

9. Zásobníkové automaty

Formální jazyky a automaty

Jiří Balun

Obsah

1 Nedeterministický zásobníkový automat (PDA)

- Popis PDA
- Výpočet PDA
- Jazyk PDA

2 Ekvivalence PDA a CFG

- Převod CFG na PDA
- Převod PDA na CFG

3 Deterministický zásobníkový automat (DPDA)

- Popis DPDA
- Deterministické bezkontextové jazyky (DCFL)
- DPDA a jednoznačnost CFL

Nedeterministický zásobníkový automat (PDA)

Motivační příklad pro PDA

- pro bezkontextové jazyky chceme najít podobný výpočetní model jako jsou DFA/NFA pro regulární jazyky
 - CYK má složitost $O(n^3 \cdot |\mathcal{G}|)$ a musí zpracovat celý vstup délky n
 - chceme najít model, který bude číst a ověřovat vstup znak po znaku

Příklad

Chceme vytvořit výpočetní model, který by rozpoznával jazyk $\{ww^R \mid w \in \Sigma^*\}$.

Základní idea:

- 1 nejprve si musíme umět někam uložit první část řetězce w ,
- 2 poté chceme uložený řetězec srovnat s druhou částí w^R .

Řešení:

- pro uložení w potřebujeme potenciálně nekonečnou paměť (délka w není omezena),
- tato paměť musí fungovat na principu LIFO (aby výpočet probíhal efektivně jen jedním průchodem řetězce na vstupu),
- pro realizaci paměti použijeme datovou strukturu **zásobník**.

Popis PDA

Intuice: rozšíříme model konečných automatů o nekonečnou paměť v podobě datové struktury zásobníku (dostaneme nekonečně stavový model).

Definice

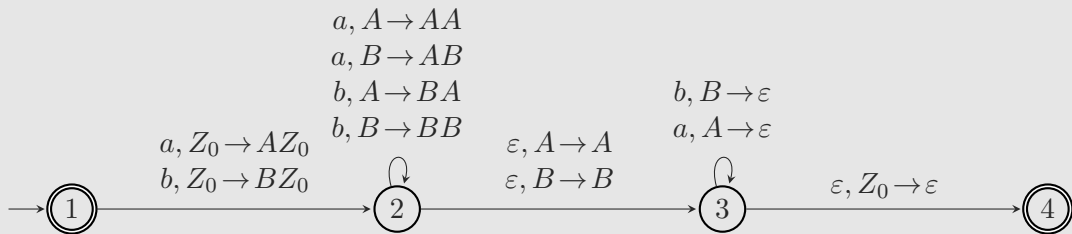
Nedeterministický zásobníkový automat (PDA z *pushdown automaton*) je reprezentován strukturou $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde:

- 1 Q je konečná množina stavů,
- 2 Σ je vstupní abeceda,
- 3 Γ je zásobníková abeceda,
- 4 δ je přechodová funkce ve tvaru $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$, která stavu, vstupnímu symbolu (nebo ε) a symbolu na vrcholu zásobníku přiřadí nějaký stav a řetězec zásobníkových symbolů,
- 5 q_0 je počáteční stav (platí pro něj $q_0 \in Q$),
- 6 Z_0 je symbol (platí $Z_0 \in \Gamma$), který je jako jediný v zásobníku na začátku výpočtu,
- 7 F je množina koncových/akceptujících stavů (platí $F \subseteq Q$).

Příklad: PDA

Příklad

PDA $P = (\{1, 2, 3, 4\}, \{a, b\}, \{A, B, Z_0\}, \delta, q_0, Z_0, \{1, 4\})$, který rozpoznává jazyk palindromů sudé délky $\{ww^R \mid w \in \{a, b\}^*\}$:



Interpretace přechodu ze stavu 1 do stavu 2 s popisem $a, Z_0 \rightarrow AZ_0$:

- čte ze vstupu symbol a ,
- z vrcholu zásobníku se odstraní symbol Z_0 (operace **pop**),
- poté se na zásobník uloží řetězec AZ_0 (operace **push**).

Konfigurace a výpočet PDA

Intuice: stav PDA popisuje kromě stavu z Q i aktuální hodnota jeho zásobníku, který má neomezenou velikost, proto má PDA potenciálně nekonečně mnoho možných stavů.

