REGULAR PATHS IN DERIVATION TREES OF CONTEXT-FREE GRAMMARS

Jiří Koutný
Doctoral Degree Programme (1), FIT BUT
E-mail: ikoutny@fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

ABSTRACT

To increase the generative capacity of context-free (CF, for short) grammars, Čulik and Maurer (see [1]) have published an idea of regular restricting the levels in the derivation trees of CF grammars. A simple and natural question is what happens if we put the same request not on the levels, but on the paths of derivation trees of CF grammars. In contrast with regular restricting the levels, regular restricting the paths does not increase the generative capacity of CF grammars. This paper formulates a formal proof.

1 INTRODUCTION

Chomsky has established a hierarchy in the classical theory of formal languages. One of the classes of the hierarchy is the class of CF languages. Parsing of this class of languages is, in contrast with the languages of higher classes, strongly sophisticated and common used e.g. in compilers. Natural request is to somehow increase the generative capacity of CF grammars and thus obtain languages from the higher class and simultaneously facilitate the usage of parsing methods for CF languages. Several types of controlling CF grammars have been developed, controlling the derivation tree among others. Čulik and Maurer in [1] have published the result that regular restrictions on the levels in the derivation tree of CF grammar increases the generative capacity of CF grammars. Marcus, Martín-Vide, Mitrana and Păun in [2] have published an idea of regular restricting a path in derivation tree of CF grammars and they have showed that such formalism does not increase the generative capacity of CF grammars. In conclusion of their paper (see [2]) they have formulated the question what happens if we require to have all the paths from a derivation tree described by a regular language. In this paper we will formulate the proof that regular restricting all the paths in the derivation trees of CF grammars does not increase the generative capacity of CF grammars.

2 PRELIMINARIES AND DEFINITIONS

This paper assumes that the reader is familiar with the theory of formal languages. The items that are not defined explicitly are standard in the theory of formal languages (see [3]). In this section, we will briefly review essential definitions required in the sequel.

Definition 1 Let Σ be an alphabet. The *regular expressions* (RE, for short) over Σ and the *languages they denote* are defined as follows: $\emptyset$ is an RE denoting the empty set; ε is an RE denoting $\{\varepsilon\}$; $a \in \Sigma$ is an RE denoting $\{a\}$; let r and s be regular expressions denoting the languages L_r and L_s, respectively, then: $(r.s)$ is an RE denoting $L = L_r L_s$; $(r+s)$ is an RE denoting $L = L_r \cup L_s$; (r^*) is an RE denoting $L = {L_r}^*$.
A language L is a *regular language* if there exists a regular expression r that denotes L, and the class of all languages generated by RE is denoted $\mathcal{L}(REG)$.

Definition 2 A *context-free grammar* is a quadruple $G = (N, T, P, S)$, where: N is an alphabet of *nonterminals*; T is an alphabet of *terminals*, $N \cap T = \emptyset$; P is a finite set of pairs $(A, x), A \in N, x \in (N \cup T)^*$, each such a pair $p = (A, x)$ is called a *production rule* and usually written as $A \to x$; $S \in N$ is the *start nonterminal*.
Let $u, v \in (N \cup T)^*$ and $p = A \to x \in P$. Then *uAv directly derives uxv* according to p in G, written as $uAv \Rightarrow uxv[p]$, and the reflexive, transitive closure of the relation $\Rightarrow$ is denoted by $\Rightarrow^*$. The *language generated by G* is denoted by $L(G)$ and defined by $L(G) = \{x \in \Sigma^* | S \Rightarrow^* x\}$.

The sequel will deal with the restrictions on the derivation trees of CF grammars and the following definitions are required.

