

Bezkontextové gramatiky s generovaním reverzov

Alena Poláčková
školiťel': RNDr. Peter Kostolányi PhD.

19. februára 2020

Obsah

Definície

Bezkontextové gramatiky s generovaním reverzov
Zásobníkové automaty s otáčacím zásobníkom

Parsovanie

Algoritmus posuň-redukuj-otoč
Ekvivalentné jednoznačné odvodenie

Bezkontextové gramatiky s generovaním reverzov

Bezkontextová gramatika je štvorica $G = (N, T, P, \sigma)$, kde

- ▶ N je množina neterminálov
- ▶ T je množina terminálov
- ▶ P je množina pravidiel tvaru $N \times (N \cup P)^*$
- ▶ σ je počiatočný neterminál

Bezkontextová gramatika s generovaním reverzov (RGCFG) je päťica $G = (N, T, P, \sigma, \textcircled{R})$, kde

- ▶ symboly N, T, P, σ definujeme ako v štandardnej bezkontextovej gramatike
- ▶ $\textcircled{R} \in T$ je špeciálny terminálny symbol

Trieda jazykov generovaná RGCFG

Gramatika obsahuje špeciálny symbol \textcircled{R} , ktorý sa na konci generovania vyhodnotí zľava doprava funkciou ϱ .

$$\varrho(w) = \begin{cases} w & w \in (T - \{\textcircled{R}\})^*, \\ u\varrho(v^R) & w = u\textcircled{R}v, \text{ kde } u \in (T - \{\textcircled{R}\})^* \text{ a } v \in T^*. \end{cases}$$

Príklad

Gramatika štandardným spôsobom vygeneruje slovo

$w = u\textcircled{R}v\textcircled{R}x\textcircled{R}y\textcircled{R}z$. Vyhodnotenie prebieha nasledovne:

$$\begin{aligned} \varrho(w) &= \varrho(u\textcircled{R}v\textcircled{R}x\textcircled{R}y\textcircled{R}z) = u\varrho(z^R\textcircled{R}y^R\textcircled{R}x^R\textcircled{R}v^R) = \\ &uz^R\varrho(v\textcircled{R}x\textcircled{R}y\textcircled{R}z) = uz^Rv\varrho(y^R\textcircled{R}x^R) = uz^Rvy^R\varrho(x) = uz^Rvy^Rx \end{aligned}$$

Trieda jazykov generovaná RGCFG (2)

Nech $G = (N, T, P, \sigma, \textcircled{R})$ je bezkontextová gramatika s generovaním reverzov. Potom pre jazyk generovaný gramatikou G platí

$$\begin{aligned} L(G) = & \{w_1 w_{2k}^R w_2 w_{2k-1}^R \dots w_k w_{k+1}^R \\ & \mid k \in N; w_1 \textcircled{R} w_2 \textcircled{R} \dots \textcircled{R} w_{2k} \in L_{CF}(G)\} \cup \\ & \cup \{w_1 w_{2k+1}^R w_2 w_{2k}^R \dots w_k w_{k+2}^R w_{k+1} \\ & \mid k \in N; w_1 \textcircled{R} w_2 \textcircled{R} \dots \textcircled{R} w_{2k+1} \in L_{CF}(G)\}. \end{aligned}$$

Táto trieda jazykov je silnejšia ako \mathcal{L}_{CF} a obsahuje napríklad slová tvaru ww .

Stredný reverz

- ▶ Pre počet n reverzov v slove, *stredný reverz* je ten v poradí $s := \lceil \frac{n+1}{2} \rceil$
- ▶ Príklad: $\varrho(u\textcircled{R}v\textcircled{R}x\textcircled{R}y\textcircled{R}z) = uz^Rvy^Rx$
- ▶ Normálny tvar so znalosťou generovaných reverzov, ak pre každý neterminál vieme povedať, ktoré reverzy generuje.
- ▶ Vďaka normálnemu tvaru vieme identifikovať neterminály generujúce stredný reverz

Zásobníkové automaty s otáčacím zásobníkom

Zásobníkový automat je sedmica $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

- ▶ K je množina stavov automatu
- ▶ Σ je vstupná abeceda
- ▶ Γ je zásobníková abeceda
- ▶ $\delta : K \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow 2_{fin}^{K \times \Gamma^*}$ je prechodová funkcia
- ▶ q_0 je počiatkový stav
- ▶ Z_0 je počiatkový znak na zásobníku
- ▶ $F \subseteq K$ je množina akceptačných stavov

Zásobníkový automat s otáčacím zásobníkom (NFPDA) je osmica $A = (K, \Sigma, \Gamma, \delta, \Delta, q_0, Z_0, F)$, kde

- ▶ symboly $K, \Sigma, \Gamma, \delta, q_0, Z_0, F$ definujeme štandardne
- ▶ $\Delta : K \rightarrow 2^K$ je špeciálna prechodová funkcia - predstavuje otočenie zásobníka

Krok výpočtu je použitie jednej z dvoch prechodových funkcií δ, Δ .