Definice

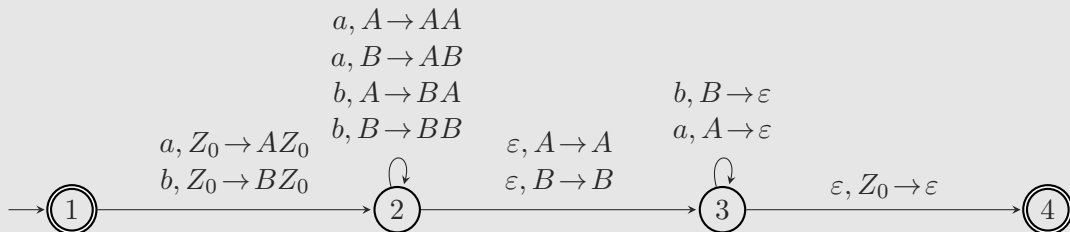
Konfigurace PDA je trojice stavu, řetězce na vstupu a obsahu zásobníku $\langle q, w, \gamma \rangle \in Q \times \Sigma^* \times \Gamma^*$, která popisuje aktuální stav výpočtu PDA.

- výpočet začíná v počáteční konfiguraci $\langle q_0, w, Z_0 \rangle$ s řetězcem $w \in \Sigma^*$ na vstupu
- $c_1 \vdash c_2$ značí, že z konfigurace c_1 je dosažitelná konfigurace c_2 v jednom kroku výpočtu
- přechod mezi konfiguracemi $\langle q, \hat{a}w, z\gamma \rangle \vdash \langle p, w, \alpha\gamma \rangle$ je možný pokud:
 - $(p, \alpha) \in \delta(q, \hat{a}, z)$ je přechod daného PDA
 - $\hat{a} \in \Sigma \cup \{\varepsilon\}$ je vstup přečtený v tomto kroku a $w \in \Sigma^*$ je zbytek vstupního řetězce
 - $z \in \Gamma$ je odebraný symbol ze zásobníku a $\gamma \in \Gamma^*$ je zbytek obsahu zásobníku
 - $\alpha \in \Gamma^*$ je řetězec, který se vloží na vrchol zásobníku
- $\vdash^*$ je relace (tranzitivní uzávěr $\vdash$), která popisuje dosažitelnost konfigurace
- PDA je nedeterministický, proto se může nacházet v libovolné z dosažitelných konfigurací (druhý pohled na nedeterminismus, PDA umí uhádnout správný přechod)

Příklad: výpočet PDA

Příklad (část 1/2)

PDA $P = (\{1, 2, 3, 4\}, \{a, b\}, \{A, B, Z_0\}, \delta, q_0, Z_0, \{1, 4\})$ pro jazyk $\{ww^R \mid w \in \{a, b\}^*\}$:



Možný průběh výpočtu P pro slovo aa

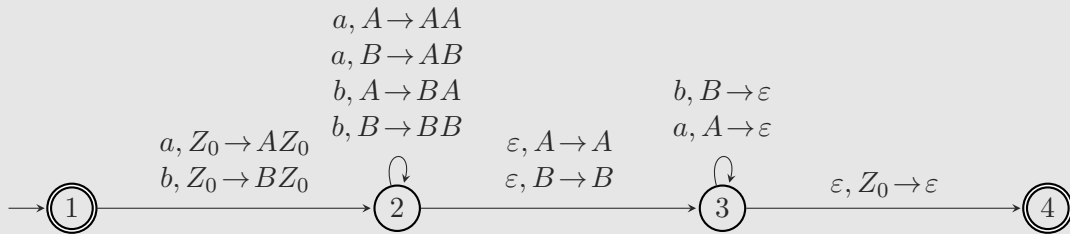
- přijímající výpočet: $\langle 1, aa, Z_0 \rangle \vdash \langle 2, a, AZ_0 \rangle \vdash \langle 3, a, AZ_0 \rangle \vdash \langle 3, \varepsilon, Z_0 \rangle \vdash \langle 4, \varepsilon, \varepsilon \rangle$
- zamítající výpočet: $\langle 1, aa, Z_0 \rangle \vdash \langle 2, a, AZ_0 \rangle \vdash \langle 2, \varepsilon, AAZ_0 \rangle \vdash \langle 3, \varepsilon, AAZ_0 \rangle \not\vdash \dots$

PDA musí „uhádnout“ správnou sekvenci přechodů, která vede do přijímací konfigurace.

