+n'-1} = 2^{n'} = 2^{n-1}$ listů.
   - Postačí tedy volit $n = m - 1$, přičemž případ neplných binárních stromů není třeba uvažovat, neboť se zajímáme o stromy s maximálním počtem listů při dané maximální délce větví.

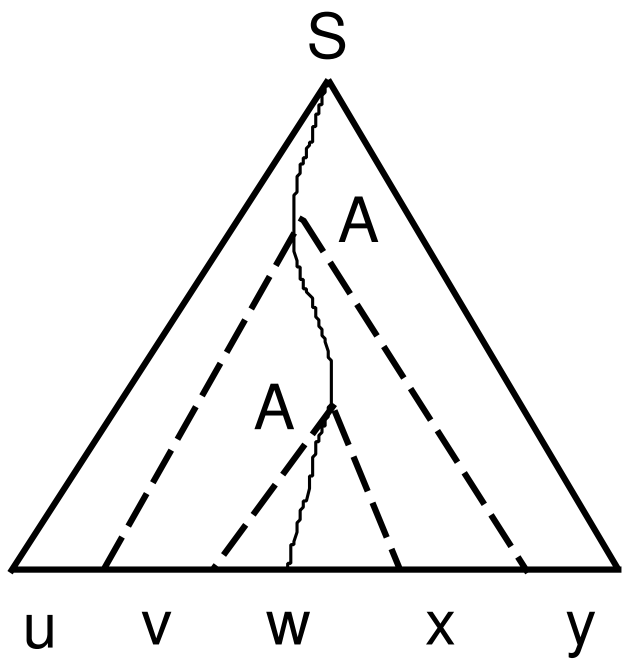
Důkaz pokračuje dále.

2. Položme $k = 2^{|N|} > 0$ a uvažujme libovolnou větu z takovou, že $|z| \geq k$.

   Označíme-li m počet vrcholů nejdelší cesty v odpovídajícím derivačním stromu, pak $2^{|N|} \leq 2^{m-2}$ a taková cesta pak obsahuje alespoň $|N| + 2$ vrcholů ($|N| + 2 \leq m$).

   Z těchto $|N| + 2$ vrcholů je jeden terminál a nutně alespoň dva jsou označeny stejným nonterminálem, řekněme A.

   Viz obrázek vpravo.



Řetězce v, x nemohou být prázdné, protože aplikované pravidlo mělo tvar $A \rightarrow BC$. Nyní uvažujme derivaci řetězce z tvaru:

$S \Rightarrow^* uAy \Rightarrow^+ uvAxy \Rightarrow^+ uvwxy = z$

To pak ovšem znamená, že v G existuje rovněž derivace:

$S \Rightarrow^* uAy \Rightarrow^+ uvAxy \Rightarrow^+ uvvAxxy \Rightarrow^+ uv^2wx^2y$,

protože $A \Rightarrow^+ w$, a tedy derivace $S \Rightarrow^* uv^iwx^iy$ pro libovolné $i > 0$, což je dokazované tvrzení. $\square$

Použití Pumping lemmatu

$$(L \in \mathcal{L}_2 \Rightarrow A) \Leftrightarrow (\neg A \Rightarrow L \notin \mathcal{L}_2) \quad \text{Obměna implikace}$$

$$\begin{aligned}
A \equiv \quad & \exists k > 0 : \\
& \forall z \in \Sigma^* : z \in L \wedge \ |z| \geq k \Rightarrow \\
& (\exists uvwxy \in \Sigma^* : z = uvwxy \wedge \ vx \neq \epsilon \ \wedge \ |vwx| \leq k \wedge \forall i \geq 0 : uv^i wx^i y \in L)
\end{aligned}$$

$$\begin{aligned}
\neg A \equiv \quad & \forall k > 0 : \\
& \exists z \in \Sigma^* : z \in L \wedge \ |z| \geq k \ \wedge \\
& (\forall u, v, w, x, y \in \Sigma^* : z = uvwxy \wedge \ vx \neq \epsilon \ \wedge \ |vwx| \leq k \Rightarrow \exists i \geq 0 : uv^i wx^i y \notin L)
\end{aligned}$$

- K důkazu, že jazyk L není bezkontextový stačí dokázat tvrzení $\neg A$.

