SYNTAX ANALYSIS OF TREE-CONTROLLED LANGUAGES

Jiří Koutný
Doctoral Degree Programme (3), FIT BUT
E-mail: ikoutny@fit.vutbr.cz

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

Abstract: Syntax analysis of generatively stronger than context-free grammars is usually a major problem because of problematic construction of practically usable parsing methods. The paper introduces a generatively stronger grammar based on the restrictions placed upon the paths in the derivation trees of context-free grammars and discusses polynomial time parsing methods possibilities for it.

Keywords: context-free grammars, tree-controlled grammars, paths, syntax analysis, parsing

1 INTRODUCTION AND MOTIVATION

A syntax analysis of some family of languages has been always considered basically from two points of view. Firstly, theoretical viewpoint is concerned with finding whether the given string over some alphabet belongs to given language. Secondly, from practical viewpoint, it is essential that the membership is decidable in polynomial time depending on the length of the string. For many well-known language families (i.e. regular, linear, or context-free family of languages), the polynomial parsing methods has been introduced and their implementations are commonly used in practice. However, for the generatively stronger grammars (typically context-sensitive grammars), the parsing methods are much more complex. In essence, for this reason, new generative models are introduced and investigated and the natural request placed on them is the polynomial parsability. Typically, they are based on some variant of the restrictions placed on some of the well-known model (see [3]).

One of such a restricted model is based on *tree-controlled grammars* (see [1]) which places several types of the restrictions upon the derivation trees of context-free grammars (see [4], [7], [8], and [9]). Especially the research of [7] introduces the restriction placed on just one path in the derivation trees and also briefly discusses the syntax analysis of such a model. In essence, this paper generalizes the model of [7] to *n*-paths restriction, then it recalls that in this case there are several language families depending on the pumping lemma for linear languages applied on the control language (see [6]), and primarily discusses the idea of parsing methods working in polynomial time for one of the families.

2 PRELIMINARIES AND DEFINITIONS

This paper assumes that the reader is familiar with the graph theory, the basics of asymptotic complexity, and the theory of formal languages, including the theory of regulated rewriting. In this section, we introduce the terminology and the definitions needed in the sequel.

For an alphabet V, V^* denotes the letter monoid (generated by V under the operation concatenation), ε is the unit of V^*, and $V^+ = V^* - \{\varepsilon\}$. For string $x \in V^*$, $|x|$ denotes the length of x. Every subset $L \subseteq V^*$ is a *language* over V.

A *context-free grammar* is a quadruple $G = (V,T,P,S)$ where, as usual, V is a total alphabet, $T \subseteq V$ is a terminal alphabet, P is a finite set of rules of the form $A \to x$ where $A \in V - T$, $x \in V^*$, and $S \in V - T$ is the starting symbol. A grammar G is *linear*, if and only if all its rules have at most one nonterminal on the right-hand side. A derivation step $\Rightarrow$ in G and the relation $\Rightarrow^*$ are defined in the standard manner. The language of context-free grammar is defined as $L(G) = \{x \in T^* | \, S \Rightarrow_G^* x\}$. The family of linear, context-free languages is denoted by **LIN**, **CF**, respectively.

Let t be a *derivation tree* of $x \in T^*$ in $G = (V,T,P,S)$. A *path* of t is any sequence of the nodes of t with the first node equals to the root of t, last node equals to a leaf of t, and there is an edge in t between each two consecutive nodes of the sequence. Let $word(s)$ denote the string obtained by concatenating all symbols of the sequence of nodes in a derivation tree.

A *tree-controlled* grammar, TC grammar for short, is a pair (G,R) where $G = (V,T,P,S)$ is a context-free grammar, and R is a control language over V. There are several types of languages generated by TC grammars, (see [6]), however, due to space restrictions, we present just one of them.

Let (G,R) be a TC grammar. The *language that* (G,R) *generates under the n-path control by* R, $n \geq 0$, is denoted by ${}_{n-path}L(G,R)$ and defined by the following equivalence:

For all $x \in T^*$, $x \in_{n-path} L(G,R)$ if and only if there is a derivation tree t of x in G such that there is set C_t of n different paths of t that are divided in a common node of t and for each $p \in C_t$, $word(p) \in R$. Set **n-path-TC** $= \{{}_{n-path}L(G,R) | \, (G,R)$ is a TC grammar$\}$.