Příklad: výpočet PDA

Příklad (část 2/2)

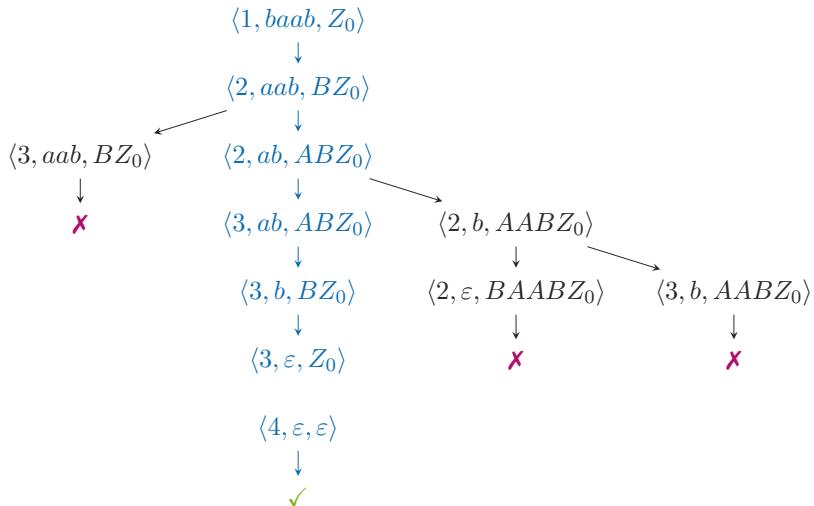
PDA $P = (\{1, 2, 3, 4\}, \{a, b\}, \{A, B, Z_0\}, \delta, q_0, Z_0, \{1, 4\})$ pro jazyk $\{ww^R \mid w \in \{a, b\}^*\}$:



Možný průběh výpočtu P pro slovo $baab$

- přijímající výpočet: $\langle 1, baab, Z_0 \rangle \vdash \langle 2, aab, BZ_0 \rangle \vdash \langle 2, ab, ABZ_0 \rangle \vdash \langle 3, ab, ABZ_0 \rangle \vdash \langle 3, b, BZ_0 \rangle \vdash \langle 3, \epsilon, Z_0 \rangle \vdash \langle 4, \epsilon, \epsilon \rangle$
- zamítající výpočet: $\langle 1, baab, Z_0 \rangle \vdash \langle 2, aab, BZ_0 \rangle \vdash \langle 3, aab, BZ_0 \rangle \not\vdash \dots$

Příklad: nedeterminismus PDA



Jazyk PDA

Intuice: PDA akceptuje slovo w , pokud je z počáteční konfigurace s w na vstupu dosažitelná přijímací konfigurace (ty lze definovat dvěma způsoby).

Definice

Definujeme dva různé jazyky pro PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$:

1 jazyk, který P přijímá **pomocí koncových stavů** (podobně jako u NFA):

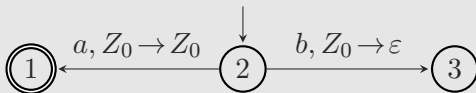
$$L(P) = \{w \in \Sigma^* \mid \langle q_0, w, Z_0 \rangle \vdash^* \langle q_f, \varepsilon, \alpha \rangle, \text{ kde } q_f \in F \text{ a } \alpha \in \Gamma^*\}$$

2 jazyk, který P přijímá **prázdným zásobníkem**:

$$N(P) = \{w \in \Sigma^* \mid \langle q_0, w, Z_0 \rangle \vdash^* \langle q, \varepsilon, \varepsilon \rangle, \text{ kde } q \in Q\}$$

Příklad

PDA $P = (\{1, 2, 3\}, \{a, b\}, \{Z_0\}, \delta, 2, Z_0, \{1\})$ s jazyky $L(P) = \{a\}$ a $N(P) = \{b\}$:



Ekvivalence jazyků $L(P)$ a $N(P)$ ($\Rightarrow$)

Věta

Ke každému PDA P existuje PDA P' takový, že $L(P) = L(P') = N(P')$.

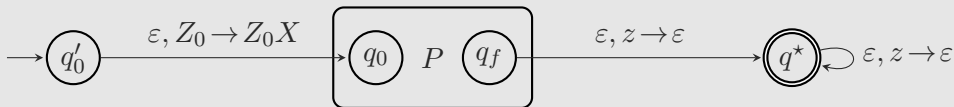
Důkaz

Idea: PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ přijímá řetězce pomocí koncových stavů, proto jej upravíme tak, aby z každého původního koncového stavu mohl přejít do nového koncového stavu q^* , kde může vyprázdnit svůj zásobník (poté už ale nesmí číst nic ze vstupu).

