1 Rozhodnutelné problémy pro bezkontextové jazyky

Prostřednictvím tohoto dokumentu se budeme zabývat otázkami týkajícími se třídy bezkontextových jazyků a ptát se, zda jsme schopni algoritmicky rozhodnout, má-li jazyk $L(G)$ určitou vlastnost (G je libovolná CFG). Např. je-li konečný, patří-li do něj vybrané slovo, zda je prázdný atd.

Na spoustu otázek odpověď nalezneme, protože již daný algoritmus známe – můžeme tedy problém určení vlastnosti považovat za rozhodnutelný. Najdeme však i problémy, pro něž neznáme algoritmus k jejich řešení a které jsme schopni řešit pouze svou intuicí, ne však algoritmem. Takové problémy považujeme za nerozhodnutelné.

1. Problém příslušnosti do jazyka

Známe algoritmus, který pro dané slovo w a danou bezkontextovou gramatiku G dokáže rozhodnout, zda $w \in L(G)$.

Poznámka.

Jedná se buď o

1. nedeterministickou syntaktickou analýzu (shora dolů či zdola nahoru),
2. případně deterministickou syntaktickou analýzu pomocí C-Y-K algoritmu.

2. Problém prázdnosti jazyka

Existuje algoritmus, který pro danou bezkontextovou gramatiku G rozhodne, zda jazyk $L(G)$ je prázdný či nikoliv.

Poznámka.

I tento algoritmus známe, i když to možná není na první pohled patrné. Možná si vzpomínáte na postup pro převod libovolné bezkontextové gramatiky na redukovanou, která již neobsahuje nepoužitelné symboly 1. typu, čili nenormované symboly. Uváděli jsme si, jak tento typ neterminálů odstranit. Budovali jsme postupně množiny $N_0, N_1, \ldots, N_i$, do kterých jsme přidávali neterminály, které je možné přepsat na terminální řetěz v i krocích ($i \in \mathbb{N}_0$). Konstrukce skončila v případě, že se dvě množiny N_k a N_{k-1} nelišily. V tom případě jsme položili $N_e = N_k$ a neterminály patřící do N_e prohlásili za normované (použitelné 1. typu), ostatní naopak nikoliv.

Pokud se nyní zaměříme pouze na startovní neterminál S, tak v případě, že $S \in N_e$, je možné S odvodit na terminální řetěz. V tu chvíli však $L(G)$ není prázdný. Jestliže však neterminál S neobjevíme v množině N_e, pak můžeme $L(G)$ prohlásit za prázdný jazyk.

Příklad 1.

Buď dána bezkontextová gramatika $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, kde množina pravidel P je následující:

$$
\begin{array}{rcl}
S & \rightarrow & aAB \mid bC \\
A & \rightarrow & aBB \\
B & \rightarrow & SC \mid bb \\
C & \rightarrow & SaB
\end{array}
$$

Zjistěte, zda jazyk $L(G)$ je prázdný či nikoliv.

Řešení. Nejprve definujeme množinu $N_0 = \emptyset$.

1. Hledáme neterminály X gramatiky G, které lze přepsat v jednom kroku na terminální řetěz w, tj. takové symboly X, pro které v množině P existuje pravidlo $X \to w$, kde $w \in \Sigma^*$. Takovou vlastnost má pouze neterminál B, tj. položíme $N_1 = \{B\}$.
2. Do množiny N_2 přidáme takové neterminály X, pro které existuje pravidlo $X \to \alpha$, kde $\alpha \in (\Sigma \cup N_1)^*$. Zjišťujeme, že tuto podmínku splňuje pravidlo $A \to aBB$, kde $B \in N_1$. Do N_2 tedy vložíme symboly B, A.
3. Všimněte si pravidla $S \to aAB$, které obsahuje buď terminál a nebo neterminály patřící do N_3. Je tedy patrné, že S vložíme do N_3 společně s A, B.
4. Poslední krok je již pro nás nepodstatný, přesto i neterminál C lze přepsat na terminální řetěz, neboť pravidlo $C \to SaB$ obsahuje pouze terminály či neterminály patřící do N_3.

Množina $N_4 = \{S, A, B, C\}$, tedy z S lze vyderivovat terminální řetěz, tím pádem $L(G) \neq \emptyset$.

3. Problém konečnosti jazyka

Existuje algoritmus, který pro libovolnou bezkontextovou gramatiku G rozhoduje, zda $L(G)$ je konečný či nikoliv.

Věta 1 (aplikace Pumping lemmatu).

Ke každé bezkontextové gramatice G lze sestrojit čísla m, n taková, že $L(G)$ je nekonečný, právě když existuje slovo $z \in L(G)$ takové, že $m < |z| \leq n$.

Důkaz.

Je-li gramatika G bezkontextová, pak i jazyk $L(G)$ je bezkontextový. Předpokládejme (bez újmy na obecnosti), že G je v Chomského normální formě. Podle Pumping lemmatu existují čísla p, q taková, že libovolné slovo $z \in L(G)$, $|z| > p$, můžeme rozdělit na pět částí $uvwxy$, kde:

1. $vx \neq \varepsilon$,
2. $|vwx| \leq q$,
3. $uv^iwx^iy \in L(G)$ pro každé $i \in \mathbb{N}_0$.