Since regular-restricted paths in the derivation trees of context-free grammars do not increase the generative capacity of context-free grammars (see [5] and Prop. 2 in [7]), we consider TC grammars with context-free component G and linear component R.

Let (G,R) be a TC grammar that generates the language under n-path restriction by R. Clearly, for a derivation tree t of $z \in L(G)$, there is some $m_{C_t} \geq 1$ that denotes a number of common nodes for all n controlled paths. Since $R \in$ **LIN**, the pumping lemma for linear languages holds for R. Depending on the value of m_{C_t} in relation to the pumping lemma for R, five types of languages in **n-path-TC** can be introduced (see [6]). However, hereafter we deal with just one of them.

Let ${}_{n-path}L(G,R)$ be a language of a TC grammar (G,R) with $G = (V,T,P,S)$, for some $n \geq 0$. Then C_t is a set of n controlled paths in a derivation tree of a sentence in ${}_{n-path}L(G,R)$. Let $p_s \in C_t$ be the shortest controlled path and $word(p_s)$ be divided into five parts $uvwxy$ according the pumping lemma for linear languages. If for every $z \in_{n-path} L(G,R)$ with derivation tree t in G and for every $p \in C_t$, $|uv| < m_{C_t} \leq |uvw|$, then ${}_{n-path}L(G,R)$ is ${}_{III-n-path}L(G,R)$. Set **III-n-path-TC** $= \{L|$ there is TC grammar (G,R) such that $L =_{III-n-path} L(G,R)\}$. It is well-known that **CF** $\subseteq$ **III-n-path-TC** even for $n = 1$ (see [6] and [7]).

3 SYNTAX ANALYSIS IDEA OF III-n-path-TC

Question 1. Can $x \in_{III-n-path} L(G,R)$ for TC grammar (G,R) be decided in $O(|x|^k)$ with $k \in \mathbb{N}$, $n \geq 0$?

Idea 1. Let ${}_{III-n-path}L(G,R)$ be a language, for some $n \geq 0$, of a TC grammar (G,R) where $G = (V,T,P,S)$. Assume, G is unambiguous. It is well-known that we can decide if $x \in L(G)$ in $O(|x|^2)$, thus, we distinguish two cases, (1) $x \notin L(G)$, and (2) $x \in L(G)$.

(1) Clearly, if $x \notin L(G)$, then $x \notin_{III-n-path} L(G,R)$.

(2) If $x \in L(G)$, then, since G is unambiguous, we can construct unique derivation tree t of $x \in L(G)$ in $O(|x|^2)$. Since each path of t ends in a leaf, t contains $|x|$ paths. Clearly, since G is unambiguous, the height of each t is bounded by some $l \in \mathbb{N}$. Thus, the length of each path p of t and therefore also $|word(p)|$ are bounded by l. Since $R \in$ **LIN** and $|word(p)| \leq l$, for each $p \in C_t$, it is well-known that we can decide if $word(p) \in L(R)$ in polynomial time. Since $R \in$ **LIN**, it is straightforward task to determine where is the middle part related to the

pumping lemma for **LIN**. If for at least n paths $p_1, p_2, \ldots, p_n$ of derivation tree of x in G holds (a) $word(p_i) \in R$, for $i \in 1,2,\ldots,n$, (b) $p_1, p_2, \ldots, p_n$ are divided in common node, (c) which is placed in the middle parts of $p_1, p_2, \ldots, p_n$, then $x \in_{III-n-path} L(G,R)$.

Answer 1. Yes, $x \in_{III-n-path} L(G,R)$ for TC grammar (G,R) can be decided in $O(|x|^k)$ with $k \in \mathbb{N}$ under assumption that G is unambiguous, $n \geq 0$.

The weakness of Idea 1 above is the assumption that a context-free grammar is unambiguous. However, it is well-known that the question if a grammar is or is not ambiguous is undecidable, since this problem can be reduced to Post correspondence problem which is undecidable.