Definition 3 A *derivation tree* for a grammar $G = (N, T, P, S)$ is a tree where: the root is the start nonterminal of G; the interior nodes are the nonterminals of G; the leaf nodes are the terminal symbols of G; the sons of a node T (from left to right) correspond to the symbols on the right hand side of some production for T in P.
Notation: Let $\triangle$ denotes a derivation tree and let $frontier(\triangle)$ be a function that returns the word obtained by concatenating all leaves of $\triangle$ from left to right. Let $\triangle(x)$ denotes the derivation tree $\triangle$, such that $frontier(\triangle) = x$, and let ${}_H\triangle(x)$ denotes the derivation tree $\triangle$ with respect to the grammar H, such that $frontier(\triangle) = x$. Let $root(\triangle)$ be a function that returns the root node of $\triangle$. Let $dist(x)$ be a function that returns the length of the path between $root(\triangle)$ and the node x.

Definition 4 Let $G = (N, T, P, S)$ be a CF grammar and let $r : A \to B_1 B_2 \ldots B_n \in P, A \in N, B_i \in (N \cup T)$, for $i = 1 \ldots n$, is a production rule. Then the *rule tree* that corresponds to the rule r is the derivation tree $\triangle$, such that $frontier(\triangle) = B_1 B_2 \ldots B_n$, for $i = 1 \ldots n$, $root(\triangle) = A$ and each node labeled by B_i is a son of the node A.

Definition 5 Let $\triangle$ be a derivation tree of a grammar G and M be a finite automata. *Enriched derivation tree*, denoted by $\overline{\triangle}$, is $\triangle$ with edges associated to the states of finite automata M. Let $\overline{\triangle}(x)$ and ${}_H\overline{\triangle}(x)$ be defined analogically with respect to the *Notation* in Definition 3.

Definition 6 Let $S({}_H\triangle(x))$ denotes the *set of all derivation trees*, such that $frontier(\triangle) = x$, with respect to grammar H. Let $S({}_H\overline{\triangle}(x))$ denotes the set of all enriched derivation trees, such that $frontier(\triangle) = x$, with respect to grammar H.

Definition 7 Let $\triangle$ be a derivation tree. Every sequence s of nodes, $s = a_1 a_2 \ldots a_i, i = 1 \ldots i$, such that for each two nodes $a_i, a_j, i \neq j, dist(a_i) = dist(a_j), i, j = 1 \ldots n$ is called the *level* of $\triangle$. Let $level(s)$ be a function that returns the word obtained by concatenating all symbols in s.

The idea of controlling the levels of the derivation trees of CF grammars was introduced by Čulik and Maurer in [1] as follows.

Definition 8 CF grammar with regular controlling the levels in the derivation trees (CFRCL, for short) is a pair (G,R), where $G=(N,T,P,S)$ is a *CF grammar* and $R \subseteq (N \cup T)^*$ is a *regular language*. The *language generated* by (G,R) is denoted by $L(G,R)$ and defined by $L(G,R)=\{x \in L(G) |$ there exists $\triangle(x)$ such that for each level s (except the last one) it holds that $level(s) \in R\}$. The class of all languages generated by CFRCL is denoted as $\mathcal{L}(CFRCL)$.

A simple and natural extension of the idea of CFRCL grammars is the *CF grammar with regular controlling the paths in its derivation tree*.

Definition 9 Let $\triangle$ be a derivation tree. Every sequence s of nodes, $s=a_1a_2\ldots a_i, i=1\ldots n$, such that $a_1=root(\triangle)$, a_n is a leaf of $\triangle$ and for each $i=1\ldots n-1$ there exists an edge from a_i to a_{i+1} in $\triangle$, is called *path* in $\triangle$. Let $path(s)$ be a function that returns the word obtained by concatentating all symbols in s.

Definition 10 CF grammar with regular controlling the paths in the derivation trees (CFRCP, for short) is a pair (G,R), where $G=(N,T,P,S)$ is a *CF grammar* and $R \subseteq (N \cup T)^*$ is a *regular language*. The *language generated* by (G,R) is denoted by $L(G,R)$ and defined by $L(G,R)=\{x \in L(G) |$ there exists $\triangle(x)$ such that for each path s it holds that $path(s) \in R\}$. The class of all languages generated by CFRCP is denoted as $\mathcal{L}(CFRCP)$.

