Bezkontextové jazyky

Jazyky typu 2

Definice 4.1 Gramatika $G = (N, \Sigma, P, S)$ si nazývá bezkontextovou gramatikou, jestliže všechna pravidla z P mají tvar

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

Lemma 4.1 Každý regulární jazyk je jazykem bezkontextovým.

❖ Proč studujeme bezkontextové jazyky?

Příklad 4.1 Jazyk $L = \{a^n b^n \mid n \geq 0\}$, jak víme, není jazykem regulárním, je však jazykem bezkontextovým:
$L = L(G)$ kde
$G = (\{S\}, \{a, b\}, \{S \to aSb, S \to \varepsilon\}, S)$

Příklad bezkontextové gramatiky

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

Příklad 4.2 $G = (\{S, A, B\}, \{a, b, c\}, P, S)$, kde P obsahuje pravidla

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

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 4.2 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 $N \cup \Sigma$; kořen stromu je označen výchozím symbolem S.
2. Přímé derivaci $v_{i-1} \Rightarrow v_i, i = 0, 1, \ldots, k$ 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,

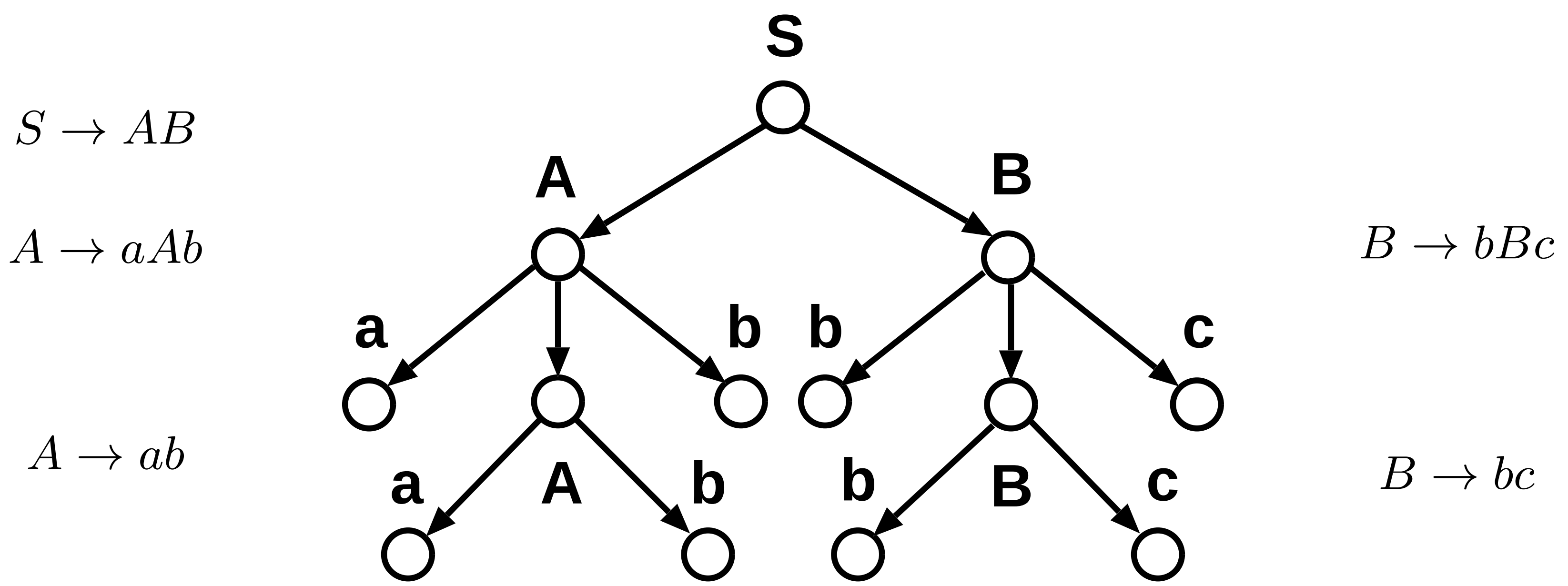
   odpovídá právě n hran $(A, X_j), j = 1, \ldots, n$ vycházejících z uzlu A, jež jsou uspořádány zleva doprava v pořadí $(A, X_1), (A, X_2), \ldots, (A, X_n)$.
3. Ohodnocení koncových uzlů derivačního stromu vytváří zleva doprava větnou formu nebo větu δ (plyne z 1. a 2.).

