Formální jazyky a automaty

Algoritmus Cocke-Younger-Kasami, zásobníkové automaty

Jan Křetínský

Fakulta informatiky, MU Brno

Jaro 2024

C-Y-K: Příklad

Problém: Lze v dané gramatice v CNF odvodit dané slovo w? Tj. $S \Rightarrow^* w$?

C-Y-K: Příklad

Problém: Lze v dané gramatice v CNF odvodit dané slovo w? Tj. $S \Rightarrow^* w$?

Řešení: Pro každé neprázdné podslovo u slova w spočítáme množinu T_u všech neterminálů, z kterých lze odvodit u.

C-Y-K: Příklad

Problém: Lze v dané gramatice v CNF odvodit dané slovo w? Tj. $S \Rightarrow^* w$?

Řešení: Pro každé neprázdné podslovo u slova w spočítáme množinu T_u všech neterminálů, z kterých lze odvodit u.

$$
\begin{aligned}
S &\rightarrow AB \mid SS \mid a \\
A &\rightarrow AA \mid BC \mid a \\
B &\rightarrow AB \mid b \\
C &\rightarrow SA \mid b
\end{aligned}
$$

$$T_{i,j} = \{X \in N \mid X \Rightarrow^* w_i w_{i+1} \dots w_{i+j-1}\}$$

$$w = abaa$$

	1	2	3	4
4				
3				
2				
1				
	a	b	a	a

Algoritmus Cocke-Younger-Kasami

Vstup: gramatika $\mathcal{G} = (N, \Sigma, P, S)$ v CNF, slovo $w = w_1 \dots w_n \neq \varepsilon$
Výstup: množiny $T_{i,j} = \{X \in N \mid X \Rightarrow^* w_i \dots w_{i+j-1}\}$

for $i \leftarrow 1$ **to** n **do**
  $T_{i,1} \leftarrow \emptyset$
  for all pravidlo tvaru $(A \to a) \in P$ **do**
    if $a = w_i$ **then** $T_{i,1} \leftarrow T_{i,1} \cup \{A\}$
for $j \leftarrow 2$ **to** n **do**
  for $i \leftarrow 1$ **to** $n - j + 1$ **do**
    $T_{i,j} \leftarrow \emptyset$
    for $k \leftarrow 1$ **to** $j - 1$ **do**
      for all pravidlo tvaru $(A \to BC) \in P$ **do**
        if $B \in T_{i,k} \ \wedge \ C \in T_{i+k,j-k}$ **then** $T_{i,j} \leftarrow T_{i,j} \cup \{A\}$

Zásobníkové automaty

[1] Zápis $\mathcal{P}_{Fin}(Q \times \Gamma^*)$ značí množinu všech konečných podmnožin množiny $Q \times \Gamma^*$.

Zásobníkové automaty

Definice 3.36.

Nedeterministický zásobníkový automat (PushDown Automaton, PDA) je sedmice $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

- Q je konečná množina, jejíž prvky nazýváme stavy,
- Σ je konečná množina, tzv. vstupní abeceda,
- Γ je konečná množina, tzv. zásobníková abeceda,
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_{Fin}(Q \times \Gamma^*)$, tzv. (parciální) přechodová funkce[1],
- $q_0 \in Q$ je počáteční stav,
- $Z_0 \in \Gamma$ je počáteční symbol v zásobníku,
- $F \subseteq Q$ je množina koncových stavů.

[1] Zápis $\mathcal{P}_{Fin}(Q \times \Gamma^*)$ značí množinu všech konečných podmnožin množiny $Q \times \Gamma^*$.

Výpočet zásobníkového automatu

Definice 3.37.

Nechť $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je PDA.

Konfigurací nazveme libovolný prvek $(p, w, \alpha) \in Q \times \Sigma^* \times \Gamma^*$.

Na množině všech konfigurací automatu $\mathcal{M}$ definujeme binární relaci krok výpočtu $\vdash_{\mathcal{M}}$ takto:

$$(p, aw, Z\alpha) \vdash_{\mathcal{M}} (q, w, \gamma\alpha) \overset{def}{\iff} \exists (q, \gamma) \in \delta(p, a, Z) \text{ pro } a \in \Sigma \cup \{\varepsilon\}$$

Reflexivní a tranzitivní uzávěr relace $\vdash_{\mathcal{M}}$ značíme $\vdash_{\mathcal{M}}^{*}$.

