8. cvičení: Pumping lemma pro bezkontextové jazyky, zásobníkové automaty, nedeterministická syntaktická analýza

Příklad 7.15. Jednu odrážku udělejte pečlivě, v dalších se soustřeďte jen na to podstatné.
Dokažte, že následující jazyky nejsou bezkontextové

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

b) $L = \{a^n b^n c^n \mid n \geq 1\}$

c) $L = \{a^n b^m c^n d^m \mid n, m \geq 1\}$

Jazyk $L = \{wcw \mid w \in \{a, b\}^*\}$ není bezkontextový

Důkaz. Důkaz vedeme s pomocí obměny lemmatu o vkládání pro bezkontextové jazyky.
Nechť $n \in \mathbb{N}$ je libovolné, ale nadále pevné. Zvolíme $z = a^n b^n c a^n b^n$, pak jistě $z \in L$ a $|z| > n$. Uvažujeme libovolné rozdělení tvaru $z = uvwxy$, kde $|vwx| \leq n$ a $vx \neq \varepsilon$, pro tvar vx mohou nastat následující případy:

1. vx padle do první poloviny slova, před c, pak stačí zvolit $i = 0$: jistě odebereme nějaké a nebo b z první poloviny a tedy výsledek nepatří do L (první část je kratší než druhá)
2. vx padne „přes střed“ slova:
   (a) c je podslovem vx: pro volbu $i = 0$ dostáváme slovo bez c, tedy nepatří do L
   (b) jinak, pro volbu $i = 0$: odebereme nějaká b z první části slova nebo nějaká a z druhé, a tedy slovo nebude patřit do L, nikdy však nemůže nastat, že bychom odebrali zároveň a z první i druhé části (nebo b z obou), protože mezi mini je $n + 1$ jiných znaků a $|vwx| \leq n$
3. vx padle do druhé poloviny slova (za c), zvolíme $i = 0$: druhá část je kratší

Ve všech možných rozděleních jsme dostali $uv^i wx^i y \notin L$ a tedy L není bezkontextový. □

Vysvětlit proč $a^n c a^n$ nestačí (problém v 2b).

Jazyk $L = \{a^n b^n c^n \mid n \leq 1\}$ není bezkontextový:

Důkaz. $n \in \mathbb{N}$ libovolné, volíme $z = a^n b^n c^n \in L$, je delší než n. Rozdělení $z = uvwxy$, $|vwx| \leq n, vx \neq \varepsilon$:

1. vx padne do *a*ček: $i = 0$ rozbijeme počet a vzhledem k b
2. vx padne na přelom ab: $i = 0$ rozbijeme počet a a b vzhledem k c
3. vx padne do *b*ček: $i = 0$ rozbijeme počet b vzhledem k a
4. vx padle na přelom bc: $i = 0$ rozbijeme počet b a c vzhledem k a
5. vx padne do *c*ček: $i = 0$ rozbijeme počet c vzhledem k a

Žádné rozdělení kdy by mohla být všechna 3 písmena v vx není, protože $|vwx| \leq n$, tedy máme všechna rozdělení.
Ve všech případech jsme dostali $uv^i wx^i y \notin L$ a tedy L není bezkontextový. □

Jazyk $L = \{a^n b^m c^n d^m \mid n, m \leq 1\}$ není bezkontextový:

Důkaz. $n \in \mathbb{N}$ libovolné, volíme $z = a^n b^n c^n d^n \in L, |z| > n$.

Idea: kdykoli tam máme a nebo c tak rozbijeme jejich vztah a kdykoli tam máme b nebo d tak rozbijeme to, nemůžeme tam mít obojí (a a c, nebo b a d) najednou, protože $|vwx| \leq n$... □

Příklad 8.1. Zmiňte prosím, že byl definován pojem *krok výpočtu*, ale pojem *výpočet* pro PDA definován nebyl. Lze si představit hned několik definic, které kromě zjevných požadavků splňují i tyto:

1. Musí se přečíst celý vstup. V tom případě by v příkladu existoval jen 1 výpočet.
2. Musí se číst „dokud to lze". V tomto případě existují 4 výpočty.
3. Stačí přečíst libovolnou část vstupu. V tom případě je výpočtů hodně.

