

Intuice k C-Y-K algoritmu

$S \rightarrow AB \mid CD \mid EF$

Platí $S \Rightarrow^* w$?

Intuice k C-Y-K algoritmu

Problém: Lze v dané gramatice v CNF vygenerovat dané slovo w ?

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

- $u = a$
- $u = ab$
- $u = abc$

Příklad

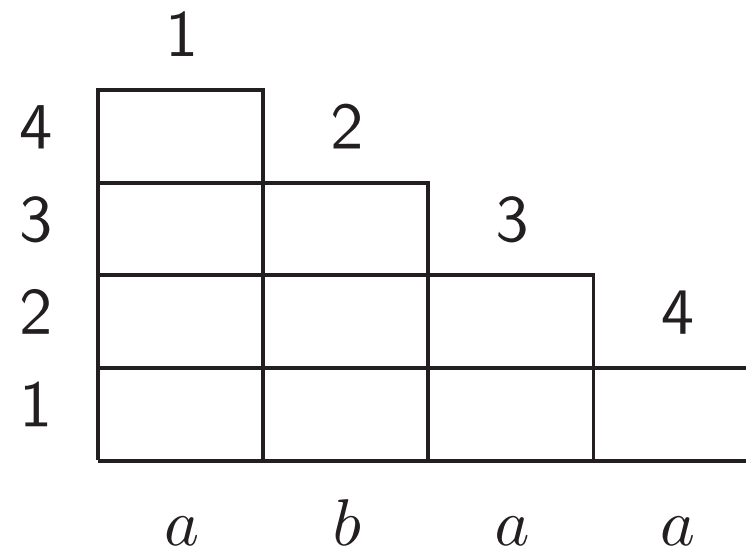
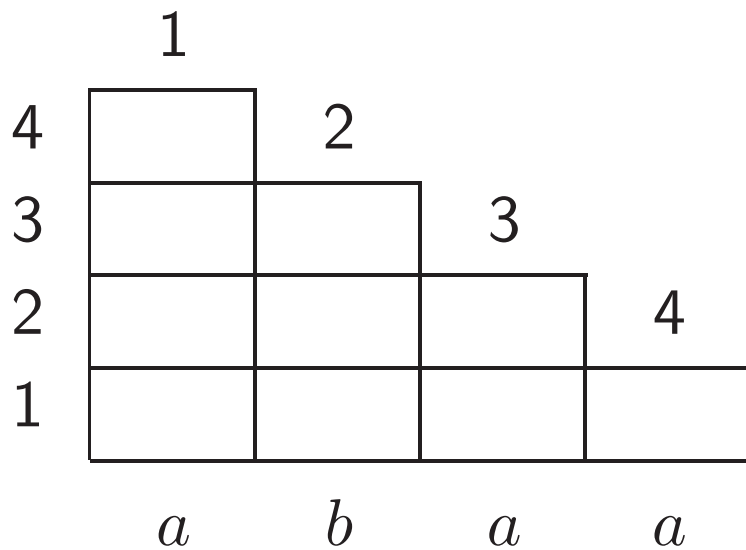
$$\begin{aligned} S &\rightarrow AB \mid SS \mid a \\ A &\rightarrow AA \mid BC \mid a \\ B &\rightarrow AB \mid b \\ C &\rightarrow SA \mid b \end{aligned}$$

platí $S \Rightarrow^* abaa$?

Příklad

$S \rightarrow AB \mid SS \mid a$
 $A \rightarrow AA \mid BC \mid a$
 $B \rightarrow AB \mid b$
 $C \rightarrow SA \mid b$

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



Algoritmus Cocke - Younger - Kasami

Vstup: gramatika $G = (N, \Sigma, P, S)$ v CNF, slovo $w = w_1 \dots w_n$

Poznámka: $T_{i,j} = \{X \mid X \rightarrow^* w_i \dots w_{i+j-1}\}$

```
1 for  $i := 1$  to  $n$  do
2    $T_{i,1} := \emptyset$ 
3   for každé pravidlo  $A \rightarrow a \in P$  do
4     if  $a = w_i$  then  $T_{i,1} := T_{i,1} \cup \{A\}$  fi
5   od od
6 for  $j := 2$  to  $n$  do
7   for  $i := 1$  to  $n - j + 1$  do
8      $T_{i,j} := \emptyset$ 
9     for  $k := 1$  to  $j - 1$  do
10      for každé pravidlo  $A \rightarrow BC \in P$  do
11        if  $B \in T_{i,k} \wedge C \in T_{i+k,j-k}$  then  $T_{i,j} := T_{i,j} \cup \{A\}$  fi
12      od od od od
```

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

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

Každá derivace v \mathcal{G} začne použitím buď $S \rightarrow S_1$ nebo $S \rightarrow S_2$. Podmínka $N_1 \cap N_2 = \emptyset$ zaručí, že při použití $S \rightarrow S_1$ (resp. $S \rightarrow S_2$) 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

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

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

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

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

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

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 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:

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

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:

$$\delta(\langle p, q \rangle, a, Z) = \{(\langle p', q' \rangle, \gamma) \mid (p', \gamma) \in \delta_1(p, a, Z) \text{ a } \hat{\delta}_2(q, a) = q'\}$$

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

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 prázdnoty

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 sestrojít 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 $n = p + q$.

(\Leftarrow) 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.

(\implies) 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. 2010**

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 sebevloženosti 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).