Je-li $\mathcal{M}$ zřejmý z kontextu, píšeme pouze $\vdash$ resp. $\vdash^{*}$.

Akceptující výpočet zásobníkového automatu

Definice 3.37. (pokračování)

Jazyk akceptovaný PDA $\mathcal{M}$ koncovým stavem definujeme jako

$$L(\mathcal{M}) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \overset{*}{\vdash} (q_f, \varepsilon, \alpha), \quad \text{kde} \quad q_f \in F, \alpha \in \Gamma^*\}$$

a jazyk akceptovaný PDA $\mathcal{M}$ prázdným zásobníkem definujeme jako

$$L_e(\mathcal{M}) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \overset{*}{\vdash} (q, \varepsilon, \varepsilon), \quad \text{kde} \quad q \in Q\}.$$

Příklad

$\mathcal{M} = (\{q_0, p, f\}, \{a, b\}, \{A, B, Z\}, \delta, q_0, Z, \{f\})$, kde

$$
\begin{aligned}
\delta(q_0, a, Z) &= \{(q_0, AZ)\} \\
\delta(q_0, a, A) &= \{(q_0, AA)\} \\
\delta(q_0, a, B) &= \{(q_0, AB)\} \\
\delta(q_0, b, Z) &= \{(q_0, BZ)\} \\
\delta(q_0, b, A) &= \{(q_0, BA)\} \\
\delta(q_0, b, B) &= \{(q_0, BB)\} \\
\delta(q_0, \varepsilon, Z) &= \{(p, Z)\} \\
\delta(q_0, \varepsilon, A) &= \{(p, A)\} \\
\delta(q_0, \varepsilon, B) &= \{(p, B)\} \\
\delta(p, a, A) &= \{(p, \varepsilon)\} \\
\delta(p, b, B) &= \{(p, \varepsilon)\} \\
\delta(p, \varepsilon, Z) &= \{(f, Z)\}
\end{aligned}
$$

Příklad

$\mathcal{M} = (\{q_0\}, \{a, b\}, \{Z, A\}, \delta, q_0, Z, \emptyset)$, kde

$$\begin{aligned}
\delta(q_0, a, Z) &= \{(q_0, A)\} \\
\delta(q_0, a, A) &= \{(q_0, AA)\} \\
\delta(q_0, b, A) &= \{(q_0, \varepsilon)\}
\end{aligned}$$

Ekvivalence dvou způsobů akceptování

Věta 3.39.

Pro každý jazyk L platí:
$L = L(\mathcal{N})$ pro nějaký PDA $\mathcal{N}$ $\iff$ $L = L_e(\mathcal{M})$ pro nějaký PDA $\mathcal{M}$.

Důkaz. **1. koncový stav $\Longrightarrow$ prázdný zásobník**

Ekvivalence dvou způsobů akceptování

Věta 3.39.

Pro každý jazyk L platí:
$L = L(\mathcal{N})$ pro nějaký PDA $\mathcal{N}$ $\iff$ $L = L_e(\mathcal{M})$ pro nějaký PDA $\mathcal{M}$.

Důkaz. 1. koncový stav $\Longrightarrow$ prázdný zásobník

K danému $\mathcal{N}$ zkonstruujeme $\mathcal{M}$ simulující jeho činnost.
Vejde-li $\mathcal{N}$ do koncového stavu, $\mathcal{M}$ se nedeterministicky rozhodne

- pokračovat v simulaci automatu $\mathcal{N}$ **nebo**
- přejít do nově přidaného stavu q_ε, v němž vyprázdní zásobník.

Ekvivalence dvou způsobů akceptování

Věta 3.39.

Pro každý jazyk L platí:
$L = L(\mathcal{N})$ pro nějaký PDA $\mathcal{N}$ $\iff$ $L = L_e(\mathcal{M})$ pro nějaký PDA $\mathcal{M}$.

Důkaz. 1. koncový stav $\Longrightarrow$ prázdný zásobník

K danému $\mathcal{N}$ zkonstruujeme $\mathcal{M}$ simulující jeho činnost.
Vejde-li $\mathcal{N}$ do koncového stavu, $\mathcal{M}$ se nedeterministicky rozhodne

- pokračovat v simulaci automatu $\mathcal{N}$ **nebo**
- přejít do nově přidaného stavu q_ε, v němž vyprázdní zásobník.

Komplikace:

Ekvivalence dvou způsobů akceptování

Věta 3.39.

