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 ve skriptech obsahuje závažnou chybu. Kdo mi jako první pošle mail s popisem chyby, získá 1 tvrdý bod. **Deadline: 31. 12. 2009**

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.

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).

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:

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

(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–$\{ww \mid w \in \{a,b\}^*\}$ 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

Turingův stroj – syntaxe

Definice. Turingův stroj (TM) je $\mathcal{M} = (Q, \Sigma, \Gamma, \triangleright, \sqcup, \delta, q_0, q_{\mathrm{acc}}, q_{\mathrm{rej}})$, kde

- Q je konečná množina, jejíž prvky nazýváme **stavy**,
- Σ je konečná množina, tzv. **vstupní abeceda**,
- Γ je konečná množina, tzv. **pracovní abeceda**, $\Sigma \subseteq \Gamma$,
- $\triangleright \in \Gamma \smallsetminus \Sigma$ je **levá koncová značka**,
- $\sqcup \in \Gamma \smallsetminus \Sigma$ je symbol označující **prázdné políčko**,
- $\delta : \; (Q \smallsetminus \{q_{\mathrm{acc}}, q_{\mathrm{rej}}\}) \times \Gamma \;\to\; Q \times \Gamma \times \{L, R\}$ je **totální přechodová funkce**,
- $q_0 \in Q$ je **počáteční stav**,
- $q_{\mathrm{acc}} \in Q$ je **akceptující stav**,
- $q_{\mathrm{rej}} \in Q$ je **zamítající stav**.

Dále požadujeme, aby pro každé $q \in Q$ existoval $p \in Q$ takový, že $\delta(q, \rhd) = (p, \rhd, R)$ (tj. $\rhd$ nelze přepsat ani posunout hlavu za okraj pásky).

Označení. $$\sqcup^\omega = \sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\ldots$$

Definice. Konfigurace Turingova stroje je trojice $(q, z, n) \in Q \times \{y\sqcup^\omega \mid y \in \Gamma^*\} \times \mathbb{N}_0$, kde

- q je stav,
- $y\sqcup^\omega$ je obsah pásky,
- n značí pozici hlavy na pásce.

Počáteční konfigurace pro vstup $w \in \Sigma^*$ je trojice $(q_0, \rhd w\sqcup^\omega, 0)$.
Akceptující konfigurace je každá trojice tvaru (q_{acc}, z, n).
Zamítající konfigurace je každá trojice tvaru (q_{rej}, z, n).

Výpočet Turingova stroje

Označení. Pro libovolný nekonečný řetěz z nad Γ, z_n označuje n-tý symbol řetězu z (z_0 je nejlevější symbol řetězu z). Dále $s_b^n(z)$ označuje řetěz vzniklý ze z nahrazením z_n symbolem b.

Definice. Na množině všech konfigurací stroje $\mathcal{M}$ definujeme binární relaci $\underset{\mathcal{M}}{\vdash}$ (**krok výpočtu**) takto:

$$(p, z, n) \quad \underset{\mathcal{M}}{\vdash} \quad \begin{cases} (q, s_b^n(z), n+1) & \text{pro } \delta(p, z_n) = (q, b, R) \\ \\ (q, s_b^n(z), n-1) & \text{pro } \delta(p, z_n) = (q, b, L) \end{cases}$$

Výpočet TM $\mathcal{M}$ na vstupu w je maximální (konečná nebo nekonečná) posloupnost konfigurací $K_0, K_1, K_2, \ldots$, kde K_0 je počáteční konfigurace pro w a $K_i \vdash_{\mathcal{M}} K_{i+i}$ pro všechna $i \geq 0$.

Stroj $\mathcal{M}$ **akceptuje** slovo w právě když výpočet $\mathcal{M}$ na w je konečný a jeho poslední konfigurace je akceptující.

Stroj $\mathcal{M}$ **zamítá** slovo w právě když výpočet $\mathcal{M}$ na w je konečný a jeho poslední konfigurace je zamítající.

Stroj $\mathcal{M}$ pro vstup w **cyklí** právě když výpočet $\mathcal{M}$ na w je nekonečný.

Jazyk akceptovaný TM $\mathcal{M}$ definujeme jako

$$L(\mathcal{M}) = \{w \in \Sigma^* \mid \mathcal{M} \text{ akceptuje } w\}.$$

Ukázky Turingových strojů a vztah k jazykům typu 0

Simulátor TM: http://www.fi.muni.cz/∼xbarnat/tafj/turing/

Věta. Jazyk L je rekursivně spočetný (tj. generovaný gramatikou typu 0) $\iff$ L je akceptovaný nějakým Turingovým strojem.

Úplný Turingův stroj a rekursivní jazyky

Definice. Turingův stroj se vstupní abecedou Σ se nazývá **úplný**, pokud každé $w \in \Sigma^*$ buď akceptuje nebo zamítá.

Jazyk se nazývá **rekursivní**, pokud je akceptovaný nějakým úplným Turingovým strojem.

Rozhodnutelné a nerozhodnutelné problémy

Problém rozhodnout, zda dané w má vlastnost P ztotožníme s množinou $\{w \mid w \text{ má vlastnost } P\}$.

Příklad. Problém rozhodnout, zda daná regulární gramatika reprezentuje konečný jazyk $= \{\mathcal{G} \mid \mathcal{G} \text{ je regulární gr. a } L(\mathcal{G}) \text{ je konečný}\}$.

Definice. Problém P je

- **rozhodnutelný** $\iff$ $\{w \mid w \text{ má vlastnost } P\}$ je rekursivní
- **nerozhodnutelný** $\iff$ $\{w \mid w \text{ má vlastnost } P\}$ není rekursivní
- **částečně rozhodnutelný (semirozhodnutelný)** $\iff$
  $\iff$ $\{w \mid w \text{ má vlastnost } P\}$ je rekursivně spočetná

Přehled jazykových tříd

Jazyky	Gramatiky (typ)	Automaty
rekursivně spočetné	frázové (0)	Turingovy stroje
rekursivní	-	úplné Turingovy stroje
kontextové	kontextové (1)	lineárně ohraničené TM
bezkontextové	bezkontextové (2)	zásobníkové automaty
deterministické CFL	-	deterministické PDA
regulární	regulární (3)	konečné automaty

Třída na nižším rádku je vždy vlastní podtřídou třídy na vyšším řádku.