Příklad derivačního stromu

Příklad 4.3 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):



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 4.3 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 4.2 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:

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

resp.

$$\gamma_i B_i y_i \text{ kde } y_i \in \Sigma^*, B_i \in N, \gamma_i \in (N \cup \Sigma)^*$$

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

Fráze větné formy

Definice 4.4 Nechť $G = (N, \Sigma, P, S)$ je gramatika a nechť řetězec $\lambda = \alpha\beta\gamma$ je větná forma. Podřetězec β se nazývá frází větné formy vzhledem k nonterminálu A z N, jestliže platí:

$$S \Rightarrow^* \alpha A \gamma$$

$$A \Rightarrow^+ \beta$$

Podřetězec β je jednoduchou frází větné formy, jestliže platí:

$$S \Rightarrow^* \alpha A \gamma$$

$$A \Rightarrow \beta$$

Nejlevější jednoduchá fráze se nazývá l-frází.

Příklad 4.4 V gramatice z příkladu 4.1 nalezněte fráze věty $aabbbbcc$.

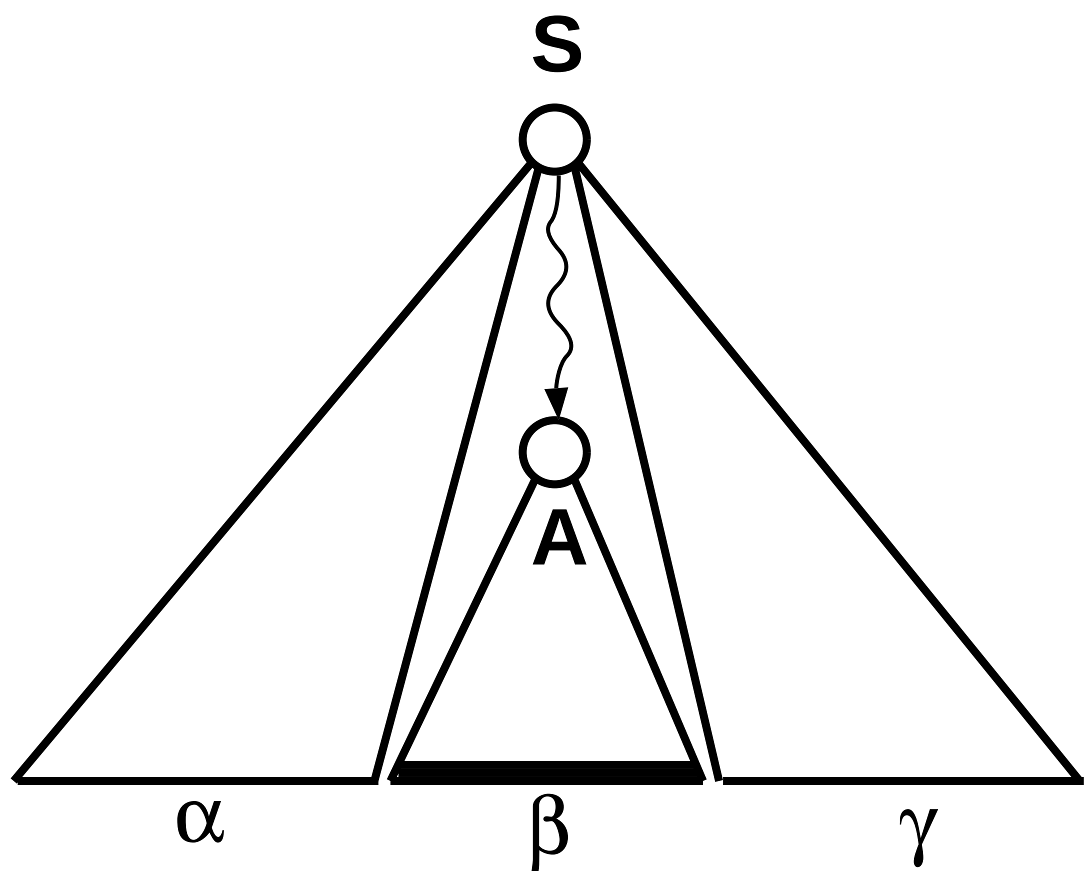
Nejdříve vytvořme libovolnou derivaci této věty (Příklad 4.2). Na základě této derivace získáme následující fráze k příslušným nonterminálům:

