Intuice pro analýzu zdola nahoru

Intuice pro analýzu zdola nahoru

$$
\begin{aligned}
S &\rightarrow XY \\
X &\rightarrow ab \\
Y &\rightarrow c
\end{aligned}
$$

Nedeterministická syntaktická analýza zdola nahoru

Věta 3.55. Nechť $\mathcal{G}$ je libovolná CFG, pak lze zkonstruovat rozšířený PDA $\mathcal{R}$ takový, že $L(\mathcal{G}) = L(\mathcal{R})$.

Důkaz. Vrchol zásobníku píšeme vpravo.
Konstruujeme rozšířený PDA $\mathcal{R}$, který simuluje pravou derivaci v $\mathcal{G}$ v obráceném pořadí.

PDA $\mathcal{R}$ má kroky dvojího typu:

1. může kdykoli načíst do zásobníku symbol ze vstupu
2. **(redukce)** je-li na vrcholu zásobníku řetězec tvořící pravou stranu nějakého pravidla v $\mathcal{G}$, může ho nahradit odpovídajícím levostranným neterminálem (a ze vstupu nic nečte)

Nechť $\mathcal{G} = (N, \Sigma, P, S)$.
Položme $\mathcal{R} = (\{q, r\}, \Sigma, N \cup \Sigma \cup \{\perp\}, \delta, q, \perp, \{r\})$, kde $\perp$ je nově přidaný symbol a kde δ je definována takto:

1. $\delta(q, a, \varepsilon) = \{(q, a)\}$ pro všechna $a \in \Sigma$
2. je-li $A \to \alpha$ pravidlo v P, pak $\delta(q, \varepsilon, \alpha)$ obsahuje (q, A)
3. $\delta(q, \varepsilon, \perp S) = \{(r, \varepsilon)\}$

	krok výpočtu	odpovídající pravidlo z $\mathcal{G}$
$(q,\ i+i*i,\ \bot)$	$\vdash^{i} (q, +i*i,\ \bot i)$	$F \to i$
	$\vdash^{\varepsilon} (q, +i*i,\ \bot F)$	$T \to F$
	$\vdash^{\varepsilon} (q, +i*i,\ \bot T)$	$E \to T$
	$\vdash^{\varepsilon} (q, +i*i,\ \bot E)$	
	$\vdash^{+} (q,\ i*i,\ \bot E+)$	
	$\vdash^{i} (q,\ *i,\ \bot E+i)$	$F \to i$
	$\vdash^{\varepsilon} (q,\ *i,\ \bot E+F)$	$T \to F$
	$\vdash^{\varepsilon} (q,\ *i,\ \bot E+T)$	
	$\vdash^{*} (q,\ i,\ \bot E+T*)$	
	$\vdash^{i} (q,\ \varepsilon,\ \bot E+T*i)$	$F \to i$
	$\vdash^{\varepsilon} (q,\ \varepsilon,\ \bot E+T*F)$	$T \to T*F$
	$\vdash^{\varepsilon} (q,\ \varepsilon,\ \bot E+T)$	$E \to E+T$
	$\vdash^{\varepsilon} (q,\ \varepsilon,\ \bot E)$	
	$\vdash^{\varepsilon} (r,\ \varepsilon,\ \varepsilon)$	

Korektnost

$$S \Rightarrow^* \alpha A y \overset{n}{\Rightarrow} xy \iff (q, xy, \perp) \vdash^* (q, y, \perp\alpha A),$$

kde $S \Rightarrow^* \alpha A y \overset{n}{\Rightarrow} xy$ je pravá derivace a A je nejpravější neterminál.

($\Longrightarrow$) indukcí k délce odvození
($\Longleftarrow$) indukcí k délce výpočtu

Pro $A = S$ a $\alpha, y = \varepsilon$ dostáváme:

$$S \Rightarrow^* x \iff (q, x, \perp) \vdash^* (q, \varepsilon, \perp S) \quad \Big[\vdash (r, \varepsilon) \Big]$$

"Výstupem"je pravá derivace v obráceném pořadí. $\square$

Efektivnost syntaktické analýzy

Nederministický PDA $\Longrightarrow$ nedeterministický algoritmus
$\Longrightarrow$ exponenciální deterministický algoritmus

Řešení:

- deterministický algoritmus složitosti $\mathcal{O}(n^3)$, kde $n = |w|$ (algoritmus Cocke - Younger - Kasami)
- deterministické zásobníkové automaty a deterministické bezkontextové jazyky
- lineární algoritmy pro speciální třídy deterministických bezkontextových jazyků