Daný ZA $A = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b, c, d\}, \{Z, A\}, \delta, q_0, Z, \{q_4\})$

$$\begin{array}{ll}
\delta(q_0, a, Z) = \{(q_0, AZ)\} & \delta(q_0, a, A) = \{(q_0, AA)\} \\
\delta(q_0, b, A) = \{(q_1, \epsilon)\} & \delta(q_1, b, A) = \{(q_1, \epsilon)\} \\
\delta(q_1, \epsilon, A) = \{(q_2, A), (q_3, A)\} & \delta(q_2, c, A) = \{(q_2, \epsilon)\} \\
\delta(q_3, d, A) = \{(q_3, \epsilon)\} & \delta(q_2, \epsilon, Z) = \{(q_4, Z)\} \\
\delta(q_3, \epsilon, Z) = \{(q_4, Z)\} &
\end{array}$$

- Načrtněte stavový diagram ZA A.
- Naznačte 4 různé výpočty na vstupu a^4b^2c (stačí na obrázku).
- Popište jazyk $L(A)$.

Zásobník se značí s vrcholem **vlevo**.

stavový diagram, výpočty na a^4b^2c

$$\begin{tikzcd}
{} \arrow[r] & q_0Z \arrow[r, "a"] & q_0AZ \arrow[r, "a"] \arrow[d, "b"] & q_0AAZ \arrow[r, "a"] \arrow[d, "b"] & q_0AAAZ \arrow[r, "a"] \arrow[d, "b"] & q_0AAAAZ \arrow[r, "a"] \arrow[d, "b"] & \cdots \\
& & q_1Z & q_1AZ \arrow[l, "b"'] \arrow[d, "\varepsilon"] \arrow[dd, bend left, "\varepsilon"] & q_1AAZ \arrow[l, "b"'] \arrow[d, "\varepsilon"] \arrow[dd, bend left, "\varepsilon"] & q_1AAAZ \arrow[l, "b"'] \arrow[d, "\varepsilon"] \arrow[dd, bend left, "\varepsilon"] & \cdots \arrow[l, "b"'] \\
& & q_2Z \arrow[dl, "\varepsilon"] & q_2AZ \arrow[l, "c"'] & q_2AAZ \arrow[l, "c"'] & q_2AAAZ \arrow[l, "c"'] & \cdots \arrow[l, "c"'] \\
& q_4Z & q_3Z \arrow[l, "\varepsilon"'] & q_3AZ \arrow[l, "d"'] & q_3AAZ \arrow[l, "d"'] & q_3AAAZ \arrow[l, "d"'] & \cdots \arrow[l, "d"']
\end{tikzcd}$$

Intuice k přechodové funkci:
V q_0 pod a přidáváme A, pod b se přesuneme do q_1 a odebíráme A (musí být alespoň jedno A na zásobníku, tedy alespoň jedno a přečteno. V q_1 čteme b a odebíráme A, nanejvíš o 1 b méně než bylo a, můžeme přejít bez čtení a změny zásobníku do q_2, q_3. V q_2 čteme c a odebíráme A, tedy načteme nejvýše tolik c

kolik zbývá A na zásobníku, když odstraníme všechna A přejdeme do q_4. V q_3 totéž jako v q_2, jen čteme d. q_4 nečte nic, jen akceptuje (pokud jsme na konci slova). Celkem musí ve slově být alespoň jedno b a alespoň jedno z c, d, a proto v něm musí být alespoň dvě a.

Tedy $L = \{a^n b^k c^{n-k} \mid n1 \le k \le n-1\} \cup \{a^n b^k d^{n-k} \mid 1 \le k \le n-1\}$.

Příklad 8.3. Udělejte pořádně aspoň dvě odrážky včetně c). Konstruujte ZA (akceptující koncovým stavem nebo prázdným zásobníkem) pro jazyky:

a) $L = \{a^i b^j \mid i \neq j,\ i, j \ge 0\}$

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

c) $L = \{a^{3n} b^{2n} \mid n \ge 1\}$

d) $L = \{a^{3n+2} b^{2n-1} \mid n \ge 1\}$

e) $L = \{w \mid w \in \{a, b, c\}^*; \#_a(w) = \#_b(w)\}$

f) $L = \{w \mid w \in \{a, b, c\}^*; \#_a(w) \neq \#_b(w)\}$

g) $L = \{a^k b^j \mid 1 \le j \le k \le 2j\}$

h) $L = \{a^{n+m} b^{m+p} c^{p+n} \mid m, p, n \ge 1\}$

i) $L = \{a^i b^j c^j \mid i, j \ge 1\} \cup \{a^k b^k c^m \mid k, m \ge 1\}$

j) $L = \{a^{k_1} b a^{k_2} b \dots b a^{k_r} \mid r > 1,\ k_i \ge 1\ (i = 1, \dots, r;$ existuje $p, s : p \neq s, k_p = k_s)\}$