Aplikace pumping teorému

Lemma 3.3 Jazyk $L = \{ww \mid w \in \{a, b, c\}^*\}$ není bezkontextovým jazykem.

Důkaz.

❖ Pro libovolné $k > 0$ zvolíme slovo $z = a^k b^k a^k b^k$ ($z \in L \wedge |z| \geq k$).

Poznámka: Uvažte, proč je volba slov typu $z = a^{2k}$ či $z = a^k b^{10} a^k b^{10}$ špatná (tj. důkaz pro tyto slova nelze provést).

❖ Dále uvažme všechny rozdělení $z = uvwxy$ kde $vx \neq \epsilon \;\wedge\; |vwx| \leq k$.

1. $vwx = a^m$: Při volbě $i \neq 1$ ve slově uv^iwx^iy porušíme počty znaků a v první a druhé části slova.
2. $vwx = b^m$: Podobně jako v (1) akorát porušíme počty znaků b.
3. $vwx = a^m b^n$: Při volbě $i \neq 1$ ve slově uv^iwx^iy porušíme shodu první a druhé části slova.
4. $vwx = b^m a^n$: Stejné jako v (3).

Uvědomme si, že volby $vwx = a^m b^n a^o$ a $vwx = b^m a^n b^o$ porušují podmínku $|vwx| \leq k$.

❖ Ukázali jsme, že pro L platí tvrzení $\neg A$ (viz. předchozí slajd) a tudíž $L \notin \mathcal{L}_2$.

Substituce jazyků

Definice 3.10 Nechť $\mathcal{L}$ je třída jazyků a nechť $L \subseteq \Sigma^*$ je jazykem třídy $\mathcal{L}$. Dále nechť $\Sigma = \{a_1, a_2, ..., a_n\}$ pro nějaké $n \in \mathbb{N}$ a nechť jazyky označené $L_{a_1}, L_{a_2}, ..., L_{a_n}$ jsou rovněž jazyky třídy $\mathcal{L}$. Říkáme, že třída $\mathcal{L}$ je uzavřena vzhledem k substituci, jestliže pro každý výběr jazyků $L, L_{a_1}, L_{a_2}, ..., L_{a_n}$ je také jazyk $\sigma_{L_{a_1}, L_{a_2}, ..., L_{a_n}}(L)$

$$\sigma_{L_{a_1}, L_{a_2}, ..., L_{a_n}}(L) = \{x_1 x_2 ... x_m \mid b_1 b_2 ... b_m \in L \wedge \forall i \in \{1, ..., m\} : x_i \in L_{b_i}\}$$

ve třídě $\mathcal{L}$.

Příklad 3.12 Nechť $L = \{0^n 1^n \mid n \geq 1\}$, $L_0 = \{a\}$, $L_1 = \{b^m c^m \mid m \geq 1\}$. Substitucí jazyků L_0 a L_1 do L dostaneme jazyk

$$L' = \{a^n b^{m_1} c^{m_1} b^{m_2} c^{m_2} ... b^{m_n} c^{m_n} \mid n \geq 1 \wedge \forall i \in \{1, ..., n\} : m_i \geq 1\}$$

Morfismus jazyků

Definice 3.11 Nechť Σ a Δ jsou abecedy a $L \subseteq \Sigma^*$ je jazyk nad abecedou Σ.

- Zobrazení $h : \Sigma^* \to \Delta^*$ nazveme morfismem nad slovy, platí-li $\forall w = a_1 a_2 ... a_n \in \Sigma^* : h(w) = h(a_1)h(a_2)...h(a_n)$.
- Morfismus jazyka $h(L)$ pak definujeme jako $h(L) = \{h(w) \mid w \in L\}$.