Fráze	Nonterminál
$aabbbbcc$	S
$aabb$	A^1
ab	A^2
$bbcc$	B^1
bc	B^2

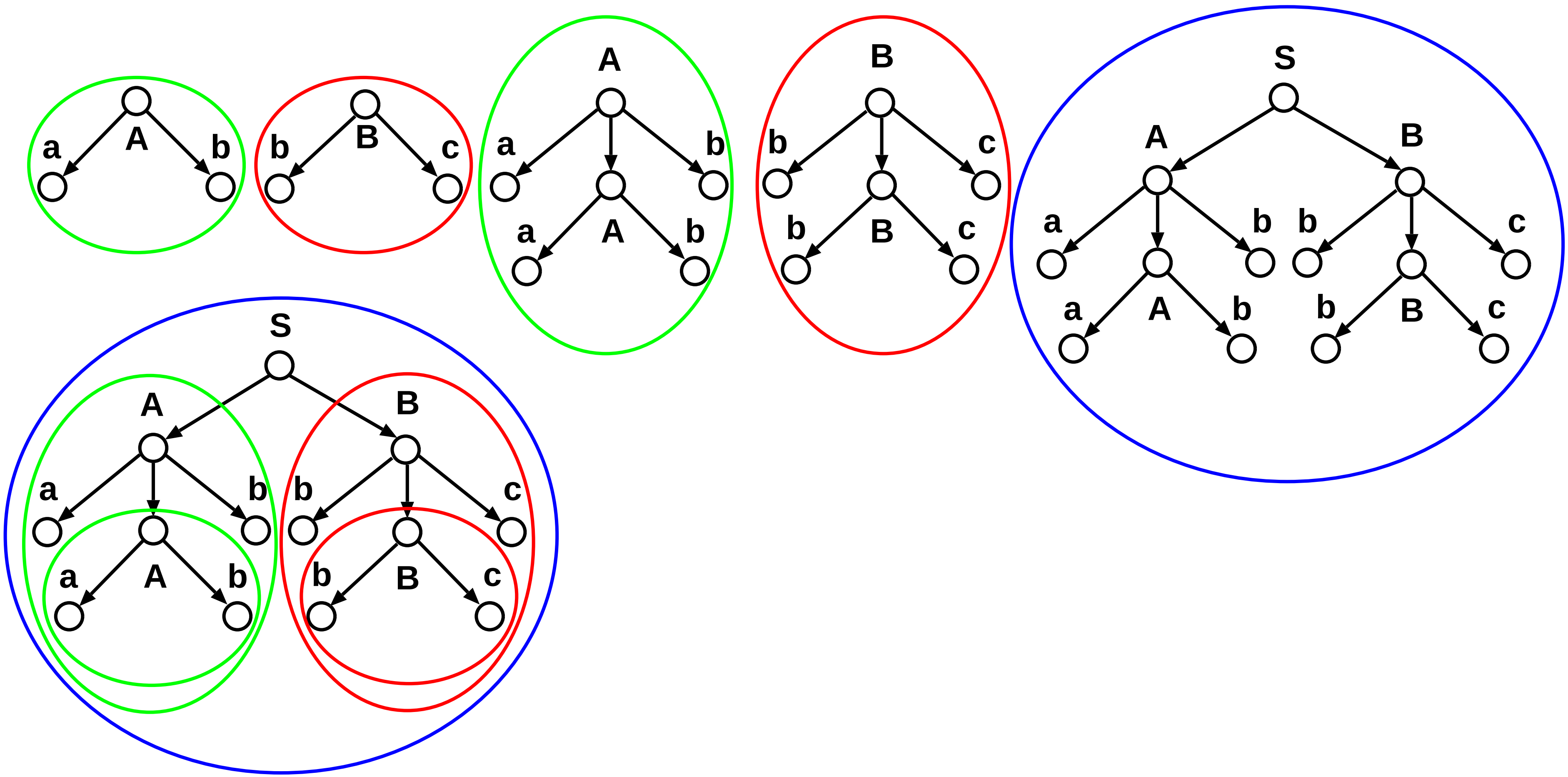
Fráze ab a bc jsou jednoduché, ab je l-fráze.