řešení:

a) $A = (\{q_1, q_2, q_a, q_b\}, \{a, b\}, \{Z, A\}, \delta, q_1, Z, \{q_a, q_b\})$ akceptuje akceptujícím stavem. V q_1 načítáme a a zapamatováváme jejich počet. V q_2 načítáme b a odečítáme ze zapamatovaného počtu a. q_b je stav kam se dostaneme, pokud je b více, než a (a můžeme stále přidávat b). q_a značí, že a je více než b, ale protože už jsme mohli načíst nějaká b, nemůžeme načíst další a.

$$
\begin{array}{lll}
\delta(q_1, a, Z) = \{(q_1, AZ)\} & \delta(q_2, b, Z) = \{(q_b, Z)\} & \delta(q_b, b, Z) = \{(q_b, Z)\} \\
\delta(q_1, a, A) = \{(q_1, AA)\} & \delta(q_2, b, A) = \{(q_2, \varepsilon)\} & \\
\delta(q_1, b, Z) = \{(q_b, Z)\} & \delta(q_2, \varepsilon, A) = \{(q_a, A)\} & \\
\delta(q_1, b, A) = \{(q_2, \varepsilon)\} & & \\
\delta(q_1, \varepsilon, A) = \{(q_a, a)\} & &
\end{array}
$$

Bylo by dobré zdůraznit, že automat může obsahovat nedeterminismus i pokud jsou všechny množiny v přechodech nejvýše jednoprvkové.

c) Budeme na zásobníku počítat *a*čka. Za každé 3 *a*čka budeme očekávat 2 *b*čka, několik řešení:

1. přidat 2 X na zásobník za každé a, odebrat 3 X za každé b

$$A = (\{q_a, q_b, q_{2X}, q_{1X}\}, \{a, b\}, \{Z, X\}, \delta, q_a, Z, \emptyset)$$

akceptuje prázdným zásobníkem

$$
\begin{array}{ll}
\delta(q_a, a, Z) = \{(q_a, XXZ)\} & \delta(q_b, b, X) = \{(q_{2X}, \varepsilon)\} \\
\delta(q_a, a, X) = \{(q_a, XXX)\} & \delta(q_{2X}, \varepsilon, X) = \{(q_{1X}, \varepsilon)\} \\
\delta(q_a, b, X) = \{(q_{2X}, \varepsilon)\} & \delta(q_{1X}, \varepsilon, X) = \{(q_b, \varepsilon)\} \\
 & \delta(q_b, \varepsilon, Z) = \{(q_b, \varepsilon)\}
\end{array}
$$

Poznáka: Zásobníkový automat (nerozšířený) nemůže přečíst, a tedy ani smazat, více než 1 znak ze zásobníku v jednom kroku, proto potřebujeme pomocné stavy q_{2X} a q_{1X} které slouží jen k odmazání dalších 2 X ze zásobníku.

2. přidat 2 X na zásobník za každé 3 a, odebrat 1 za b
3. přidat 1 X na zásobník za každá 3 a, odebrat 1 za každá 2 b (ukončit načítání aček a začít s bčky pak umožníme jen pokud je počet a dělitelný 3 a zároveň jsme již viděli nějaká ačka)

$$A' = (\{q_{a0}, q_{a1}, q_{a2}, q_{b0}, q_{b1}, q_{acc}\}, \{a, b\}, \{Z, X\}, \delta, q_{a0}, Z, \{q_{acc}\})$$

Akceptuje akceptujícím stavem.

$$\begin{aligned}
\delta(q_{a0}, a, Z) &= \{(q_{a1}, Z)\} \\
\delta(q_{a0}, a, A) &= \{(q_{a1}, A)\} \\
\delta(q_{a1}, a, Z) &= \{(q_{a2}, Z)\} \\
\delta(q_{a1}, a, A) &= \{(q_{a2}, A)\} \\
\delta(q_{a2}, a, Z) &= \{(q_{a0}, AZ)\} \\
\delta(q_{a2}, a, A) &= \{(q_{a0}, AA)\} \\
\delta(q_{a0}, b, A) &= \{(q_{b1}, A)\}
\end{aligned}
\qquad
\begin{aligned}
\delta(q_{b0}, b, A) &= \{(q_{b1}, A)\} \\
\delta(q_{b1}, b, A) &= \{(q_{b0}, \varepsilon)\} \\
\delta(q_{b0}, \varepsilon, Z) &= \{(q_{acc}, \varepsilon)\}
\end{aligned}$$

Příklad 8.6. Analýzu provádějte na slově *ababaa*.
Pro danou G navrhněte (rozšířený) ZA, který provádí syntaktickou analýzu:

a) shora dolů,

b) zdola nahoru.