It is also well-known that for some ambiguous context-free grammars, there exist equivalent context-free grammar which is unambiguous. The ambiguity of a context-free grammar can be restricted basically by removing the chain rules. Without any loss of generality, assume that a context-free grammar contains only usable rules—that is, only those rules, which can be used during the derivation. Clearly, if $G = (V,T,P,S)$ is a context-free grammar with $r : A \rightarrow B \in P$, for some $A, B \in V - T$, then G is potentially ambiguous.

Obviously, since chain rules generate nothing, they can be removed from a context-free grammar G without affecting $L(G)$. However, removing the chain rules from G in a TC grammar (G,R) affect the paths in the derivation trees of $x \in L(G)$. Thus, the equivalence ${}_{III-n-path}L(G,R) =_{III-n-path} L(G',R)$, where G' is obtained by removing the chain rules from G, does not hold, however the equivalence $L(G) = L(G')$ holds. Therefore, the second fundamental question is:

Question 2. In a TC grammar (G,R), can we obtain G' by removing the chain rules from G and modify R to R' such that ${}_{III-n-path}L(G,R) =_{III-n-path} L(G',R')$, $n \geq 0$?

Idea 2. Let ${}_{III-n-path}L(G,R)$ be a language, for some $n \geq 0$, of a TC grammar (G,R) where $G = (V,T,P,S)$. Let G' be a context-free grammar obtained from G by removing chain rules. Therefore, G' can be created by well-known algorithm in polynomial time. We get $G' = (V,T,P',S)$ such that for all $x \in L(G')$, there is no derivation in G' of the form $B \Rightarrow^* A$, for some $A, B \in V - T$.

The paths in the derivation trees of G' are described by the strings of the form N^*T. Basically, we need to read such a strings and remove such symbols $A \in N$ which corresponds to the application of $B \rightarrow A \in P$ in G. This is done by gsm mapping M (see [3] for the definition) such that M reads the strings s of the form N^*T and nondeterministicaly removes or lets unchanged each symbol $A \in N$ with $B \rightarrow A \in P$ and BA is substring of s. This way, we get $M(R)$ such that $M(R) \neq R$, however, ${}_{III-n-path}L(G,R) =_{III-n-path} L(G',M(R))$. Since **LIN** is closed under gsm mappings (see [2]), also $M(R) \in$ **LIN**.

Answer 2. Yes, a TC grammar (G,R) can be transformed into TC grammar (G',R'), where G' does not contain chain rules and ${}_{III-n-path}L(G,R) =_{III-n-path} L(G',R')$, $n \geq 0$.

Clearly, the application of Idea 2 above does not guarantee that in the resulting (G',R'), G' is unambiguous because of the chain rules are not the only cause of the ambiguity. Consider, however, any $x \in_{III-n-path} L(G',R')$. Obviously, there is a derivation tree t of x in G'. Since there is no chain rules in G' and for each $x \in L(G')$, $|x|$ is finite, the height of t is at most equal to $log|x|/log2$. Thus, there is at most m, for some $m \in \mathbb{N}$, derivation trees of x in G'—that is, G' is m-ambiguous. The following question arises:

Question 3. Can $x \in_{III-n-path} L(G,R)$ for TC grammar (G,R) be decided in $O(|x|^k)$, with $k \in \mathbb{N}$, $n \geq 0$, and G is m-ambiguous?

Idea 3. Let ${}_{III-n-path}L(G,R)$ be a language, for some $n \geq 0$, of a TC grammar (G,R). We can straightforwardly adjust Idea 1 above as follows. The decision if $x \in L(G)$ is done exactly in the same way as in Idea 1 above. If $x \in L(G)$, then we can construct at most m derivation trees of $x \in L(G)$ in $O(m.|x|^2)$. Now, the idea is also the same as in Idea 1, the only modification is in the last step, which can be reformulated as: If for at least n paths $p_1, p_2, \dots, p_n$ of at least one derivation tree of x holds (a) $word(p_i) \in R$, for $i \in 1, 2, \dots, n$, (b) $p_1, p_2, \dots, p_n$ are divided in common node, (c) which is placed in the middle parts of $p_1, p_2, \dots, p_n$, then $x \in_{III-n-path} L(G,R)$.