Vztah fráze a derivačního stromu

❖ Podstrom derivačního stromu odpovídá frázi příslušné větné formy. Fráze je tvořena koncovými uzly podstromu. Jednoduchá fráze odpovídá podstromu, jenž je výsledkem přímé derivace $A \Rightarrow \beta$, a jeho hloubka je rovna jedné. Situaci ilustruje následující obrázek.



Příklad 4.5 Podstromy derivačního stromu věty $aabbbbcc$ (příklad 4.3) jsou následující stromy odpovídající též frázím z předchozího příkladu.



Víceznačnost gramatik

Definice 4.5 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.

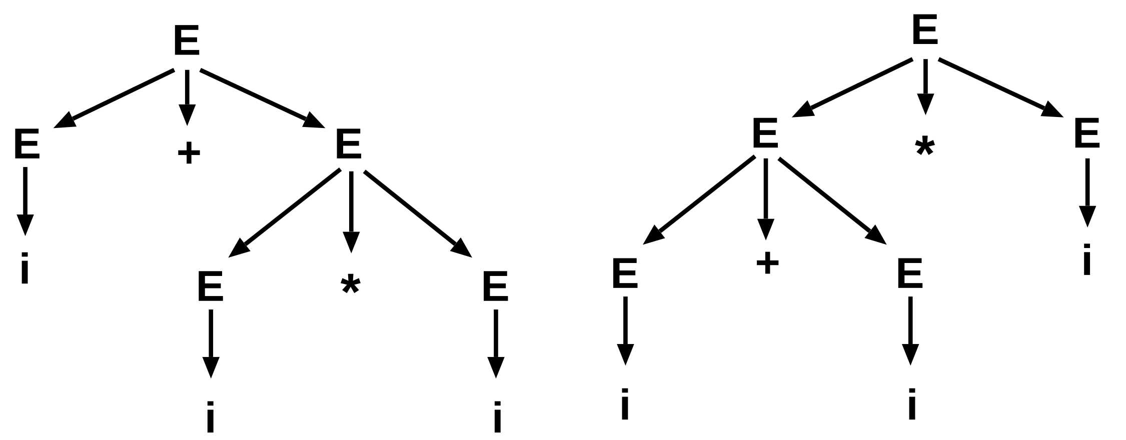
- Problém víceznačnosti gramatik je nerozhodnutelný, tj. neexistuje algoritmus, který by byl schopen v konečném čase rozhodnout, zda daná gramatika je nebo není víceznačná.
- Víceznačnost gramatiky je pokládána za negativní rys (vede k větám, které mají několik interpretací). Na druhé straně může být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

Víceznačnost gramatik

Příklad 4.6 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).

