IB102 – úkol 6, příklad 2 – řešení **Odevzdání: 4. 11. 2013**

Vypracoval(a): UČO:
Skupina:

2. [2 body] Nechť $\Sigma = \{a, b\}$. Napište algoritmus, který pro zadanou bezkontextovou gramatiku $G = (N, \Sigma, P, S)$ rozhodne, zda jazyk generovaný touto gramatikou obsahuje alespoň 1 slovo ve tvaru x^n pro nějaké $x \in \Sigma$ a $n > 0$.

Tedy algoritmus rozhodne, zda gramatika generuje alespoň jedno slovo délky alespoň 1 skládající se pouze z a-ček, nebo pouze z b-ček.

Popište princip fungování vašeho algoritmu a dokažte, že je tento algoritmus konvergentní (vždy skončí).

Můžete využívat libovolné algoritmy z přednášky, musíte na to však upozornit v textu.

Pro řešení tohoto problému nejprve využijeme toho, že pro každou bezkontextovou gramatiku G existuje jazykově ekvivalentní gramatika $\widehat{G}$, která je v Chomského normální formě.

Algoritmus tedy bude probíhat ve 2 fázích:

1. Transformuj G na $\widehat{G} = (\widehat{N}, \Sigma, \widehat{P}, \widehat{S})$ takovou, že $\mathcal{L}(G) = \mathcal{L}(\widehat{G})$ a $\widehat{G}$ je v Chomského normální formě (algoritmus v 6. přednášce).
2. Rozhodni, zda $\widehat{G}$ generuje slovo ve tvaru a^k nebo b^k pro nějaké $k > 0$.

Pro druhý bod nyní navrhneme algoritmus. Budeme tvořit množiny neterminálů X_a a X_b, z nichž lze vygenerovat neprázdné slovo obsahující pouze písmena a, respektive pouze písmena b, tedy:

$$X_a = \{A \in \widehat{N} \mid A \Rightarrow^* a^k, k \in \mathbb{N}, k > 0\}$$
$$X_b = \{A \in \widehat{N} \mid A \Rightarrow^* b^k, k \in \mathbb{N}, k > 0\}$$

Potom gramatika $\widehat{G}$ splňuje požadavek ze zadání, jestliže se kořenový neterminál $\widehat{S}$ nachází alespoň v jedné z těchto množin, tedy pokud platí $\widehat{S} \in X_a \vee \widehat{S} \in X_b$.

Jelikož $\widehat{G}$ je jazykově ekvivalentní s G, tak i G splňuje požadavek ze zadání právě tehdy, když jej splňuje $\widehat{G}$.

Množiny X_a, X_b budeme tvořit iterativně: budeme tvořit množiny X_a^i, respektive X_b^i, které představují množiny neterminálů, které jsou kořeny derivačního stromu v G hloubky nejvýše i+1, jehož listy jsou jenom terminály a, respektive b.

Začneme od neterminálů, které generují slovo a (respektive b) – to jsou množiny X_a^0, X_b^0. Nyní budeme postupně tyto množiny rozšiřovat takto: máme-li X_a^i, pak X_a^{i+1} získáme tak, že k X_a^i přidáme neterminály A pro něž existuje pravidlo ve tvaru $A \to CD$, kde $C, D \in X_a^i$, tedy takové, které dokážeme přepsat na neterminály z minulé iterace:

$$X_a^0 = \{A \in \widehat{N} \mid A \to a \in \widehat{P}\}$$
$$X_a^{i+1} = X_a^i \cup \{A \in \widehat{N} \mid A \to CD \in \widehat{P}, C \in X_a^i, D \in X_a^i\}$$
$$X_b^0 = \{A \in \widehat{N} \mid A \to b \in \widehat{P}\}$$
$$X_b^{i+1} = X_b^i \cup \{A \in \widehat{N} \mid A \to CD \in \widehat{P}, C \in X_b^i, D \in X_b^i\}$$

Tyto iterace budou pokračovat tak dlouho, dokud se množiny mění, tedy do té doby, než nastane $X_a^i = X_a^{i+1}$, respektive $X_b^j = X_b^{j+1}$. Potom platí, že $X_a = X_a^i, X_b = X_b^j$.