Pro každý jazyk L platí:
$L = L(\mathcal{N})$ pro nějaký PDA $\mathcal{N}$ $\iff$ $L = L_e(\mathcal{M})$ pro nějaký PDA $\mathcal{M}$.

Důkaz. 1. koncový stav $\Longrightarrow$ prázdný zásobník

K danému $\mathcal{N}$ zkonstruujeme $\mathcal{M}$ simulující jeho činnost.
Vejde-li $\mathcal{N}$ do koncového stavu, $\mathcal{M}$ se nedeterministicky rozhodne

- pokračovat v simulaci automatu $\mathcal{N}$ **nebo**
- přejít do nově přidaného stavu q_ε, v němž vyprázdní zásobník.

Komplikace: $\mathcal{N}$ vyprázdní zásobník bez akceptování

Řešení:

Ekvivalence dvou způsobů akceptování

Věta 3.39.

Pro každý jazyk L platí:
$L = L(\mathcal{N})$ pro nějaký PDA $\mathcal{N}$ $\iff$ $L = L_e(\mathcal{M})$ pro nějaký PDA $\mathcal{M}$.

Důkaz. 1. koncový stav $\Longrightarrow$ prázdný zásobník

K danému $\mathcal{N}$ zkonstruujeme $\mathcal{M}$ simulující jeho činnost.
Vejde-li $\mathcal{N}$ do koncového stavu, $\mathcal{M}$ se nedeterministicky rozhodne

- pokračovat v simulaci automatu $\mathcal{N}$ **nebo**
- přejít do nově přidaného stavu q_ε, v němž vyprázdní zásobník.

Komplikace: $\mathcal{N}$ vyprázdní zásobník bez akceptování

Řešení: Před zahájením simulace bude u $\mathcal{M}$ na dně zásobníku nový symbol, který nedovolíme odstranit jinde, než ve stavu q_ε.

Konstrukce: Nechť $\mathcal{N} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$.
Klademe $\mathcal{M} = (Q \cup \{q_0', q_\varepsilon\}, \Sigma, \Gamma \cup \{Z'\}, \delta', q_0', Z', \emptyset)$,
kde $Z' \notin \Gamma$, $q_0', q_\varepsilon \notin Q$ a δ' je definována takto:

- $\delta'(q_0', \varepsilon, Z') = \{(q_0, Z_0 Z')\}$
- jestliže $\delta(q, a, Z)$ obsahuje (r, γ), pak $\delta'(q, a, Z)$ obsahuje (r, γ)
- $\delta'(q, \varepsilon, Z)$ obsahuje (q_ε, Z)
  pro všechny $q \in F$ a $Z \in \Gamma \cup \{Z'\}$
- $\delta'(q_\varepsilon, \varepsilon, Z) = \{(q_\varepsilon, \varepsilon)\}$
  pro všechny $Z \in \Gamma \cup \{Z'\}$

2. prázdný zásobník $\Longrightarrow$ koncový stav

2. prázdný zásobník $\Longrightarrow$ koncový stav

K danému $\mathcal{M}$ zkonstruujeme $\mathcal{N}$ simulující jeho činnost.

- $\mathcal{N}$ si před simulací přidá na dno zásobníku nový symbol.
- Je-li $\mathcal{N}$ schopen číst tento symbol (tj. zásobník automatu $\mathcal{M}$ je prázdný) tak $\mathcal{N}$ přejde do nově přidaného stavu q_f, který je koncovým stavem.

Konstrukce: Nechť $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$.
Klademe $\mathcal{N} = (Q \cup \{q_0', q_f\}, \Sigma, \Gamma \cup \{Z'\}, \delta', q_0', Z', \{q_f\})$,
kde $Z' \notin \Gamma$, $q_0', q_f \notin Q$ a δ' je definována takto:

- $\delta'(q_0', \varepsilon, Z') = \{(q_0, Z_0 Z')\}$

- jestliže $\delta(q, a, Z)$ obsahuje (r, γ), pak $\delta'(q, a, Z)$ obsahuje (r, γ)
- $\delta'(q, \varepsilon, Z') = \{(q_f, \varepsilon)\}$
  pro všechny $q \in Q$ $\square$

Rozšířený zásobníkový automat

Definice 3.44.

Rozšířený zásobníkový automat je sedmice $\mathcal{R} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

- všechny symboly až na δ mají tentýž význam jako v definici PDA,
- δ je zobrazením
  z konečné podmnožiny množiny $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*$
  do konečných podmnožin množiny $Q \times \Gamma^*$.