V obou případech proveďte analýzu slova *ababaa*.
$G = (\{S, A, B\}, \{a, b\}, P, S)$, kde

$$\begin{array}{llllllll}
P = \{ & S & \to & \epsilon & \mid & abSA, & & \\
 & A & \to & AaB & \mid & aB & \mid & a, \\
 & B & \to & aSS & \mid & bA & \} &
\end{array}$$

Shora dolů

- simuluje levou derivaci
- jediný stav q, akceptuje prázdným zásobníkem
- zásobníková abeceda jsou terminály i neterminály gramatiky
- za každé pravidlo tvaru $A \to \alpha$ v gramatice přidáme (q, α) do $\delta(q, \varepsilon, A)$
- pro každý terminál $a \in \Sigma$ přidáme přechod $\delta(q, a, a) = \{(q, \varepsilon)\}$
- začínáme s kořenovým neterminálem na zásobníku
- gramatika střídavě „expanduje pravidla na zásobníku“ a „kontroluje, že písmeno na vrcholu zásobníku sedí se vstupem“

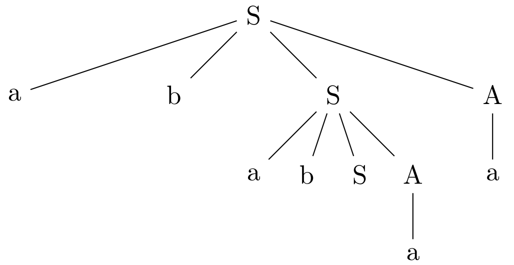
$$M = (\{q\}, \{a, b\}, \{a, b, S, A, B\}, \delta, q, S, \emptyset)$$

akceptuje prázdným zásobníkem

$$\begin{aligned}
\delta(q, \varepsilon, S) &= \{(q, \varepsilon), (q, abSA)\} \\
\delta(q, \varepsilon, A) &= \{(q, AaB), (q, aB), (q, a)\} \\
\delta(q, \varepsilon, B) &= \{(q, aSS), (q, bA)\}
\end{aligned}
\qquad
\begin{aligned}
\delta(q, a, a) &= \{(q, \varepsilon)\} \\
\delta(q, b, b) &= \{(q, \varepsilon)\}
\end{aligned}$$

Analýza *ababaa*:

Nejprve vytvoříme derivační strom:



Analyzátor shora dolů bude realizovat levou derivaci, tedy

$$S \Rightarrow abSA \Rightarrow ababSAA \Rightarrow ababAA \Rightarrow ababaA \Rightarrow ababaa$$

Pravidla tedy musíme na zásobníku expandovat v tomto pořadí (je však možné expandovat neterminál jedině pokud se nachází na vrcholu zásobníku, jinak musíme číst ze vstupu a tím „zkontrolovat“ terminál na zásobníku).

$$\begin{aligned}(q, ababaa, S) &\vdash^{\varepsilon}_{S\to abSA} (q, ababaa, abSA) \vdash^{a} (q, babaa, bSA) \vdash^{b} (q, abaa, SA) \vdash^{\varepsilon}_{S\to abSA} (q, abaa, abSAA)\\ &\vdash^{a} (q, baa, bSAA) \vdash^{b} (q, aa, SAA) \vdash^{\varepsilon}_{S\to\varepsilon} (q, aa, AA) \vdash^{\varepsilon}_{A\to a} (q, aa, aA)\\ &\vdash^{a} (q, a, A) \vdash^{\varepsilon}_{A\to a} (q, a, a) \vdash^{a} (q, \varepsilon, \varepsilon)\end{aligned}$$

Akceptovali jsme. Symboly na odvozovací relaci jsou jen pomocné, nahoře uvádíme znak vstupu, který se načetl, dole případné pravidlo, jehož expanze proběhla.