Zkonstruujeme PDA $P' = (Q \cup \{q'_0, q^*\}, \Sigma, \Gamma \cup \{X\}, \delta', q'_0, Z_0, \{q^*\})$ přidáním přechodů:

- $\delta(q'_0, \varepsilon, Z_0) = \{(q_0, Z_0X)\}$ z nového počátečního stavu q'_0 ,
- $\delta(q_f, \varepsilon, z) = (q^*, \varepsilon, z)$ z každého koncového stavu $q_f \in F$ a pro každý $z \in \Gamma \cup \{X\}$,
- $\delta(q^*, \varepsilon, z) = (q^*, \varepsilon, \varepsilon)$ pro každý $z \in \Gamma \cup \{X\}$.

Nový symbol X přidáváme proto, aby P' nemohl vyprázdnit zásobník jinde než v q^* .



Ekvivalence jazyků $L(P)$ a $N(P)$ ($\Leftarrow$)

Věta

Ke každému PDA P existuje PDA P' takový, že $N(P) = L(P') = N(P')$.

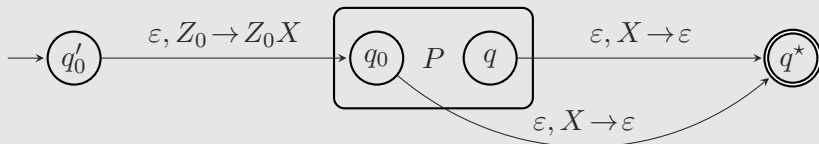
Důkaz

Idea: PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$ přijímá řetězce prázdným zásobníkem, proto jej upravíme tak, aby mohl zcela vyprázdnit svůj zásobník jen v novém koncovém stavu q^* .

Sestrojíme PDA $P' = (Q \cup \{q'_0, q^*\}, \Sigma, \Gamma \cup \{X\}, \delta', q'_0, Z_0, \{q^*\})$ přidáním přechodů:

- $\delta(q'_0, \varepsilon, Z_0) = \{(q_0, Z_0X)\}$ z nového počátečního stavu q'_0 ,
- $\delta(q, \varepsilon, X) = (q^*, \varepsilon, \varepsilon)$ z každého stavu $q \in Q$,

Nový symbol X přidáváme proto, aby P' nemohl vyprázdnit zásobník jinde než v q^* .



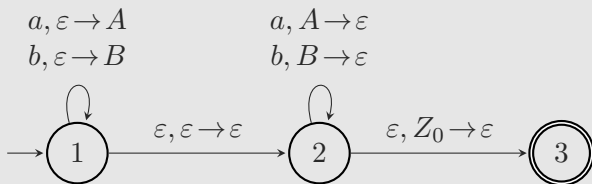
Konzervativní modifikace modelu PDA

Umožníme PDA dělat přechody bez odebrání vrchního symbolu zásobníku:

- přechodová funkce je potom ve tvaru $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \rightarrow 2^{Q \times \Gamma^*}$
- umožňuje například přechod $(p, \varepsilon) \in \delta(q, \varepsilon, \varepsilon)$, který lze udělat vždy
- tento model budeme používat **pouze** při návrhu PDA (pro zpřehlednění konstrukcí)

Příklad

PDA $P = (\{1, 2, 3, 4\}, \{a, b\}, \{A, B, Z_0\}, \delta, q_0, Z_0, \{1, 4\})$ s rozšířenými přechody, který rozpoznává jazyk $\{ww^R \mid w \in \{a, b\}^*\}$:



Kde například přechod $\delta(1, \varepsilon, \varepsilon) = \{(2, \varepsilon)\}$ mezi stavy 1 a 2 byl v předchozím řešení realizován dvěma přechody $\delta(2, \varepsilon, A) = \{(3, A)\}$ a $\delta(2, \varepsilon, B) = \{(3, B)\}$.

Ekvivalence PDA a CFG

Převod CFG na PDA

Intuice: PDA P v této konstrukci **simuluje nejlevější derivaci** vstupního řetězce v $\mathcal{G}$ na svém zásobníku.