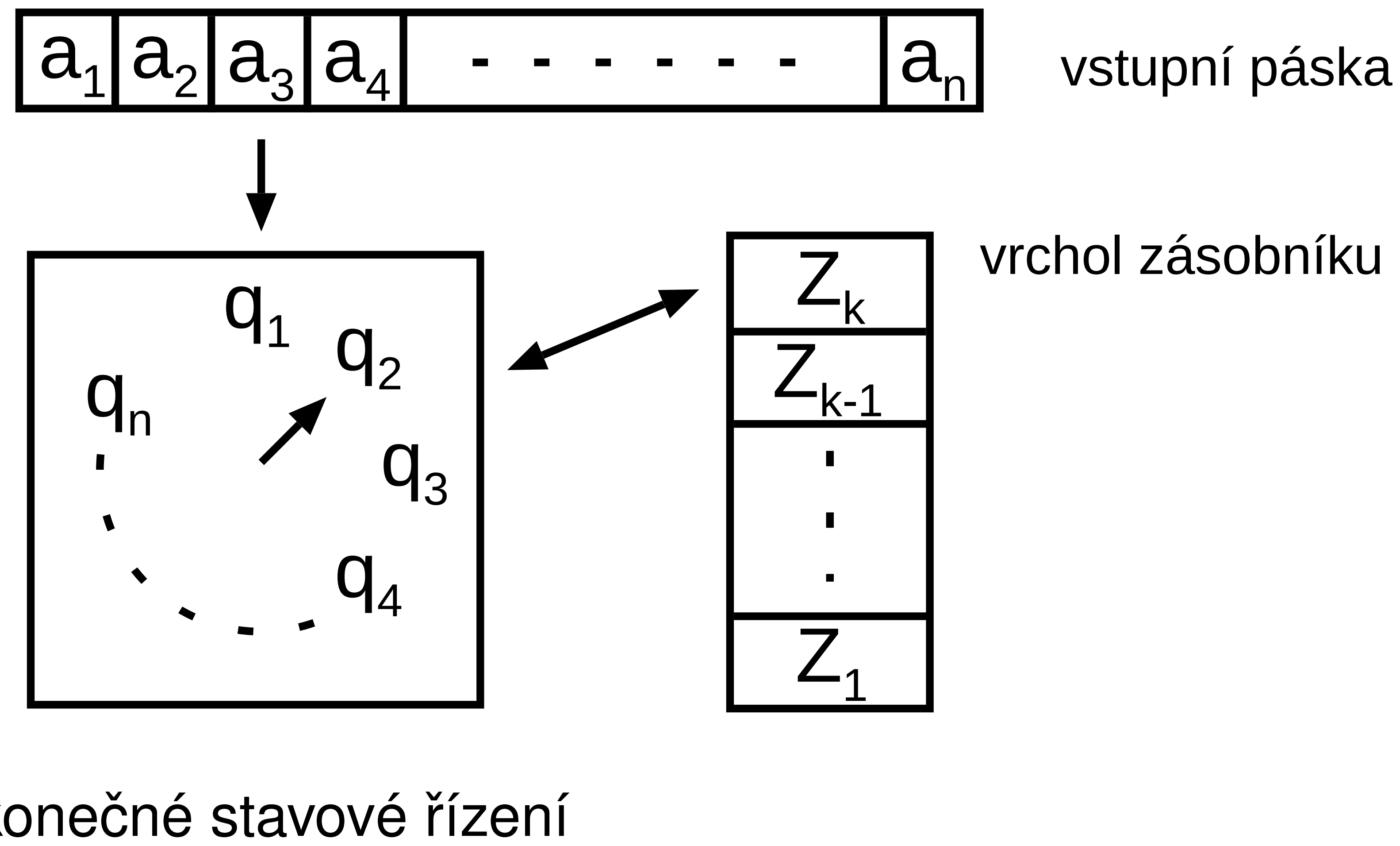
Příklad 4.7 Jednoznačnou gramatikou generující tentýž jazyk je gramatika $G = (\{E, T, F\}, \{+, -, *, /, (,), i\}, P, E)$ s množinou přepisovacích pravidel P definovanou následujícím způsobem:

