THE GENERAL SYNTAX ANALYSIS FOR MODIFIED E0L GRAMMARS

Radek BIDLO, Master Degree Programme (5)
Dept. of Information Systems, FIT, BUT
E-mail: xbidlo00@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

E0L grammars are parallel grammars from the vast family of L-systems. Commonly, the starting element is a string. This document describes the algoritm for the syntax analysis of languages generated from E0L grammars in the binary normal form. The starting structure will not be only a string, but a language generated from some context free grammar in strong Greibach normal form. In general, this language can be finite or infinite. We will get not only a powerful grammars — these grammars are more powerful than the context–free grammars — but also a tool for syntax analysis of languages generated from them.

1 ÚVOD

V tomto příspěvku představíme E0L gramatiku v binární normální formě (dále jen BNF) jejíž startovací strukturou je jiný jazyk generovaný nějakou bezkontextovou gramatikou v silné Greibachové normální formě (dále jen SGNF). Celá gramatika má potom větší vyjadřovací sílu než pouhá bezkontextová gramatika, tvary přepisovacích pravidel však respektují tvar pravidel bezkontextových gramatik. U čtenáře předpokládáme hlubší znalost teorie formálních jazyků.

2 PARALELNÍ GRAMATIKY

Paralelní gramatiky generují řetězce přepisem všech symbolů aktuální větné formy najednou. Mezi tento typ gramatik patří i E0L gramatiky. Protože standardní E0L gramatiky nepoužívají jako startovací strukturu celý jazyk a nemusí být nutně v BNF, označíme si naši gramatiku jako *E0L–M* gramatiku (*M* — modifikovaná). Pod tímto označením budeme v dalším textu rozumět vždy E0L gramatiku v BNF, jejíž startovací strukturou je jazyk generovaný nějakou bezkontextovou gramatikou v SGNF.

2.1 E0L–M GRAMATIKA

E0L–M gramatika G je čtveřice $G = (N, \Sigma, P, G_{start})$, kde

- N resp. Σ je konečná množina *nonterminálních* resp. *terminálních* symbolů, platí $N \cap \Sigma = \emptyset$
- P je konečná množina pravidel tvaru $A \to a$ a $A \to BC$, kde $A, B, C \in N$ a $a \in \Sigma$
- $G_{start} = (N_{start}, N, P_{start}, S)$ je gramatika v SGNF se startovacím symbolem $S \in N_{start}$, $L(G_{start}) \neq \emptyset$ je potom startovací jazyk gramatiky G

2.2 DERIVACE V E0L–M GRAMATICE

Nechť $G = (N, \Sigma, P, G_{start})$ je *E0L–M* gramatika a $\lambda = a_1 a_2 \ldots a_n$ a $\mu = \mu_1 \mu_2 \ldots \mu_n$ jsou větné formy této gramatiky. Mezi řetězci λ a μ platí relace $\Rightarrow$ nazývaná *přímá derivace*, jestliže $a_1 \to \mu_1 \in P \wedge a_2 \to \mu_2 \in P \wedge \ldots \wedge a_n \to \mu_n \in P$, kde μ_i pro $i = 1, 2, \ldots, n$ jsou pravé strany použitých přepisovacích pravidel. Potom píšeme $\lambda \Rightarrow \mu$ a říkáme, že řetězec μ lze přímo generovat z řetězce λ v uvažované *E0L–M* gramatice. Relace $\Rightarrow^n$, $\Rightarrow^*$ a $\Rightarrow^+$ mají stejný význam jako v oblasti sekvenčních gramatik.

2.3 JAZYK GENEROVANÝ E0L–M GRAMATIKOU

Nechť $G = (N, \Sigma, P, G_{start})$ je *E0L–M* gramatika. Jazyk nad touto gramatikou je definován množinou všech vět $L(G) = \{w | w_{start} \Rightarrow^* w \wedge w \in \Sigma^* \wedge w_{start} \in L(G_{start})\}$.