Pojmy konfigurace a akceptovaný jazyk (koncovým stavem, prázdným zásobníkem) zůstávají beze změny.

Krok výpočtu $\vdash_{\mathcal{R}}$ definujeme takto:

$$(p, aw, \gamma_1\alpha) \vdash_{\mathcal{R}} (q, w, \gamma_2\alpha) \overset{def}{\iff} \exists(q, \gamma_2) \in \delta(p, a, \gamma_1) \text{ pro } a \in \Sigma \cup \{\varepsilon\}$$

Ekvivalence rozšířených PDA a PDA

Lemma 3.45.

Ke každému rozšířenému PDA existuje ekvivalentní (*obyčejný*) PDA.

Důkaz. Nechť $\mathcal{R} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je rozšířený PDA a
$m = \max\{|\alpha| \mid \delta(q, a, \alpha)$ je definováno pro nějaké $q \in Q, a \in \Sigma \cup \{\varepsilon\}\}$.

Ekvivalence rozšířených PDA a PDA

Lemma 3.45.

Ke každému rozšířenému PDA existuje ekvivalentní (*obyčejný*) PDA.

Důkaz. Nechť $\mathcal{R} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je rozšířený PDA a
$m = \max\{|\alpha| \mid \delta(q, a, \alpha) \text{ je definováno pro nějaké } q \in Q, a \in \Sigma \cup \{\varepsilon\}\}$.

Definujeme $\mathcal{P} = (Q_1, \Sigma, \Gamma_1, \delta_1, q_1, Z_1, F_1)$, kde

- $Q_1 = \{[q, \alpha] \mid q \in Q, \alpha \in \Gamma_1^*, 0 \leq |\alpha| \leq m\}$,
- $\Gamma_1 = \Gamma \cup \{Z_1\}$, kde Z_1 je nový symbol,
- $q_1 = [q_0, Z_0 Z_1^{m-1}]$,
- $F_1 = \{[q, \alpha] \mid q \in F, \alpha \in \Gamma_1^*, 0 \leq |\alpha| \leq m\}$.

- δ_1 je definována takto:
  - jestliže $\delta(q, a, X_1 \ldots X_k)$ obsahuje $(r, Y_1 \ldots Y_l)$, pak

    $l < k$: $\delta_1([q, X_1 \ldots X_k \alpha], a, Z)$ obsahuje $([r, Y_1 \ldots Y_l \alpha Z], \varepsilon)$
    pro všechny $Z \in \Gamma_1$ a $\alpha \in \Gamma_1^*$ takové, že $|\alpha| = m - k$

- δ_1 je definována takto:
  - jestliže $\delta(q, a, X_1 \dots X_k)$ obsahuje $(r, Y_1 \dots Y_l)$, pak

    $l < k$: $\delta_1([q, X_1 \dots X_k\alpha], a, Z)$ obsahuje $([r, Y_1 \dots Y_l\alpha Z], \varepsilon)$
    pro všechny $Z \in \Gamma_1$ a $\alpha \in \Gamma_1^*$ takové, že $|\alpha| = m - k$

    $l \geq k$: $\delta_1([q, X_1 \dots X_k\alpha], a, Z)$ obsahuje $([r, \beta], \gamma Z)$,
    kde $\beta\gamma = Y_1 \dots Y_l\alpha$ a $|\beta| = m$,
    pro všechny $Z \in \Gamma_1$ a $\alpha \in \Gamma_1^*$ takové, že $|\alpha| = m - k$

- δ_1 je definována takto:
  - jestliže $\delta(q, a, X_1 \ldots X_k)$ obsahuje $(r, Y_1 \ldots Y_l)$, pak

    $l < k$: $\delta_1([q, X_1 \ldots X_k\alpha], a, Z)$ obsahuje $([r, Y_1 \ldots Y_l\alpha Z], \varepsilon)$
    pro všechny $Z \in \Gamma_1$ a $\alpha \in \Gamma_1^*$ takové, že $|\alpha| = m - k$
    $l \geq k$: $\delta_1([q, X_1 \ldots X_k\alpha], a, Z)$ obsahuje $([r, \beta], \gamma Z)$,
    kde $\beta\gamma = Y_1 \ldots Y_l\alpha$ a $|\beta| = m$,
    pro všechny $Z \in \Gamma_1$ a $\alpha \in \Gamma_1^*$ takové, že $|\alpha| = m - k$
  - $\delta_1([q, \alpha], \varepsilon, Z) = \{([q, \alpha Z], \varepsilon)\}$
    pro všechny $q \in Q$, $Z \in \Gamma_1$ a $\alpha \in \Gamma_1^*$ takové, že $|\alpha| < m$