3 MAIN RESULT

Čulik and Maurer in [1] have proved that the *CFRCL grammars* can generate all the laguages of type 0 of the Chomsky hierarchy, thus *CFRCL grammars* have the generative capacity equal to Turing machines. One could expected that a simple and natural extension of their concept in the form of regular controlling the paths in the derivation trees will also increase generative capacity and it would be interesting to study the relationship between language classes $\mathcal{L}(CFRCL)$ and $\mathcal{L}(CFRCP)$. However, the result is in contrast to the expectations.

In this section we will examine the generative capacity of CFRCP grammars and we will show that regular controlling the paths in the derivation trees of CF grammars do not increase generative capacity of CF grammar. We will formulate the formal proof of this claim based on an algorithm that for each CFRCP grammar (G,R) finds CF grammar H, such that $L(G,R)=L(H)$.

Algorithm 1 *Transform CFRCP (G,R) to CF H.* Let (G,R) be a *CFRCP* grammar, where $G=(N_G,T,P_G,S_G)$ and R is a *regular language*. Without any loss of generality, we can assume that there exists *deterministic finite automata* $M=(Q_M,N_G \cup T,R_M,s_M,F_M)$, such that $L(M)=R$. Let $H=(N_H,T,P_H,S_H)$ be a CF grammar, defined by the following algorithm:

- For each $p: A \rightarrow B_1B_2\ldots B_n \in P_G$:
  - If $qA \rightarrow q_A \in R_M$, for any $q \in Q_M$, and it holds that $q_AB_i \rightarrow q_{B_i} \in R_M$, for each $i, i=1,2,\ldots,n$, then

– Add $\langle A, q_A \rangle$ to N_H and if $A = S_G$ and $q = s_M$, then $\langle A, q_A \rangle = S_H$
– For each $j, j = 1, 2, \ldots, n$:
  – Add $\langle B_j, q_{B_j} \rangle$ to N_H and if it holds that $B_j \in T$ and $q_{B_j} \in F_M$, then add $\langle B_j, q_{B_j} \rangle \rightarrow B_j$ to P_H
– Add $\langle A, q_A \rangle \rightarrow \langle B_1, q_{B_1} \rangle \langle B_2, q_{B_2} \rangle \ldots \langle B_n, q_{B_n} \rangle$ to P_H

Lemma 1 There exists $t \in S({}_{(G,R)}\overline{\triangle}(x)), x \in (N_G \cup T)^*$, if and only if there exists $d \in S({}_H\triangle(y))$, $y = \langle B_1, q_{B_1} \rangle \langle B_2, q_{B_2} \rangle \ldots \langle B_n, q_{B_n} \rangle \in N_H^*, n \geq 0$, such that $t \cong d$ and $x = B_1 B_2 \ldots B_n$.

Proof: Let g be a bijection on $S({}_{(G,R)}\overline{\triangle}(B_1 B_2 \ldots B_n)), B_i \in (N \cup T)$, and $S({}_H\triangle(\langle B_1, q_{B_1} \rangle \langle B_2, q_{B_2} \rangle \ldots \langle B_n, q_{B_n} \rangle)), \langle B_n, q_{B_n} \rangle \in N_H$, for each $n \geq 0$, defined by: For each node $A \neq root(t)$ and an edge q, such that $q'A \rightarrow q \in R_M$, for any $q' \in Q_M$, it holds $g(A) = \langle A, q \rangle$, where $\langle A, q \rangle$ is a node in d. For $S_G = root(t)$ it holds $g(S_G) = \langle S_G, q_{S_G} \rangle$, where $s_M S_G \rightarrow q_{S_G} \in R_M$ and $\langle A, q \rangle$ is the root node of d. For each edge $q(A, B)$, where A, B are nodes in t, it holds $g(q(A, B)) = q'(g(A), g(B))$ is an edge in d. Let g^{-1} be inverse of g.