Vlastnosti bezkontextových jazyků

Věta 3.58. (a 3.61.) Třída bezkontextových jazyků ($\mathcal{L}_2$) **je** uzavřena vzhledem k operacím:

1. sjednocení
2. zřetězení
3. iterace
4. pozitivní iterace
5. průnik s regulárním jazykem

Věta 3.60. Třída bezkontextových jazyků ($\mathcal{L}_2$) **není** uzavřena vzhledem k operacím:

1. průnik
2. doplněk

Sjednocení

L_1 je generován CFG $\mathcal{G}_1 = (N_1, \Sigma_1, P_1, S_1)$ a
L_2 je generován CFG $\mathcal{G}_2 = (N_2, \Sigma_2, P_2, S_2)$.

Bez újmy na obecnosti můžeme předpokládat $N_1 \cap N_2 = \emptyset$.

Definujeme $\mathcal{G} = (N_1 \cup N_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P, S)$, kde S je nový symbol a

$$P = P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}.$$

Každá derivace v $\mathcal{G}$ začne použitím buď $S \to S_1$ nebo $S \to S_2$. Podmínka $N_1 \cap N_2 = \emptyset$ zaručí, že při použití $S \to S_1$ (resp. $S \to S_2$) lze v dalším derivování používat jen pravidla z P_1 (resp. P_2).

Jazyk $L = L_1 \cup L_2$ je generován gramatikou $\mathcal{G}$.

Zřetězení

L_1 je generován CFG $\mathcal{G}_1 = (N_1, \Sigma_1, P_1, S_1)$ a
L_2 je generován CFG $\mathcal{G}_2 = (N_2, \Sigma_2, P_2, S_2)$.

Bez újmy na obecnosti můžeme předpokládat $N_1 \cap N_2 = \emptyset$.

Definujeme $\mathcal{G} = (N_1 \cup N_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P, S)$, kde S je nový symbol a

$$P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}.$$

Jazyk $L = L_1.L_2$ je generován gramatikou $\mathcal{G}$.

Iterace a pozitivní iterace

L_1 je generován CFG $\mathcal{G}_1 = (N_1, \Sigma_1, P_1, S_1)$.

Definujeme $\mathcal{G} = (N_1 \cup \{S\}, \Sigma_1, P, S)$, kde S je nový symbol a

$$P = P_1 \cup \{S \to SS_1 \mid \varepsilon\}.$$

Jazyk $L = L_1^*$ je generován gramatikou $\mathcal{G}$.

Definujeme $\mathcal{G} = (N_1 \cup \{S\}, \Sigma_1, P, S)$, kde S je nový symbol a

$$P = P_1 \cup \{S \to SS_1 \mid S_1\}.$$

Jazyk $L = L_1^+$ je generován gramatikou $\mathcal{G}$.

Korektnost konstrukce pro iteraci

Dokážeme $L(\mathcal{G}) = L_1^*$.

Průnik a doplněk

$L_1 = \{a^n b^n c^m \mid m, n \geq 1\}$ $\quad$ $L_2 = \{a^m b^n c^m \mid m, n \geq 1\}$

Oba tyto jazyky jsou CFL.

Kbyby $\mathcal{L}_2$ byla uzavřena vzhledem k operaci průniku, pak by i
$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 1\}$ musel být bezkontextový, což však není.

Neuzavřenost $\mathcal{L}_2$ vůči doplňku plyne z její uzavřenosti na sjednocení, neuzavřenosti na průnik a z De Morganových pravidel:

$$L_1 \cap L_2 = co{-}(co{-}L_1 \cup co{-}L_2),$$

tj., kdyby $\mathcal{L}_2$ byla uzavřena na doplněk, musela by být uzavřena i na průnik, což však není.

Průnik s regulárním jazykem

$L = L(\mathcal{P})$, kde $\mathcal{P}$ je PDA $\mathcal{P} = (Q_1, \Sigma, \Gamma, \delta_1, q_1, Z_0, F_1)$
$R = L(\mathcal{A})$, kde $\mathcal{A}$ je deterministický FA $\mathcal{A} = (Q_2, \Sigma, \delta_2, q_2, F_2)$

Sestrojíme PDA $\mathcal{P}'$ takový, že $L(\mathcal{P}') = L \cap R$.