3 SYNTAKTICKÁ ANALÝZA

Pro syntaktickou analýzu zavedeme jednoduchou datovou strukturu — řetězec množin $X = M_1 M_2 \ldots M_n$, kde M_i pro $i = 1, 2, \ldots n$ jsou množiny. Ten bude mít stejné vlastnosti jako klasický řetězec s tím rozdílem, že jeho *symboly* budou množiny, do kterých teprve budeme vkládat skutečné symboly. Je to z důvodu možného výskytu přepisovacích pravidel se stejnou pravou a různou levou stranou v množině P. Použitím řetězce množin vyřešíme problém nedeterminismu, který by v uvedeném případě při syntaktické analýze nepochybně vznikl. $i-$tá množina v řetězci bude zpřístupňována zápisem $X[i]$. V syntaktické analýze se na řetězec množin budeme dívat jako na skutečný řetězec s tím rozdílem, že pokud budeme redukovat symboly na pozici i a $i+1$ uložené v řetězci množin v množinách $X[i]$ a $X[i+1]$, podíváme se do těchto množin a při zjišťování možné pravé strany přepisovacího pravidla budeme uvažovat všechny možné kombinace dvojic symbolů $\{AB | A \in X[i] \wedge B \in X[i+1]\}$. Výsledek syntaktické analýzy bude záviset na tom, zda algoritmus dokáže zredukovat vstupní řetězec na větnou formu, která je zároveň větou startovacího jazyka $L(G_{start})$. Protože je gramatika $L(G_{start})$ v SGNF, použijeme pro určení příslušnosti řetězce do tohoto jazyka algoritmus z [2], který je podrobně popsán na stranách 164–166. V případě kladné odpovědi analyzovaný řetězec patří do jazyka generovaného *E0L–M* gramatikou G. Pro účely našeho algoritmu si tento dílčí algoritmus označíme jako *SGNF–Algoritmus*.

4 ALGORITMUS SYNTAKTICKÉ ANALÝZY EOL-M GRAMATIK

Vstupem algoritmu je $E0L-M$ gramatika G a analyzovaný řetězec $w = a_1a_2\ldots a_n$, $n \geq 1$, **výstupem** potom informace, zda $w \in L(G)$, či nikoliv.

- vytvoř a inicializuj řetězce množin $str = w$ a $newstr = \varepsilon$ a proměnné *symbol* a *nextsymbol*; poznamenejme, že ε je prázdný řetězec
- dokud není konec řetězce *str*
  - do proměnné *symbol* ulož další symbol z řetězce *str*; jestliže $symbol \in \Sigma$ a $A \rightarrow symbol \in P$, potom přidej $\{A|A \rightarrow symbol \in P\}$ na konec řetězce *newstr*, jinak $w \notin L(G)$
- přiřaď $str = newstr$ a $newstr = \varepsilon$
- aplikuj *SGNF–Algoritmus* na *str*; dokud $str \notin L(G_{start})$
  - dokud není konec řetězce *str*
    - do proměnných *symbol* a *nextsymbol* ulož postupně další dva symboly z řetězce *str*
    - jestliže $A \rightarrow XY \in P$, $X \in symbol$, $Y \in nextsymbol$, potom přidej $\{A|A \rightarrow XY \in P\}$ na konec řetězce *newstr*, jinak $w \notin L(G)$
  - přiřaď $str = newstr$ a $newstr = \varepsilon$
- $w \in L(G)$

Poznamenejme, že analýza řetězce množin *str SGNF-Algoritmem* zahrnuje analýzu všech možných řetězců z tohoto řetězce množin. Podmínkou přijetí řetězce w je potom úspěšné přijetí alespoň jednoho řetězce ze *str SGNF–algoritmem*.

5 ZÁVĚR

Třída jazyků generovatelných z představené gramatiky je rozhodně vyšší, než je třída bezkontextových jazyků, neboť samotný startovací jazyk je již jazykem bezkontextovým (lze dokázat, že každá bezkontextová gramatika je transformovatelná na ekvivalentní gramatiku v SGNF). Uvedený algoritmus představuje mocný nástroj k syntaktické analýze jazyků generovaných *E0L–M* gramatikami.

REFERENCE

[1] MEDUNA, A.: *Automata and Languages*. London, Springer, 2000

[2] PRE–GRADUATE STUDENTS: *Proceedings of 9th Conference and Competition Student EEICT 2003 Volume 1*. Brno, 2003