Zdola nahoru

- simuluje pravou derivaci v obráceném pořadí
- rozšířený PDA, zásobník píšeme obráceně (vrchol vpravo)
- zásobníková abeceda jsou terminály, neterminály a speciální symbol $\perp$ umožňující nám poznat dno zásobníku
- má dva stavy q, kde probíhá výpočet, a r, který slouží jen k akceptování
- pro každý terminál $a \in \Sigma$ přidáme do přechodové funkce $\delta(q, a, \varepsilon) = \{(q, a)\}$
- pro každé pravidlo $A \to \alpha$ přidáme do $\delta(q, \varepsilon, \alpha)$ dvojici (q, A)
- speciální pravidlo $\delta(q, \varepsilon, \perp S) = \{(r, \varepsilon)\}$ slouží k akceptování (pokud vstup nebyl dočten automat se zasekne)
- automat střídavě „přesouvá znaky ze vstupu na zásobník“ a „provádí redukci pravidel gramatiky na zásobníku z jejich pravé strany na levou“

$$R = (\{q, r\}, \{a, b\}, \{a, b, S, A, B, \perp\}, \delta, q, \perp, \{r\})$$

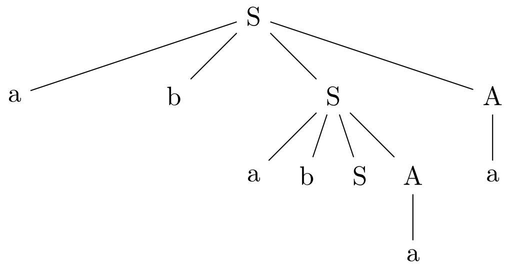
Rozšířený PDA akceptuje **vždy** akceptujícím stavem. Pozor, vrchol zásobníku je v analýze zdola nahoru otočen **vpravo**.

$$\begin{aligned}
\delta(q,\varepsilon,\varepsilon) &= \{(q,S)\} & \delta(q,a,\varepsilon) &= \{(q,a)\}\\
\delta(q,\varepsilon,abSA) &= \{(q,S)\} & \delta(q,b,\varepsilon) &= \{(q,b)\}\\
\delta(q,\varepsilon,AaB) &= \{(q,A)\} & \delta(q,\varepsilon,\perp S) &= \{(r,\varepsilon)\}\\
\delta(q,\varepsilon,aB) &= \{(q,A)\} & &\\
\delta(q,\varepsilon,a) &= \{(q,A)\} & &\\
\delta(q,\varepsilon,aSS) &= \{(q,B)\} & &\\
\delta(q,\varepsilon,bA) &= \{(q,B)\} & &
\end{aligned}$$

Pravidla redukují větnou formu na zásobníku. Pokud se nějaká pravá strana v gramatice objevuje opakovaně, pak jí musí odpovídat jedno pravidlo s více možnostmi přechodu.

Analýza *ababaa*:

Nejprve vytvoříme derivační strom:



Ze stromu odvodíme, že pravá derivace našeho slova je

$$S \Rightarrow abSA \Rightarrow abSa \Rightarrow ababSAa \Rightarrow ababSaa \Rightarrow ababaa$$

Analyzátor bude tuto derivaci realizovat v opačném pořadí a to vždy na vrcholu zásobníku (tj. například $S \to \varepsilon$ musíme provést v okamžiku, kdy máme načtené *abab*, protože S touto redukcí přidáme na vrchol zásobníku).

$$\begin{aligned}
(q,ababaa,\perp) &\overset{a}{\vdash} (q,babaa,\perp a) \overset{b}{\vdash} (q,abaa,\perp ab) \overset{a}{\vdash} (q,baa,\perp aba) \overset{b}{\vdash} (q,aa,\perp abab)\\
&\vdash^{\varepsilon}_{S\to\varepsilon} (q,aa,\perp ababS) \overset{a}{\vdash} (q,a,\perp ababSa) \vdash^{\varepsilon}_{A\to a} (q,a,\perp ababSA) \vdash^{\varepsilon}_{S\to abSA} (q,a,\perp abS)\\
&\overset{a}{\vdash} (q,\varepsilon,\perp abSa) \vdash^{\varepsilon}_{A\to a} (q,\varepsilon,\perp abSA) \vdash^{\varepsilon}_{S\to abSA} (q,\varepsilon,\perp S) \vdash^{\varepsilon}_{acc} (r,\varepsilon,\varepsilon)
\end{aligned}$$

Akceptovali jsme. Opět platí, že symboly na odvozovací relaci jsou jen pomocné, nahoře uvádíme znak vstupu, který se načetl, dole případné pravidlo, jehož redukce proběhla.