$\mathcal{P}' = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

- $Q = Q_1 \times Q_2$
- $q_0 = \langle q_1, q_2 \rangle$
- $F = F_1 \times F_2$
- δ : pro každé $p \in Q_1$, $q \in Q_2$, $a \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ platí:

$$\delta(\langle p, q\rangle, a, Z) = \{(\langle p', q'\rangle, \gamma) \mid (p', \gamma) \in \delta_1(p, a, Z) \text{ a } \widehat{\delta_2}(q, a) = q'\}$$

Zřejmě platí $w \in L(\mathcal{P}') \iff w \in L(\mathcal{P}) \cap L(\mathcal{A})$.

Rozhodnutelné problémy pro bezkontextové jazyky

Problém příslušnosti
Existuje algoritmus, který pro libovolnou danou CFG $\mathcal{G}$ a slovo w rozhoduje, zda $w \in L(\mathcal{G})$ či nikoliv.

Problém prázdnosti
Existuje algoritmus, který pro libovolnou danou CFG $\mathcal{G}$ rozhoduje, zda $L(\mathcal{G}) = \emptyset$ či nikoliv.

Problém konečnosti
Existuje algoritmus, který pro libovolnou danou CFG $\mathcal{G}$ rozhoduje, zda $L(\mathcal{G})$ je konečný či nikoliv.

Konečnost

Věta 3.68. Ke každé CFG $\mathcal{G}$ lze sestrojit čísla m, n taková, že $L(\mathcal{G})$ je nekonečný právě když existuje slovo $z \in L(\mathcal{G})$ takové, že $m < |z| \leq n$.

Důkaz. Předpokládejme, že $\mathcal{G}$ je v CNF.

Nechť p, q jsou čísla s vlastnostmi popsanými v Lemmatu o vkládání. Položme $m = p$ a $n = p + q$.

$(\Longleftarrow)$ Jestliže $z \in L(\mathcal{G})$ je takové slovo, že $|z| > p$, pak existuje rozdělení $z = uvwxy$ splňující $vx \neq \varepsilon$ a $uv^iwx^iy \in L(\mathcal{G})$ pro všechna $i \geq 0$. Tedy jazyk $L(\mathcal{G})$ obsahuje nekonečně mnoho slov tvaru uv^iwx^iy, je tedy nekonečný.

($\Longrightarrow$) Nechť $L(\mathcal{G})$ je nekonečný. Pak obsahuje i nekonečně mnoho slov délky větší než p – tuto množinu slov označme M. Zvolme z M libovolné takové slovo z, které má minimální délku a ukažme, že musí platit $p < |z| \leq p + q$.

Kdyby $|z| > p + q$, pak (opět dle Pumping lemmatu pro CFL) lze z psát ve tvaru $z = uvwxy$, kde $vx \neq \varepsilon$, $|vwx| \leq q$ a $uv^iwx^iy \in L(\mathcal{G})$ pro všechna $i \geq 0$.

Pro $i = 0$ dostáváme, že $uwy \in L(\mathcal{G})$ a současně $|uwy| < |uvwxy|$.

Z nerovností $|uvwxy| > p + q$ a $|vwx| \leq q$ plyne, že $|uwy| > (p + q) - q = p$. Tedy $uwy \in M$, což je spor s volbou z jako slova z M s minimální délkou. Celkem tedy musí být $|z| \leq p + q$. $\square$

Vlastnost sebevložení

Definice 3.70. Nechť $\mathcal{G} = (N, \Sigma, P, S)$ je CFG. Řekneme, že $\mathcal{G}$ má **vlastnost sebevložení**, jestliže existují $A \in N$ a $u, v \in \Sigma^+$ taková, že $A \Rightarrow^+ uAv$.

CFL L má **vlastnost sebevložení**, jestliže každá bezkontextová gramatika, která jej generuje, má vlastnost sebevložení.

Věta 3.71. CFL L má vlastnost sebevložení, právě když L není regulární.

Důkaz. Viz skripta.

Nerozhodnutelné problémy pro bezkontextové jazyky

Problém regularity
Neexistuje algoritmus, který pro libovolnou danou CFG $\mathcal{G}$ rozhoduje, zda $L(\mathcal{G})$ je regulární či nikoliv.
(Tedy není rozhodnutelné, zda $L(\mathcal{G})$ má vlastnost sebevložení či nikoliv.)

Problém univerzality
Neexistuje algoritmus, který pro libovolnou danou CFG $\mathcal{G}$ rozhoduje, zda $L(\mathcal{G}) = \Sigma^*$ či nikoliv.