Korektnost: Ověříme, že platí

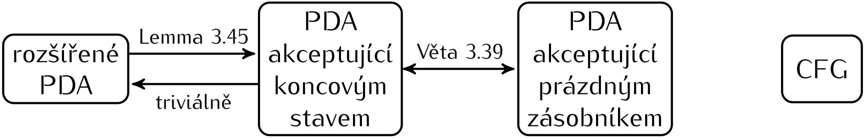
$$(q, aw, X_1 \dots X_k X_{k+1} \dots X_n) \vdash_{\mathcal{R}} (r, w, Y_1 \dots Y_l X_{k+1} \dots X_n)$$

$$\iff \quad ([q, \alpha], aw, \beta) \vdash_{\mathcal{P}}^{+} ([r, \alpha'], w, \beta'),$$

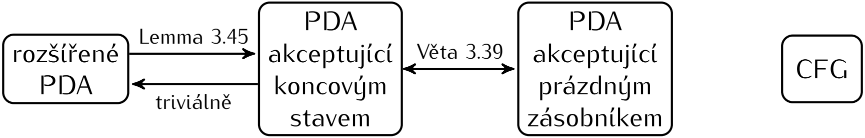
kde

1. $\alpha\beta = X_1 \dots X_n Z_1^m$,
2. $\alpha'\beta' = Y_1 \dots Y_l X_{k+1} \dots X_n Z_1^m$,
3. $|\alpha| = |\alpha'| = m$ a
4. mezi dvěma výše uvedenými konfiguracemi PDA $\mathcal{P}$ neexistuje taková konfigurace, kde druhá komponenta stavu (tj. buffer) by měla délku m. $\square$

Zásobníkové automaty a bezkontextové jazyky



Zásobníkové automaty a bezkontextové jazyky



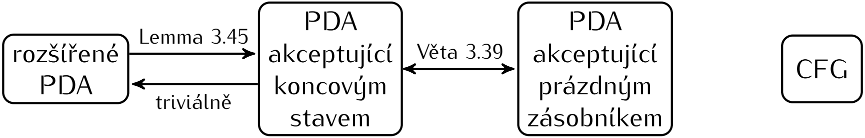
Ekvivalence CFG a PDA:

Věta 3.47.

Ke každé CFG $\mathcal{G}$ lze sestrojit PDA $\mathcal{M}$ takový, že $L(\mathcal{G}) = L_e(\mathcal{M})$.

Idea:

Zásobníkové automaty a bezkontextové jazyky



Ekvivalence CFG a PDA:

Věta 3.47.

Ke každé CFG $\mathcal{G}$ lze sestrojit PDA $\mathcal{M}$ takový, že $L(\mathcal{G}) = L_e(\mathcal{M})$.

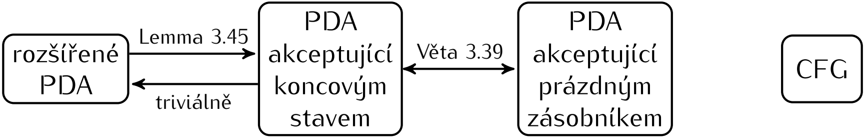
Idea: větná forma $wN\alpha$ odpovídá přečtení w a zásobníku $N\alpha$ jednostavového PDA

Věta 3.51.

Ke každému PDA $\mathcal{M}$ lze sestrojit CFG $\mathcal{G}$ takovou, že $L_e(\mathcal{M}) = L(\mathcal{G})$.

Idea:

Zásobníkové automaty a bezkontextové jazyky



Ekvivalence CFG a PDA:

Věta 3.47.

Ke každé CFG $\mathcal{G}$ lze sestrojit PDA $\mathcal{M}$ takový, že $L(\mathcal{G}) = L_e(\mathcal{M})$.

Idea: větná forma $wN\alpha$ odpovídá přečtení w a zásobníku $N\alpha$ jednostavového PDA

Věta 3.51.

Ke každému PDA $\mathcal{M}$ lze sestrojit CFG $\mathcal{G}$ takovou, že $L_e(\mathcal{M}) = L(\mathcal{G})$.

Idea: naopak + redukce vícestavového PDA na jednostavový