❖ Morfismus jazyků je zvláštní případ substituce, kde každý substituovaný jazyk má právě jednu větu.

Uzavřenost vůči substituci

Věta 3.7 Třída bezkontextových jazyků je uzavřena vůči substituci.

Důkaz.

- Ve shodě s definicí substituce nechť $\Sigma = \{a_1, a_2, ..., a_n\}$ je abeceda bezkontextového jazyka L a L_a pro $a \in \Sigma$ libovolné bezkontextové jazyky. Nechť $G = (N, \Sigma, P, S)$ a $G_a = (N_a, \Sigma_a, P_a, S_a)$ pro $a \in \Sigma$ jsou gramatiky, pro které $L = L(G)$ a $L_a = L(G_a)$ pro $a \in \Sigma$.
- Předpokládejme, že $N \cap N_a = \emptyset$ a $N_a \cap N_b = \emptyset$ pro každé $a, b \in \Sigma$, $a \neq b$. Sestrojme gramatiku $G' = (N', \Sigma', P', S)$ takto:
  1. $N' = N \cup \bigcup_{a \in \Sigma} N_a$.
  2. $\Sigma' = \bigcup_{a \in \Sigma} \Sigma_a$.
  3. Nechť h je morfismus na $N \cup \Sigma$ takový, že
     - $h(A) = A$ pro $A \in N$ a
     - $h(a) = S_a$ pro $a \in \Sigma$

     a nechť $P' = \{A \rightarrow h(\alpha) \mid (A \rightarrow \alpha) \in P\} \cup \bigcup_{a \in \Sigma} P_a$.
- Uvažujme libovolnou větu $a_{i_1} a_{i_2} ... a_{i_m} \in L$ a věty $x_j \in L_{a_j}$, $1 \leq j \leq m$. Pak $S \underset{G'}{\overset{*}{\Rightarrow}} S_{a_{i_1}} S_{a_{i_2}} .... S_{a_{i_m}} \underset{G'}{\overset{*}{\Rightarrow}} x_1 S_{a_{i_2}} .... S_{a_{i_m}} \underset{G'}{\overset{*}{\Rightarrow}} ... \underset{G'}{\overset{*}{\Rightarrow}} x_1 x_2 ... x_m$ a tedy $L' \subseteq L(G')$. Podobně $L(G') \subseteq L'$. $\square$

Důkaz uzavřenosti $\mathcal{L}_2$ jazyků

Nechť L_a a L_b jsou bezkontextové jazyky.

1. Uzavřenost vůči $\cup$ plyne ze substituce L_a, L_b do jazyka $\{a, b\}$.
2. Uzavřenost vůči $.$ plyne ze substituce L_a, L_b do jazyka $\{ab\}$.
3. Uzavřenost vůči $*$ plyne ze substituce L_a do jazyka $\{a\}^*$.
4. Uzavřenost vůči $+$ plyne ze substituce L_a do jazyka $\{a\}^+$.
5. Nechť h je daný morfismus a $L'_a = \{h(a)\}$ pro $a \in \Sigma$. Substitucí jazyků L'_a do jazyka L získáme jazyk $h(L)$.

Věta 3.8 Bezkontextové jazyky jsou uzavřeny vzhledem k průniku s regulárními jazyky.

Důkaz. Snadno zkonstruujeme ZA přijímající příslušný průnik – konstruujeme průnik na konečném řízení, zásobníkové operace zůstávají. □

Neuzavřenost $\mathcal{L}_2$ vůči průniku a doplňku

Věta 3.9 Bezkontextové jazyky *nejsou* uzavřeny vůči průniku a doplňku.

Důkaz.