$\Rightarrow$*:* This is established by induction on the number of the rule trees in $t \in S({}_{(G,R)}\overline{\triangle}(x)), x \in (N_G \cup T)^*$.
Basis: Let $i = 0$. The only rule tree in t is S_G, the node which corresponds to the start nonterminal of (G, R). Clearly the only rule tree in d is $g(S_G)$, the node which corresponds to the start nonterminal of H.
IH: Let us suppose that our claim holds for any enriched derivation tree $t \in S({}_{(G,R)}\overline{\triangle}(x)), x \in (N_G \cup T)^*$, that contains at most k, for some $k \geq 0$, rule trees.
IS: Let $t \in S({}_{(G,R)}\overline{\triangle}(x)), x \in (N_G \cup T)^*$, be any enriched derivation tree that contains $k+1$ rule trees. Let $x = uvw$, where $u, v, w \in (N \cup T)^*$, and let t contains an enriched subtree $\overline{\triangle}(v)$ such that all its nodes (except the $root(\overline{\triangle}(v))$) are leaves. Let us remove just one rule tree from t that is if $root(\overline{\triangle}(v)) = B$, then $\overline{\triangle}(uBv)$, where u and w are prefix and suffix of x, respectively, and $B \rightarrow v \in P_G$, is a subtree of t. Thus, by induction hypothesis, $g(\overline{\triangle}(uBv))$ is a subtree of d and, by Algorithm 1, $frontier(g(\overline{\triangle}(uBv))) = u' \langle B, q \rangle w'$, where $\langle B, q \rangle \in N_H$ and $u', w' \in (N_H \cup T)^*$. Because for each B_i in v there exists $q_B B_i \rightarrow q_{B_i} \in R_M$, then there exists the rule $r : \langle B, q \rangle \rightarrow \langle B_1, q_{B_1} \rangle \langle B_2, q_{B_2} \rangle \ldots \langle B_n, q_{B_n} \rangle \in P_H$, where $B_1 B_2 \ldots B_n = v$, and we can generate $d = \overline{\triangle}(u' \langle B_1, q_{B_1} \rangle \langle B_2, q_{B_2} \rangle \ldots \langle B_n, q_{B_n} \rangle w')$ such that $g(t) = d$.

$\Leftarrow$*:* This is established by induction on the number of the rule trees in $d \in S({}_H\triangle(y)), y = \langle B_1, q_{B_1} \rangle \langle B_2, q_{B_2} \rangle \ldots \langle B_n, q_{B_n} \rangle \in N_H^*, n \geq 0$.
Basis: Let $i = 0$. The only rule tree in d is $\langle S_G, q_{S_G} \rangle$, the node that corresponds to the start nontermnal of H. The only rule tree in t is $g^{-1}(S_G, q_{S_G})$, the node that corresponds to the start nonterminal of (G, R).
IH: Let us suppose that our claim holds for any derivation tree $d \in S({}_H\triangle(y)), y = \langle B_1, q_{B_1} \rangle \langle B_2, q_{B_2} \rangle \ldots \langle B_n, q_{B_n} \rangle \in N_H^*, n \geq 0$, that contains at most k, for some $k \geq 0$, rule trees.
IS: Let $d \in S({}_H\triangle(y)), y = \langle B_1, q_{B_1} \rangle \langle B_2, q_{B_2} \rangle \ldots \langle B_n, q_{B_n} \rangle \in N_H^*, n \geq 0$, be any derivation tree that contains $k+1$ rule trees. Let $y = uvw$, where $u, v, w \in N_H^*$, and let d contains a subtree $\triangle(v)$ such that all its nodes (except the $root(\triangle(v))$) are leaves. Let us remove jus one rule tree from d that is if $root(\triangle(v)) = \langle B, q \rangle$, then $\triangle(u' \langle B, q \rangle v')$, where v' and w' are prefix and suffix of y, respectively, and $\langle B, q \rangle \rightarrow v \in P_H$, is a subtree of d. Thus, by induction hypothesis, there exists

a subtree t' of t such that $g(t') = \triangle(u'\langle B,q\rangle v')$. Let $frontier(t') = uBv$, where $B \in N_G$ and $u,w \in (N_G \cup T)^*$. Because $r : \langle B,q\rangle \to v \in P_H$, where $v = \langle B_1,q_1\rangle\langle B_2,q_2\rangle \ldots \langle B_n,q_n\rangle \in N_H^*, n \geq 0$, then the rule $r : B \to B_1B_2 \ldots B_n \in P_G$ and $qB_1 \to q_{B_1}, qB_2 \to q_{B_2}, \ldots, qB_n \to q_{B_n} \in R_M$ and we can genereate $t = \triangle(uB_1B_2 \ldots B_nw)$ such that $g(t) = d$.