Problémy ekvivalence a inkluze také nejsou rozhodnutelné (plyne z nerozhodnutelnosti problému univerzality).

Deterministické zásobníkové automaty

Definice 3.72. Řekneme, že PDA $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je **deterministický** (DPDA), jestliže jsou splněny tyto podmínky:

1. pro všechna $q \in Q$ a $Z \in \Gamma$ platí:
   kdykoliv $\delta(q, \varepsilon, Z) \neq \emptyset$, pak $\delta(q, a, Z) = \emptyset$ pro všechna $a \in \Sigma$
2. pro žádné $q \in Q$, $Z \in \Gamma$ a $a \in \Sigma \cup \{\varepsilon\}$ neobsahuje $\delta(q, a, Z)$ více než jeden prvek

Řekneme, že L je **deterministický bezkontextový jazyk** (DCFL), právě když existuje DPDA $\mathcal{M}$ takový, že $L = L(\mathcal{M})$.

Vlastnosti deterministických bezkontextových jazyků

Věta 3.82. Třída DCFL **je** uzavřena na doplněk.

Intuice: DPDA má nad každým slovem právě jeden výpočet. Pro doplněk stačí zaměnit koncové a nekoncové stavy.

Komplikace 1: DPDA nemusí dočíst vstupní slovo do konce, protože se vyprázdní zásobník nebo přechod není definován.

Řešení:

Komplikace 2: DPDA nemusí dočíst vstupní slovo do konce, protože přestane číst vstup a neustále provádí ε-kroky pod kterými zásobník neomezeně roste.

Řešení: $s = |Q|, \quad t = |\Gamma|$
$r = \max\{|\gamma| \mid (p', \gamma) \in \delta(p, \varepsilon, Z),\ p, p' \in Q,\ Z \in \Gamma\}$

zásobník neomezeně roste při ε-krocích $\iff$ během posloupnosti ε-kroků jeho délka vzroste o více než $r \cdot s \cdot t$

Komplikace 3: DPDA nemusí dočíst vstupní slovo do konce, protože přestane číst vstup a neustále provádí ε-kroky pod kterými zásobník neroste neomezeně, tj. po jistém počtu kroků se jeho obsah opakuje.

Řešení: $s = |Q|, \quad t = |\Gamma|$

$$r = \max\{|\gamma| \mid (p', \gamma) \in \delta(p, \varepsilon, Z),\ p, p' \in Q,\ Z \in \Gamma\}$$

Komplikace 4: DPDA dočte slovo, ale pak pod ε-kroky prochází koncové i nekoncové stavy (tj. některá slova jsou akceptována původním DPDA i DPDA se zaměněnými koncovými stavy).

Řešení:

Průnik a sjednocení

Věta. Třída DCFL **není** uzavřena na průnik.

Důkaz. $L_1 = \{a^n b^n c^m \mid m, n \geq 1\}$ a $L_2 = \{a^m b^n c^m \mid m, n \geq 1\}$ jsou DCFL, ale $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 1\}$ není ani bezkontextový. $\square$

Věta. Třída DCFL **je** uzavřena na průnik s regulárním jazykem.

Věta. Třída DCFL **není** uzavřena na sjednocení.

Důkaz. Plyne z uzavřenosti na doplněk, neuzavřenosti na průnik a z De Morganových pravidel:

$$L_1 \cap L_2 = co{-}(co{-}L_1 \cup co{-}L_2)$$

(Z uzavřenosti na sjednocení by plynula uzavřenost na průnik.) $\square$

Vztah deterministických a nedeterministických CFL

Věta. Třída DCFL tvoří vlastní podtřídu třídy bezkontextových jazyků. Zejména existují bezkontextové jazyky, které nejsou DCFL.

Příklad. Jazyk $co\text{-}\{ww \mid w \in \{a,b\}^*\}$ je CFL, ale není DCFL.

Aplikace (deterministických) bezkontextových jazyků

- syntaxe programovacích jazyků je definována pomocí CFG (dobře uzávorkované výrazy, *if–then–else* konstrukty)
- DTD (Document Type definition) umožňuje definovat bezkontexové jazyky – využití ve značkovacích jazycích (HTML, XML,. . .)
- nástroje pro tvorbu parserů/překladačů využívají různé algoritmy pro lineární deterministickou syntaktickou analýzu:
  LALR(1) - Yacc, Bison, javacup
  LL(k) - JavaCC, ANTLR