1. Neuzavřenost vůči $\cap$:
   Uvažujme jazyky $L_1 = \{a^m b^m c^n \mid n, m \geq 1\}$ a $L_2 = \{a^m b^n c^n \mid m, n \geq 1\}$, které jsou oba bezkontextové. Ovšem $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 1\}$, což není bezkontextový jazyk (lze ukázat např. pomocí Pumping lemmatu).
2. Neuzavřenost vůči doplňku: Předpokládejme, že bezk. jazyky jsou uzavřeny vůči doplňku. Z De Morganových zákonů (a z uzavřenosti vůči sjednocení) pak ovšem plyne uzavřenost vůči průniku $L_1 \cap L_2 = \overline{\overline{L_1 \cap L_2}} = \overline{\overline{L_1} \cup \overline{L_2}}$, což je spor.

$\square$

Rozhodnutelné problémy pro $\mathcal{L}_2$

Věta 3.10 Následující problémy jsou rozhodnutelné, tj. jsou algoritmicky řešitelné:

1. problém neprázdnosti jazyka $L(G)$ pro libovolnou bezkontextovou gramatiku G,
2. problém příslušnosti řetězce $w \in \Sigma^*$ do jazyka $L(G)$ pro libovolnou bezkontextovou gramatiku G,
3. problém konečnosti jazyka $L(G)$ pro libovolnou bezkontextovou gramatiku G.

Důkaz.

1. K rozhodování neprázdnosti lze využít algoritmus iterativně určující množinu N_t nonterminálů generujících terminální řetězce uvedený v předchozí přednášce. Pak $L(G) \neq \emptyset \Leftrightarrow S \in N_t$.
2. U problému příslušnosti řetězce můžeme např. určit průnik NZA s KA přijímajícím právě řetězec w a pak ověřit neprázdnost.

Důkaz pokračuje dále.

Pokračování důkazu.

3. Problém konečnosti můžeme rozhodovat na základě platnosti Pumping lemma pro CFL:
   - Dle Pumping lemma pro bezkontextové jazyky existuje pro každý bezkontextový jazyk L konstanta $k \in \mathbb{N}$ taková, že každou větu $w \in L$, $|w| \geq k$, můžeme rozepsat jako $uvwxy$, kde $vx \neq \varepsilon$ a $|vwx| \leq k$, a $\forall i \in \mathbb{N} : uv^iwx^iy \in L$.
   - Pro testování konečnosti tedy postačí ověřit, že žádný řetězec ze Σ^* o délce mezi k a $2k - 1$ nepatří do daného jazyka:
     - Pokud takový řetězec existuje, může být „napumpován“ a dostáváme nekonečně mnoho řetězců patřících do daného jazyka.
     - Jestliže takový řetězec neexistuje, $k - 1$ je horní limit délky řetězců L.
     - Pokud by existoval řetězec délky $2k$ nebo větší patřící do L, můžeme v něm podle Pumping lemma najít vwx a vypustit vx. Vzhledem k tomu, že $0 < |vx| \leq k$, postupným opakováním vypouštění bychom se dostali k nutné existenci řetězce z L o délce mezi k a $2k - 1$.
   - K určení konstanty k postačí reprezentovat L pomocí bezkontextové gramatiky v CNF s n nonterminály a zvolit $k = 2^n$ (viz důkaz Pumping lemma).

$\square$

Nerozhodnutelné problémy pro $\mathcal{L}_2$

Věta 3.11 Následující problémy jsou nerozhodnutelné, tj. nejsou algoritmicky řešitelné:

1. problém ekvivalence jazyků bezkontextových gramatik, tj. otázka, zda $L(G_1) = L(G_2)$ pro dvě bezkontextové gramatiky G_1, G_2,
2. problém inkluze jazyků bezkontextových gramatik, tj. otázka, zda $L(G_1) \subseteq L(G_2)$ pro dvě bezkontextové gramatiky G_1, G_2.