Vstup: CFG $\mathcal{G} = (N, \Sigma, P, S)$

Výstup: PDA $P = (\{q\}, \Sigma, N \cup \Sigma, \delta, q, S, \emptyset)$ takový, že $L(G) = N(P)$

- 1 PDA P má pouze jeden stav, který je zároveň počáteční (nemá koncové stavy),
- 2 zásobníková abeceda P obsahuje všechny terminály a neterminály z $\mathcal{G}$,
- 3 počáteční zásobníkový symbol je počáteční neterminál S ,
- 4 do přechodové funkce δ přidáme přechody dvou typů:
 - **expanze:** pro každé pravidlo $A \rightarrow \alpha$ přidáme přechod $(q, \alpha) \in \delta(q, \varepsilon, A)$,
 - **srovnání:** dále pro každý terminál $a \in \Sigma$ přidáme přechod $\delta(q, a, a) = \{(q, \varepsilon)\}$.

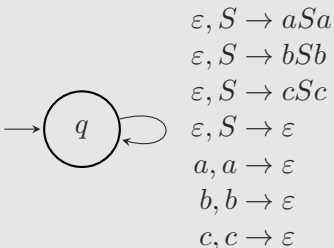
Příklad: převod CFG na PDA

Příklad

Mějme CFG $\mathcal{G} = (\{S\}, \{a, b, c\}, P, S)$ generující jazyk $L = \{ww^R \mid w \in \{a, b, c\}^*\}$:

$$S \rightarrow \varepsilon \mid aSa \mid bSb \mid cSc$$

a převedeme ji na PDA $P = (\{q\}, \{a, b, c\}, \{a, b, c, S\}, \delta, q, S, \emptyset)$ s $N(P) = L$:



Výpočet P pro slovo $baab$: $\langle q, baab, S \rangle \vdash \langle q, baab, bSb \rangle \vdash \langle q, aab, Sb \rangle \vdash \langle q, aab, aSab \rangle \vdash \langle q, ab, Sab \rangle \vdash \langle q, ab, ab \rangle \vdash \langle q, b, b \rangle \vdash \langle q, \varepsilon, \varepsilon \rangle$

Ekvivalence CFG a PDA ($\Rightarrow$)

Věta

Ke každé CFG $\mathcal{G}$ existuje PDA P takový, že $L(\mathcal{G}) = N(P)$.

Idea důkazu (část 1/3)

Mějme gramatiku $\mathcal{G} = (N, \Sigma, P, S)$, která je pro jednoduchost v **Greichachové normální formě** (GNF), tj. všechny pravidla jsou v jednom z těchto dvou tvarů:

- 1 $A \rightarrow aA_1 \dots A_k$, kde $k \geq 0$ (neterminál vygeneruje jeden terminál a libovolný počet neterminálů),
- 2 $S \rightarrow \varepsilon$ (pouze startovní neterminál generuje ε).

Každou CFG lze do tohoto tvaru převést (ponecháme bez důkazu). Libovolná větná forma nejlevější derivace gramatiky v GNF je ve tvaru $a_1 a_2 \dots a_n A_1 A_2 \dots A_m \in \Sigma^* N^*$.

Mějme PDA $P = (\{q\}, \Sigma, N \cup \Sigma, \delta, q, S, \emptyset)$, který byl získán dříve popsanou konstrukcí. Musíme ukázat, že pro všechny řetězce $w \in \Sigma^*$ platí $\langle q, w, S \rangle \vdash^* \langle q, \varepsilon, \varepsilon \rangle$ v P právě tehdy, když v $\mathcal{G}$ platí $S \Rightarrow_{lm}^* w$.

Idea důkazu (část 2/3)

Nejprve ukážeme $L(\mathcal{G}) \subseteq N(P)$. Mějme derivaci libovolného řetězce v $L(\mathcal{G})$:

$$\begin{aligned} S &\Rightarrow_{l_m}^* a_1 a_2 \dots a_n A_1 A_2 \dots A_m \\ &\Rightarrow_{l_m} a_1 a_2 \dots a_n b B_1 \dots B_k A_1 A_2 \dots A_m \end{aligned}$$

PDA P simuluje nejlevnější derivaci, proto po přčtení $a_1 a_2 \dots a_n$ má na zásobníku řetězec neterminálů $A_1 A_2 \dots A_m$. Přčtení dalšího symbolu b ze vstupu P získáme z:

- dle derivace v $\mathcal{G}$ musí existovat pravidlo ve tvaru $A_1 \rightarrow b B_1 \dots B_k$,
- proto dle konstrukce v P existuje přechod $(q, b B_1 \dots B_k) \in \delta(q, \varepsilon, A_1)$, který **expanduje** pravou stranu daného pravidla na zásobník,
- poté je na vstupu i na vrcholu zásobníku terminál b , proto můžeme použít přechod pro **srovnání** $(q, \varepsilon) \in \delta(q, b, b)$,
- tím jsme rozšířili přčtené symboly ze vstupu o b a zásobník o symboly $B_1 \dots B_k$.