Theorem 1 $\mathcal{L}(CFRCP) = \mathcal{L}(CF)$.

Proof: By using Lemma 1, we will show the equivalence $L(G,R) = L(H)$, where (G,R) is any CFRCP grammar and H is a CF grammar that is constructed by Algorithm 1.

$L(G,R) \subseteq L(H)$: Let $t \in S({}_{(G,R)}\overline{\triangle}(x)), x \in T^*$, be any enriched derivation tree, such that $g(t) = d \in S({}_H\triangle(y)), y = \langle B_1,q_{B_1}\rangle\langle B_2,q_{B_2}\rangle \ldots \langle B_n,q_{B_n}\rangle \in N_H^*, n \geq 0,$ where $x = B_1B_2 \ldots B_n, n \geq 0$. Then $x \in L(G,R)$. Because for each B_i in x there exists $qB_i \to q_{B_i} \in R_M$, where $q_{B_i} \in F_M$, then $y = \langle B_1,q_{B_1}\rangle\langle B_2,q_{B_2}\rangle \ldots \langle B_n,q_{B_n}\rangle$, $q_{B_i} \in F_M$, for each $i = 1 \ldots n$, and for each $\langle B_i,q_{B_i}\rangle$ in y there exists $r_i : \langle B_i,q_{B_i}\rangle \to B_i \in P_H$, where $B_i \in T$. Thus, we can generate $B_1B_2 \ldots B_n \in L(H)$ and thus $L(G,R) \subseteq L(H)$.

$L(G,R) \supseteq L(H)$: Let $d' \in S({}_H\triangle(y))$ be any derivation tree, such that $y = B_1B_2 \ldots B_n \in L(H)$. Let d be a subtree of d' obtained by removing all the nodes labeled by $B \in T$. Because for each B_i in y there exists $\langle B_i,q_{B_i}\rangle \to B_i \in P_H$, where $q_{B_i} \in F_M$, then $frontier(d) = \langle B_1,q_{B_1}\rangle\langle B_2,q_{B_1}\rangle \ldots \langle B_n$, where $q_{B_1}\rangle$,$q_{B_i} \in F_M$, for each $i = 1 \ldots n$, and thus there exists enriched derivation tree t such that $g(t) = d$ and $frontier(t) = B_1B_2 \ldots B_n = x \in (G,R)$. Thus, $L(G,R) \supseteq L(H)$.
Thus, $\mathcal{L}(CFRCP) = \mathcal{L}(CF)$.

4 CONCLUSION

As we have showed in Theorem 1, regular controlling the paths in the derivation trees of the CF grammars does not increase the generative capacity of CF grammars. It is interesting that if we allow the control set of such grammar to be a language of slightly higher type, a linear language to be concrete, then the generative capacity will increase significantly and it is possible to generate languages beyond $\mathcal{L}(CF)$. Moreover, it is surprising that if we allow a control set of CF grammar to be linear language and if we require to have at least one path from derivation tree described by a linear language, then it is still sufficient formalism to generate languages beyond the CF class (see [2]).

REFERENCES

[1] K. Čulik and H. A. Maurer. Tree controlled grammars. *Computing*, 19:129–139, 1977.

[2] S. Marcus, C. Martín-Vide, V. Mitrana, and G. Paun. A new-old class of linguistically motivated regulated grammars. In *CLIN*, volume 37 of *Language and Computers - Studies in Practical Linguistics*, pages 111–125. Rodopi, 2000.

[3] A. Meduna. *Automata and Languages: Theory and Applications [Springer, 2000]*. Springer Verlag, 2005.