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

Zásobníkové automaty

Základní schéma

Schéma zásobníkového automatu:



Základní definice

Definice 4.6 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 $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$
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 4.7 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:

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

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 $\vdash_P^i, \vdash_P^*, \vdash_P^+$ 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

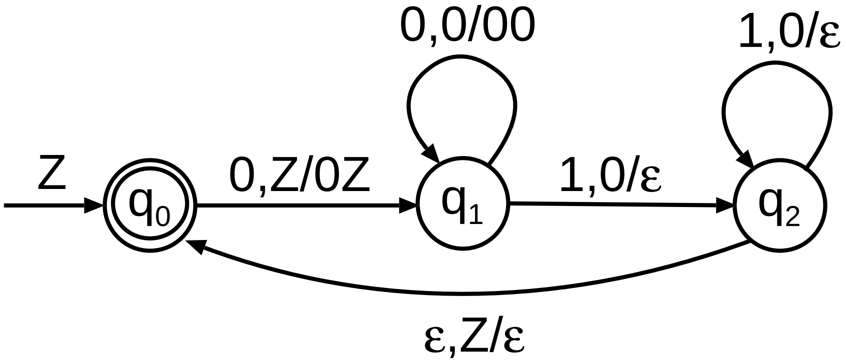
Příklad 4.8 Sestrojme zásobníkový automat, který přijímá jazyk $L = \{0^n 1^n \mid n \geq 0\}$.
– Řešením je $P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{Z, 0\}, \delta, q_0, Z, \{q_0\})$, kde

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

– 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:



Varianty zásobníkových automatů

Rozšířený zásobníkový automat

Definice 4.8 Rozšířený zásobníkový automat P je sedmice $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde δ je přechodová funkce definovaná takto:

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^* \to 2^{Q \times \Gamma^*}$$

Ostatní složky mají stejný význam jako v definici 5.2.

Příklad 4.9 Rozšířený zásobníkový automat $P = (\{p, q\}, \{a, b\}, \{a, b, S, Z\}, \delta, q, Z, \{p\})$, kde

$$\begin{array}{lll} \delta(q, a, \varepsilon) = \{(q, a)\} & \delta(q, b, \varepsilon) = \{(q, b)\} & \delta(q, \varepsilon, \varepsilon) = \{(q, S)\} \\ \delta(q, \varepsilon, aSa) = \{(q, S)\} & \delta(q, \varepsilon, bSb) = \{(q, S)\} & \delta(q, \varepsilon, SZ) = \{(p, \varepsilon)\} \end{array}$$

přijímá jazyk $L = \{ww^R | w \in \{a, b\}^+\}$.

Např: $(q, aabbaa, Z) \vdash (q, abbaa, aZ) \vdash (q, bbaa, aaZ) \vdash (q, baa, baaZ) \vdash (q, baa, SbaaZ) \vdash (q, aa, bSbaaZ) \vdash (q, aa, SaaZ) \vdash (q, a, aSaa) \vdash (q, a, SaZ) \vdash (q, \varepsilon, aSaZ) \vdash (q, \varepsilon, SZ) \vdash (p, \varepsilon, \varepsilon)$

Ekvivalence RZA a ZA

Věta 4.1 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 symblů je v zásobníku automatu P.

Nahrazuje-li automat P k vrcholových symboů 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 $L(P) = L(P_1)$. $\Box$

ZA přijímající s vyprázdněním zás.