Indukcí přes počet kroků výpočtu dostaneme, že pokud $S \Rightarrow^* w$, pak $\langle q, w, S \rangle \vdash^* \langle q, \varepsilon, \varepsilon \rangle$, a tedy $L(\mathcal{G}) \subseteq N(P)$.

Idea důkazu (část 3/3)

Nyní ukážeme $N(P) \subseteq L(\mathcal{G})$. Mějme výpočet P na $w = a_1 a_2 \dots a_n \in N(P)$ dán jako:

$$\langle q, w, S \rangle = \langle q, a_1 a_2 \dots a_n, S \rangle \vdash^* \langle q, \varepsilon, \varepsilon \rangle$$

pak v prvním a druhém kroku výpočtu P (dle konstrukce) musí použít:

- nejprve jeden z přechodů, který je ve tvaru **expanze** $(q, a_1 A_1 A_2 \dots A_k) \in \delta(q, \varepsilon, S)$, kde $S \rightarrow a_1 A_1 A_2 \dots A_k$ je pravidlo $\mathcal{G}$:
- po tomto kroku je na vstupu i na vrcholu zásobníku symbol a_1 , proto lze použít jedině pravidlo **srovnání** $(q, \varepsilon) \in \delta(q, a_1, a_1)$,
- a tedy platí, že v $\mathcal{G}$ lze odvodit $S \Rightarrow^* a_1 A_1 \dots A_k$.

Dále by P provedl tyto přechody $(q, a_2 B_1 B_2 \dots B_m) \in \delta(q, \varepsilon, A_1)$ a $(q, \varepsilon) \in \delta(q, a_2, a_2)$, kde $A_1 \rightarrow a_2 B_1 B_2 \dots B_m$ je opět pravidlo $\mathcal{G}$, a tedy platí $S \Rightarrow^* a_1 a_2 B_1 \dots B_m A_1 \dots A_k$.

Opakováním předchozí úvahy dostáváme $S \Rightarrow^* w$, protože obsah zásobníku po každé této dvojici přechodů odpovídá větné formě nejlevější derivace w v $\mathcal{G}$, a tedy $N(P) \subseteq L(\mathcal{G})$. $\square$

- aplikování konstrukce pro ekvivalenci $L(P) = N(P)$ dokazuje další tvrzení:
 - ke každé CFG $\mathcal{G}$ existuje PDA P takový, že $L(\mathcal{G}) = L(P)$

Převod PDA na CFG

Intuice: sestavíme CFG $\mathcal{G}$, kde každý neterminál $\langle pXq \rangle$ generuje právě všechny řetězce $w \in \Sigma^*$ takové, že platí $\langle p, w, X \rangle \vdash^* \langle q, \varepsilon, \varepsilon \rangle$. Například symboly $\langle q_0 Z_0 p \rangle$ generují řetězce, pro které P přejde ze stavu q_0 do nějakého stavu p a přitom odstraní vše ze zásobníku.

Vstup: PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

Výstup: CFG $\mathcal{G} = (N, \Sigma, P, S)$ taková, že $N(P) = L(\mathcal{G})$.

1 Množina neterminálů N obsahuje:

- symbol S pro počáteční neterminál,
- symbol $\langle pXq \rangle$ pro každou dvojici stavů $p, q \in Q$ a zásobníkový symbol $X \in \Gamma$.

2 Pravidla $\mathcal{G}$ jsou následující:

- pro každý stav $p \in Q$ přidáme pravidlo $S \rightarrow \langle q_0 Z_0 p \rangle$,
- pro přechod $(r, Y_1 \dots Y_k) \in \delta(q, \hat{a}, X)$, kde $\hat{a} \in (\Sigma \cup \{\varepsilon\})$ a $k \geq 0$ definujeme pravidlo:

$$\langle qXr_k \rangle \rightarrow \hat{a} \langle rY_1r_1 \rangle \langle r_1Y_2r_2 \rangle \dots \langle r_{k-1}Y_kr_k \rangle$$

pro všechny kombinace stavů $r_1, \dots, r_k \in Q$ (po přečtení $\hat{a}$ je P ve stavu r , přičemž odebereme X ze zásobníku. Odstranění nových symbolů $Y_1, \dots, Y_k$ ze zásobníku může vyžadovat čtení dalších symbolů ze vstupu a přechod do nějakých stavů $r_1, \dots, r_k$).