Ekvivalencia gramatík a automatov

$\mathcal{L}(RGCFG_k)$ je trieda všetkých jazykov L takých, že $L = L(G)$ pre nejakú RGCFG G generujúcu najviac k reverzov.

$\mathcal{L}(NFPDA_k)$ je trieda všetkých jazykov L takých, že $L = L(A)$ pre nejaký nedeterministický automat s otáčacím zásobníkom A , pracujúci s najviac k otočeniami zásobníka.

Nech $k \in \mathbb{N}$. Potom $\mathcal{L}(NFPDA_k) = \mathcal{L}(RGCFG_k)$.

Nech $\mathcal{L}(NFPDA)$ je trieda jazykov akceptovaných nejakým automatom NFPDA s neobmedzeným počtom prevrátení. Potom $\mathcal{L}(NFPDA) = \mathcal{L}(RE)$. (Sú rovnako silné ako Turingove stroje.)

Cieľ práce

Preskúmať vybrané vlastnosti bezkontextových gramatík s generovaním reverzov. Zamerať sa predovšetkým na efektívne metódy syntaktickej analýzy.

Efektívne parsovanie LR

- ▶ algoritmus *posuň-redukuj* pre parsovanie bezkontextových gramatík
- ▶ vieme, že gramatiky s generovaním reverzov sú rovnako silné ako automaty s otáčacím zásobníkom
- ▶ chceli by sme vedieť efektívne parsovať slová z takýchto gramatík na automate s otáčacím zásobníkom - dá sa to?

Algoritmus posuň-redukuj

- ▶ $P = \{\sigma \rightarrow \alpha\beta, \alpha \rightarrow x, \beta \rightarrow yz\}$
- ▶ $\sigma \Rightarrow \alpha\beta \Rightarrow x\beta \Rightarrow xyz$

zásobník	vstup	operácia
⊢	xyz	posuň
⊢ x	yz	redukuj
⊢ α	yz	posuň
⊢ αy	z	posuň
⊢ $\alpha y z$		redukuj
⊢ $\alpha\beta$		redukuj
⊢ σ		akceptuj

Algoritmus *posuň-redukuj-otoč*

- ▶ Zoberme si gramatiku $G = (N, T, P, \sigma, \textcircled{R})$, kde
 $N = \{\sigma, \alpha, \beta\}$, $T = \{x, y, z, \textcircled{R}\}$,
 $P = \{\sigma \rightarrow \alpha\beta, \alpha \rightarrow x\textcircled{R}, \beta \rightarrow y\textcircled{R}z\}$.
- ▶ Odvodenie slova: $\sigma \Rightarrow^* x\textcircled{R}y\textcircled{R}z$.
 $\rho(x\textcircled{R}y\textcircled{R}z) = x\rho(z^R\textcircled{R}y^R) = xz^R\rho(y) = xz^Ry$.
- ▶ Pridáme operáciu na zásobníku **otoč** do štandardného algoritmu *posuň-redukuj* a skúsime parsovať vygenerované slovo w .

Algoritmus *posuň-redukuj-otoč* (2)

zásobník	vstup	operácia
$\vdash \parallel$	$xz^R y$	posuň
$\vdash \parallel x$	$z^R y$	otoč
$\vdash \textcircled{R} x^R \parallel$	$z^R y$	posuň
$\vdash \textcircled{R} x^R \parallel z^R$	y	otoč
$\vdash \textcircled{R} z \parallel x \textcircled{R}$	y	redukuj podľa $\alpha \rightarrow x \textcircled{R}$
$\vdash \textcircled{R} z \parallel \alpha$	y	posuň
$\vdash \textcircled{R} z \parallel \alpha y$		redukuj podľa $\beta \rightarrow y \textcircled{R} z$ (vyber y , otoč, vyber $\textcircled{R} z$)
$\vdash \alpha \parallel \beta$		redukuj podľa $\sigma \rightarrow \alpha \beta$ (vyber β , otoč, vyber α)
$\vdash \parallel \sigma$		akceptuj

Ekvivalentné jednoznačné odvodenie

- ▶ Algoritmus od konca simuluje odvodenie vstupného slova v gramatike (na základe redukovaných pravidiel).
- ▶ Toto konkrétne odvodenie vieme popísať vďaka normálnemu tvaru so znalosťou generovaných reverzov.
- ▶ Vo vetnej forme prepisujeme najprv neterminály generujúce stredný reverz.
- ▶ Následne, podľa parity počtu generovaných reverzov definujeme postupnosť, podľa ktorej sa budú generovať ostatné reverzy v slove. Pre párný počet reverzov, kde s označuje index stredného reverzu, generujeme reverzy v poradí $s, s - 1, s + 1, \dots, n, 1$. Pre nepárny počet reverzov je postupnosť $s, s + 1, s - 1, \dots, n, 1$.

Čo ďalej?

- ▶ Dôkaz správnosti algoritmu cez jednoznačné odvodenie
- ▶ Analýza zložitosti algoritmu
- ▶ Deterministický algoritmus

Ďakujem za pozornosť