IB102 – úkol 6, příklad 2 – řešení **Odevzdání: 4. 11. 2013**

Vypracoval(a): UČO:
Skupina:

Poznámka: Při sestavování množin X_a^0 (respektive X_b^0) stačí uvažovat pouze neterminály C takové, že existuje pravidlo tvaru $C \to c$, kde c je terminál (díky tomu, že gramatika je v CNF). Navíc při rozšiřování množin X_a^i, X_b^i nemusíme uvažovat pravidla jiného tvaru, než $A \to CD$, opět proto, že gramatika je v CNF.

Algorithm 1 Algoritmus rozhodující, zda gramatika G generuje slovo ve tvaru x^k pro nějaké $x \in \{a, b\}, k \in \mathbb{N}$

Vstup: bezkontextová gramatika $G = (N, \{a, b\}, P, S)$.

$\widehat{G} \leftarrow G$ transformovaná do CNF
nechť $\widehat{G} = (\widehat{N}, \Sigma, \widehat{P}, \widehat{S})$
$i \leftarrow 0$
$ii \leftarrow 0$
$X_a^i \leftarrow \{A \in \widehat{N} \mid A \to a \in \widehat{P}\}$
repeat
    $ii \leftarrow i$
    $i \leftarrow i + 1$
    $X_a^i \leftarrow X_a^{ii}$
    ▷ Přidáme všechny neterminály, které lze přepsat na dva neterminály z X_a^{ii}
    for all $A \to CD \in \widehat{P}$ **do**
        if $C \in X_a^{ii} \wedge D \in X_a^{ii}$ **then**
            $X_a^i \leftarrow X_a^i \cup \{A\}$
        end if
    end for
until $X_a^{ii} = X_a^i$ ▷ Dokud nedosáhneme toho, že se množina již dále nemění
$X_a \leftarrow X_a^i$

$j \leftarrow 0$
$jj \leftarrow 0$
$X_b^j \leftarrow \{A \in \widehat{N} \mid A \to b \in \widehat{P}\}$
repeat
    $jj \leftarrow j$
    $j \leftarrow j + 1$
    $X_b^j \leftarrow X_b^{jj}$
    for all $A \to CD \in \widehat{P}$ **do**
        if $C \in X_b^{jj} \wedge D \in X_b^{jj}$ **then**
            $X_b^j \leftarrow X_b^j \cup \{A\}$
        end if
    end for
until $X_b^{jj} = X_b^j$
$X_b \leftarrow X_b^j$

return $\widehat{S} \in X_a \vee \widehat{S} \in X_b$

Vypracoval(a): UČO:
Skupina:

Tento algoritmus je konvergentní (vždy skončí).

Důkaz. Konvergence algoritmu pro konverzi libovolné bezkontextové gramatiky do CNF byla dokázána na přednášce.

Řádky 3 – 17 a řádky 19 – 32 jsou strukturálně stejné a navzájem na sobě nezávislé, jen pro jiné vstupy, stačí tedy dokázat jeden případ, rozebereme tedy řádky 3 – 17.

- Cyklus for all na řádcích 11 – 14 iteruje přes konečnou množinu pravidel, tedy musí jistě skončit.
- Cyklus repeat skončí proto, že
  - vždy platí, že $X_a^{ii} \subseteq X_a^i$, tedy velikost množin se nemůže zmenšovat a jednou přidaný neterminál již nelze z množiny odstranit,
  - pokud do množiny X_a^i není přidán v dané iteraci žádný neterminál (tedy $X_a^i = X_a^{ii}$) pak cyklus skončí touto iterací,
  - množina X_a^i nemůže růst nekonečně, může nastat nejvýše případ, že $X_a^i = \widehat{N}$, tedy, že do množiny přidáme všechny neterminály.

  Proto nutně musí ukončovací podmínka cyklu nastat po konečném počtu kroků. □

Poznámka: Jistě by bylo možné tento příklad řešit i bez převodu gramatiky do CNF. Algoritmus by vypadal v podstatě stejně, nicméně konstrukce množin X_a, X_b by musela zohledňovat obecný tvar pravidel v bezkontextových gramatikách.