Ekvivalence CFG a PDA ($\Leftrightarrow$)

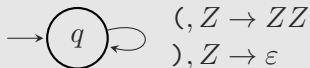
Věta

Ke každému PDA P existuje CFG $\mathcal{G}$ taková, že $N(P) = L(\mathcal{G})$. □

- důkaz vynecháme, dokazuje se korektnost předchozí konstrukce
- tj. že platí $S \Rightarrow \langle q_0 Z_0 p \rangle \Rightarrow^* w$ právě tehdy, když $\langle q_0, w, Z_0 \rangle \vdash^* \langle p, \varepsilon, \varepsilon \rangle$

Příklad

Mějme PDA $P = (\{q\}, \{ (,) \}, \{Z\}, \delta, q, Z)$, který detekuje první špatně umístěnou pravou závorku (před kterou se nevyskytovala odpovídající levá závorka):



Ekvivalentní CFG $\mathcal{G} = (\{S, \langle qZq \rangle\}, \{ (,) \}, P, S)$ získaná předchozí konstrukcí:

$$\begin{aligned} S &\rightarrow \langle qZq \rangle \\ \langle qZq \rangle &\rightarrow (\langle qZq \rangle \langle qZq \rangle \mid) \end{aligned}$$

Deterministický zásobníkový automat (DPDA)

Popis DPDA

Motivace: chceme oslabit model PDA tak, aby jejich jazyky šlo efektivně parsovat, ale aby pořád rozpoznávali zajímavé jazyky.

Definice

PDA $D = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je **deterministický** (DPDA) právě tehdy, když jsou splněny tyto podmínky:

- 1 $|\delta(q, \hat{a}, X)| \leq 1$ pro všechny $q \in Q$, $\hat{a} \in (\Sigma \cup \{\varepsilon\})$ a $X \in \Gamma$,
- 2 pokud $\delta(q, a, X) \neq \emptyset$ pro nějaké $a \in \Sigma$, pak $\delta(q, \varepsilon, X) = \emptyset$.

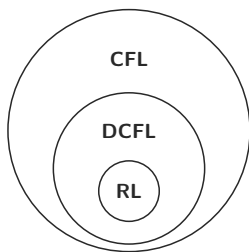
- PDA je deterministický pokud platí:
 - z každého stavu q existuje nejvýše jeden přechod pro každou kombinaci vstupního symbolu a a zásobníkového symbolu X (přechodová funkce nemusí být úplná)
 - navíc nesmí existovat volba mezi ε -přechodem a normálním přechodem pro stejný zásobníkový symbol X
- DPDA a PDA **nejsou** ekvivalentní
 - každý jazyk DPDA lze rozpoznat pomocí PDA
 - ale existují CFL, které nelze rozpoznat pomocí DPDA (například libovolný dědičně nejednoznačný jazyk)

Deterministické CFL

Definice

Bezkontextový jazyk L je **deterministický** (DCFL z *deterministic context free language*), pokud existuje DPDA D takový, že $L(D) = L$.

- ke každému $w \in L(D)$ existuje unikátní sekvence konfigurací popisující výpočet
 - **hlavní výhoda DFCL:** lze je parsovat v lineárním čase (proto většina programovacích jazyků patří do této třídy jazyků)
 - každý jazyk, který je rozpoznán DPDA, je **jednoznačný**
 - ale existují i jednoznačné CFL, které nelze rozpoznat pomocí DPDA
- třída jazyků DFCL obsahuje zároveň i všechny regulární jazyky



Příklady: DCFL a protipříklad

Příklad

Jazyk $\{ww^R \mid w \in \Sigma^*\}$ **není deterministický** (ale je jednoznačný), protože PDA, který jej rozpoznává, musí „uhádnout“ střed řetězce (krok kdy začít srovnávat obsah zásobníku).

Příklad

Jazyk $\{w\#w^R \mid w \in \Sigma^*\}$ **je deterministický**, protože střed řetězce je označen znakem $\#$ a DPDA, který jej rozpoznává, má jen jednu možnost kdy začít srovnávat obsah zásobníku.