Definice 4.9 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 4.2 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 bezkontextových jazyků a jazyků přijímaných zásobníkovým automatem

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. $\mathcal{L}_2 \subseteq \mathcal{L}_P$
- a ke každému zásobníkovému automatu existuje ekvivalentní gramatika typu 2, tj. $\mathcal{L}_P \subseteq \mathcal{L}_2$

Pro důkaz inkluze $\mathcal{L}_2 \subseteq \mathcal{L}_P$ zkonstruujeme (redundantně) automaty modelující oba typy syntaktické analýzy příslušného bezkontextového jazyka.

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

Věta 4.3 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 4.10 Ke gramatice

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

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

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

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

Skutečně, např. derivaci

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

odpovídá posloupnost přechodů automatu P:

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

Věta 4.4 Nechť $G = (N, \Sigma, P, S)$ je bezkontextová gramatika. Pak lze ke gramatice G sestrojit RZA P takový, že $L(G) = L(P)$.

Důkaz. RZA P sestrojme tak, aby modeloval syntaktickou analýzu zdola nahoru. Nechť P je RZA

$$P = (\{q, r\}, \Sigma, N \cup \Sigma \cup \{\#\}, \delta, q, \#, \{r\})$$

kde δ je určena takto:

1. Je-li $A \rightarrow \alpha$ pravidlo z P, pak $\delta(q, \varepsilon, \alpha)$ obsahuje (q, A^R). - redukce
2. $\delta(q, a, \varepsilon) = \{(q, a)\}$ pro všechna $a \in \Sigma$ - shift
3. $\delta(q, \varepsilon, S\#) = \{(r, \varepsilon)\}$

Indukcí lze opět dokázat $L(G) = L(P)$.

□

Příklad 4.11 Na gramatiku

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

aplikujeme nyní větu 5.4. Výsledný RZA bude mít tvar:

$$P = (\{q, r\}, \{0, 1\}, \{0, 1, S, \#\}, \delta, q, \#, \{r\})$$

kde δ je definována takto

$$\begin{array}{ll}
\delta(q, \varepsilon, 0S1) = \{(q, S)\} & \text{redukce} \\
\delta(q, \varepsilon, 01) = \{(q, S)\} & \text{redukce} \\
\delta(q, 0, \varepsilon) = \{(q, 0)\} & \text{shift} \\
\delta(q, 1, \varepsilon) = \{(q, 1)\} & \text{shift} \\
\delta(q, \varepsilon, \#S) = \{(r, \varepsilon)\} &
\end{array}$$

Derivaci $S \Rightarrow 0S1 \Rightarrow 0011$ odpovídá posloupnosti konfigurací $(q, 0011, \#) \vdash (q, 011, \#0) \vdash (q, 11, \#00) \vdash (q, 1, \#001) \vdash (q, 1, \#0S) \vdash (q, \varepsilon, \#0S1) \vdash (q, \varepsilon, \#S) \vdash (r, \varepsilon, \varepsilon)$

Poznámka 4.1 Vrchol zásobníku uvádíme, pro lepší čitelnost, vpravo

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

Věta 4.5 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 $(r, X_1 X_2 \dots X_k) \in \delta(q, a, Z)$, $k \geq 1$, pak k P přidej pravidla tvaru
  $$[qZs_k] \to a[rX_1s_1][s_1X_2s_2]\dots[s_{k-1}X_ks_k]$$
  pro každou posloupnost stavů $s_1, s_2, \dots, s_k$ z množiny Q
- Jestliže $(r, \varepsilon) \in \delta(q, a, Z)$, pak k P přidej pravidlo $[qZr] \to a$ (pro $a \in \Sigma \cup \{\varepsilon\}$)
- Pro každý stav $q \in Q$ přidej k P pravidlo $S \to [q_0Z_0q]$

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

$\square$

Ekvivalence $\mathcal{L}_2 = \mathcal{L}_P$

Věta 4.6 **Třída bezkontextových jazyků a třída jazyků přijímaných zásobníkovými automaty jsou totožné.**

Důkaz. Přímý důsledek vět 5.4 a 5.5. $\square$