Položme $m = p$, $n = p + q$.

$\Leftarrow$: Předpokládejme, že existuje slovo $z \in L(G)$ takové, že $m < |z| \leq n$. Nahradíme-li p za m, získáváme nerovnost $|z| > p$. Dle Pumping lemmatu pak platí, že $z = uvwxy$, kde $vx \neq \varepsilon, |vwx| \leq q$ a $uv^iwx^iy \in L(G)$ pro $\forall i \in \mathbb{N}_0$. Jazyk $L(G)$ obsahuje nekonečně mnoho slov uv^iwx^iy, čili je nekonečný.

$\Rightarrow$: Předpokládejme, že $L(G)$ je nekonečný. Určitě existuje nekonečně mnoho slov $z \in L(G)$, $|z| > p = m$. Označme $M = \{z \in L(G), |z| > p\}$. Vyberme slovo $z \in M$, které má **minimální** délku. Naším úkolem je ukázat, že délka slova z je ohraničená i shora, tj. $|z| \leq p + q$.

Předpokládejme sporem, že z má tu vlastnost, že $|z| > p+q$. Opět můžeme použít Pumping lemma: tedy z lze rozdělit na 5 částí $uvwxy$ tak, že $vx \neq \varepsilon$, $|vwx| \leq q$ a $uv^iwx^iy \in L(G)$ pro libovolné $i \in \mathbb{N}_0$.

Pro $i = 0$ dostáváme, že $uwy \in L(G)$. Protože $vx \neq \varepsilon$, platí, že $|uwy| < |uvwxy|$. Jelikož

$$|uvwxy| > p + q \ \wedge \ |vwx| \leq q,$$

dostáváme, že $|uwy| > p+q-q = p$. To však znamená, že slovo uwy patří do množiny M a má přitom prokazatelně menší délku než $z = uvwxy$. Docházíme ke sporu s předpokladem, že slovo $z \in M$ má minimální délku ze všech slov množiny M. Náš úvodní předpoklad, že $|z| > p + q$, je mylný a platí opak: $|z| \leq p + q$. Dáme-li vše dohromady:

$$m = p < |z| \leq p + q = n.$$

Definice – vlastnost sebevložení

Buď $G = (N, \Sigma, P, S)$ bezkontextová gramatika. Řekněme,že G má vlastnost sebevložení právě tehdy, když existuje $A \in N,\ u, v \in \Sigma^+$ takové, že $A \Rightarrow^* uAv$.

Jazyk L má vlastnost sebevložení, právě když každá gramatika, které jej generuje, má vlastnost sebevložení.

Příklad 2.

Mějme bezkontextovou gramatiku $G = (\{S, A, B\}, \{a, b\}, P, S)$, kde množina pravidel P je následující:

$$\begin{array}{lcl} S & \rightarrow & a \mid aA \\ A & \rightarrow & bA \mid Bb \mid b \\ B & \rightarrow & Aa \mid b \end{array}$$

Ukažte, zda gramatika G má vlastnost sebevložení.

Řešení. Uveďme následující derivaci v gramatice G:

$$A \Rightarrow bA \Rightarrow bBb \Rightarrow bAaab$$

Všimněte si, že $A \Rightarrow^+ bAaab$, z čehož vyplývá, že G má vlastnost sebevložení.

Příklad 3.

Je dána bezkontextová gramatika $G = (\{S\}, \{a, b\}, P, S)$, kde množina pravidel

$$P = \{S \rightarrow aS \mid Sb \mid a\}.$$

Zkuste promyslet odpovědi na následující otázky a zdůvodnit Váš názor.

1. Má gramatika G vlastnost sebevložení?
2. Určete jazyk $L(G)$.
3. Má jazyk $L(G)$ vlastnost sebevložení?
4. Je jazyk generovaný gramatikou G regulární?
5. Jaký je vztah mezi vlastností sebevložení a regularitou?

Poznámka.

Odpověď na otázku 5. podává i následující věta, jejíž důkaz nebudeme uvádět.

Věta 2 (vztah mezi regularitou a sebevložením).

Jazyk L není regulární právě tehdy, když má vlastnost sebevložení.

2 Nerozhodnutelné problémy pro bezkontextové jazyky

1. Problém regularity

Neexistuje algoritmus, který by pro libovolnou bezkontextovou gramatiku G rozhodl, zda je regulární či není.

Poznámka.

Je sice pravda, že jazyk $L(G)$ je regulární, právě když gramatika G nemá vlastnost sebevložení (dle Věty 2). Neexistuje však postup, jak určit, že daná gramatika má či nemá vlastnost sebevložení. Vše je opět záležitostí naší intuice, zda jsme schopni najít derivaci znamenající potvrzení vlastnosti sebevložení.

2. Problém univerzality

Neexistuje algoritmus, který by pro libovolnou bezkontextovou gramatiku $G = (N, \Sigma, P, S)$ rozhodl, zda $L(G) = \Sigma^*$.

3. Problém ekvivalence a inkluze

Neexistuje algoritmus, který by pro dvě libovolné bezkontextové gramatiky G_1, G_2 rozhodl, zda $L(G_1) = L(G_2)$, případně $L(G_1) \subseteq L(G_2)$ či naopak.

Poznámka.

Plyne právě z nerozhodnutelnosti problému univerzality.