Důkaz. Důkaz lze vést pomocí techniky redukce. Více v pozdějších přednáškách o nerozhodnutelnosti. $\square$

Regularita

Definice 3.12 Bezkontextová gramatika $G = (N, \Sigma, P, S)$ má vlastnost sebevložení, jestliže existují $A \in N$ a $u, v \in \Sigma^+$ takové, že $A \Rightarrow^+ uAv$ a A není zbytečný nonterminál. Bezkontextový jazyk má vlastnost sebevložení, jestliže každá gramatika, která jej generuje, má vlastnost sebevložení.

Věta 3.12 Bezkontextový jazyk má vlastnost sebevložení právě tehdy, když není regulární.

Důkaz. Můžeme využít GNF – blíže viz doporučená literatura. □

- Problém, zda daná bezkontextová gramatika generuje regulární jazyk, není algoritmicky rozhodnutelný.

Deterministické zásobníkové automaty

Deterministický zásobníkový automat

Definice 3.13 Zásobníkový automat $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ nazýváme deterministický zásobníkový automat (DZA), jestliže pro každé $q \in Q$ a $z \in \Gamma$ platí buď

- $\forall a \in \Sigma : |\delta(q, a, z)| \leq 1 \wedge \delta(q, \varepsilon, z) = \emptyset$, nebo
- $\forall a \in \Sigma : \delta(q, a, z) = \emptyset \wedge |\delta(q, \varepsilon, z)| \leq 1$.

Definice 3.14 Nechť $L = L(P)$, kde P je deterministický zásobníkový automat. Jazyk L se pak nazývá deterministickým bezkontextovým jazykem.

Příklad 3.13 Uvažujme gramatiku $G = (\{X, Y\}, \{a, b, c\}, P, X)$ s pravidly:

$$\begin{array}{lcl} X & \longrightarrow & aXa \mid cYc \mid b \\ Y & \longrightarrow & aYbX \mid c \end{array}$$

Jedná se o LL(1) gramatiku a tudíž můžeme sestrojit DZA
$P = (\{q\}, \{a, b, c\}, \{X, Y, a, b, c\}, \delta, q, X, \emptyset)$ takový, že $L(G) = L(P)$ a provádí $LL(1)$ analýzu :

$$\begin{array}{lll} \delta: & \delta(q, a, X) = (q, Xa) & \delta(q, c, Y) = (q, \varepsilon) \\ & \delta(q, b, X) = (q, \varepsilon) & \delta(q, a, a) = (q, \varepsilon) \\ & \delta(q, c, X) = (q, Yc) & \delta(q, b, b) = (q, \varepsilon) \\ & \delta(q, a, Y) = (q, YbX) & \delta(q, c, c) = (q, \varepsilon) \end{array}$$

Skutečně, např. derivaci $X \Rightarrow aXa \Rightarrow aba$ odpovídá přijímající posloupnost konfigurací $(a, aba, X) \vdash (q, ba, Xa) \vdash (q, a, a) \vdash (q, \varepsilon, \varepsilon)$.

Neekvivalence NZA a DZA

Věta 3.13 DZA mají striktně menší vyjadřovací sílu než NZA.

Důkaz. (idea) Bezkontextový jazyk $L = \{ww^R \mid w \in \Sigma^+\}$ nelze přijímat žádným DZA. Neformálně řečeno, DZA nemá možnost uhádnout, kdy končí w a začíná w^R. $\square$

❖ *Poznámka:* Jiná možnost důkazu věty je přes následně uvedenou uzavřenost jazyků DZA vůči doplňku a přes uvážení, že $\overline{\{a^n b^n c^n | n \geq 1\}}$ je bezkontextový jazyk.

❖ Problém, zda daný bezkontextový jazyk je jazykem nějakého DZA, není obecně rozhodnutelný (podobně jako není rozhodnutelná víceznačnost).

❖ Deterministické bezkontextové jazyky mají jiné uzávěrové vlastnosti. Více viz opora (nezkouší se).