$a, Z_0 \rightarrow AZ_0$

$b, Z_0 \rightarrow BZ_0$

$a, A \rightarrow AA$

$a, B \rightarrow AB$

$b, A \rightarrow BA$

$b, B \rightarrow BB$

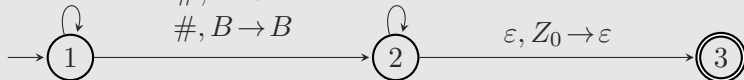
$\#, Z_0 \rightarrow Z_0$

$\#, A \rightarrow A$

$\#, B \rightarrow B$

$b, B \rightarrow \varepsilon$

$a, A \rightarrow \varepsilon$



DCFL a RL

Věta

Pokud je L regulární jazyk, pak existuje DPDA D takový, že $L(D) = L$.

Důkaz

Jazyk L je regulární, proto existuje DFA $A = (Q, \Sigma, \delta_A, q_0, F)$ takový, že $L(A) = L$. Zkonstruujeme DPDA $D = (Q, \Sigma, \{Z_0\}, \delta_D, q_0, Z_0, F)$ následovně:

- definujeme přechod $\delta_D(q, a, Z_0) = \{(p, Z_0)\}$ pokud $\delta_A(q, a) = p$, kde $p, q \in Q$,
- D v podstatě jen simuluje A , aniž by jakkoli použil svůj zásobník.

Platí $\langle q_0, w, Z_0 \rangle \vdash^* \langle p, \varepsilon, Z_0 \rangle$ právě tehdy, když $\hat{\delta}_A(q_0, w) = p$, a tedy $L(A) = L(D)$. □

- stejné tvrzení neplatí pro rozpoznání pomocí vyprázdnění zásobníku
 - existují regulární jazyky, které DPDA nemohou rozpoznat pomocí zásobníku
 - pro DPDA třídy jazyků $L(D)$ a $N(D)$ **nejsou stejné!**

Prefixová vlastnost

Intuice: řekneme, že jazyk L má **prefixovou vlastnost**, pokud pro libovolné dva různé řetězce $x, y \in L$ platí, že x není prefixem y .

Věta

DCFL L má prefixovou vlastnost právě tehdy, když existuje DPDA D takový, že platí $N(D) = L$. □

- pokud DCFL nemá tuto prefixovou vlastnost, pak jej DPDA rozpoznají jen pomocí koncových stavů

Příklad

Mějme regulární jazyk $L = \{a\}^*$, který **nemá** prefixovou vlastnost, proto **neexistuje** DPDA D , pro který platí $N(D) = L$.

Jazyk L ale můžeme upravit na $L' = L \cdot \$$ tím, že za každé slovo přidáme nový ukončovací symbol $\$$. Nový L' už má prefixovou vlastnost, protože například $aa\$$ není prefixem $aaa\$$.

DPDA a jednoznačnost CFL

Věta

Pokud $L = L(D)$ pro nějaký DPDA D , pak existuje jednoznačná gramatika generující L .

Idea důkazu

Definujeme jazyk $L' = L \cdot \$$, který má prefixovou vlastnost (kde $\$$ je nový ukončovací znak). Jazyk L je DCFL, proto existuje DPDA D' takový, že $N(D') = L'$.

Pokud aplikujeme konstrukci pro převod PDA na CFG na deterministický PDA D' , dostaneme jednoznačnou gramatiku $\mathcal{G}'$ pro jazyk $L(\mathcal{G}') = N(D') = L'$:

- D' je DPDA, proto existuje unikátní výpočet pro každé $w \in N(D')$,
- každý přechod $(r, Y_1 \dots Y_k) \in \delta(q, \hat{a}, X)$ udává v $\mathcal{G}'$ jediné aplikovatelné pravidlo ve tvaru $\langle qXr_k \rangle \rightarrow \hat{a}\langle rY_1r_1 \rangle \langle r_1Y_2r_2 \rangle \dots \langle r_{k-1}Y_kr_k \rangle$ pro danou sekvenci konfigurací,
- proto v $\mathcal{G}'$ existuje unikátní nejlevější derivace každého $w \in L(\mathcal{G}')$, tedy je jednoznačná.

Nyní z $\mathcal{G}'$ vytvoříme jednoznačnou CFG $\mathcal{G}$ takovou, že $L(\mathcal{G}) = L$:

- z terminálního symbolu $\$$ v $\mathcal{G}'$ uděláme nově neterminál v $\mathcal{G}$,
- přidáme do $\mathcal{G}$ nové pravidlo $\$ \rightarrow \varepsilon$ (čímž neporušíme jednoznačnost).