Answer 3. Yes, $x \in_{III-n-path} L(G,R)$ for TC grammar (G,R) can be decided in $O(|x|^k)$ with $k \in \mathbb{N}$, $n \geq 0$, under assumption that G is m-ambiguous.

However, as follows from Idea 1 and Idea 3 above, the parsing is done basically in two phases—(a) construction of derivation tree t of x in G, (b) checking that at least n paths divided in common node of at least one t are described by R. From the practical viewpoint, the situation may occur in which we already know during the phase (a) above that currently constructed derivation tree cannot contain the required number of paths described by the strings from R—informally, we do not have to wait with starting the phase (b) until the phase (a) is completely done. Therefore, another question arises:

Question 4. Is it possible that the above-mentioned phases of syntax analysis run concurrently?

Idea 4. Let ${}_{III-n-path}L(G,R)$ be a language, for some $n \geq 0$, of a TC grammar (G,R) where $G = (V,T,P,S)$ is m-ambiguous. We can adjust Idea 3 as follows. Without any loss of generality, we assume that R is generated by linear grammar $G_R = (V_R, V, P_R, S_R)$.

Consider labeled derivation tree with the set of labels $\{0,1\}$. The semantic is as follows. Let e be an edge between two nodes of derivation tree t in G. Then, label 0 means that for path p that contains e, $word(p) \notin R$. Label 1 means that path p that contains e can potentially be described by R.

Consider that for the decision if $x \in L(G)$, we use a top-down parsing method to construct derivation tree t of x in G—that is, started from S, we try to construct derivation tree according to the rules of G such that the frontier of t is equal to x. Let us suppose that a rule $p : A \to A_1 A_2 \dots A_n \in P$ is used in the derivation $X \Rightarrow Y$ in G and in addition, we need to determine the value of the labels for the edges between A and each A_i, for $i = 1, 2, \dots, n$. Let t' be a derivation tree that corresponds to the derivation $S \Rightarrow^* w_1 A_1 A_2 \dots A_n w_2$, for some $w_1, w_2 \in V^*$, in G. Essentially, t' is a subtree of t. Clearly, each path of t' is the beginning part of at least one path in t.

Let t'' be a subtree of t' such that t'' contains just those nodes of t' which are connected by the edges labeled by 1, or by the edges without label. If t'' contains less than n paths, then $x \notin_{III-n-path} L(G,R)$. If all the edges of t'' are labeled, we can proceed to next derivation step in G. If some of the edges in t'' are not labeled, we need to compute the values of missing labels. Thus, for each path p'' in t'', we check whether G_R can generate the string of the form $word(p'')w$ with $w \in N^*T$. Since $R \in \mathbf{LIN}$, each sentential form in G_R is of the form $w_1 C w_2$, where $w_1, w_2 \in V^*$, $C \in V_R - V$.

For each path p'' in t'', since $|word(p'')|$ is finite, we can check whether $w_1 = word(p'')$ for at least one of the possible derivations of the form $w_1 C w_2$ with $|w_1| = |word(p'')|$ in G_R. If there is $w_1 = word(p'')$, we add label 1 to the input edge of the last node of p''. If $w_1 \neq word(p'')$ for all the possible derivations, we add label 0 to the input edge of the last node of p''. Notice that since $R \in \mathbf{LIN}$, this phase can also be optimized in such a way that we do the test whether $w_1 = word(p'')$ symbol-by-symbol during the generation of w_1 in G_R. The details of this optimization represents a straightforward task which is left to the reader.

Now, we construct subtree t''' of t'' such that t''' contains only those nodes of t'' which are connected by the edges labeled by 1 and we do the final check whether t''' contains at least n paths. If t''' do not contain at least n paths, then $x \notin_{III-n-path} L(G,R)$. If t''' contains at least n paths and if t' contains at least one leaf labeled by symbol of N, we proceed to next derivation step in G. If t''' contains at least n paths divided in the common node in the middle part in relation to the pumping lemma for R, and if all the leafs of t' are labeled by the symbols of T, then $x \in_{III-n-path} L(G,R)$.

Answer 4. Yes, it is possible to check whether the paths of derivation tree t of context-free grammar can potentially be described by given linear language already during the building of t.

Idea 4 deals in principle with top-down parsing method. Essentially the same idea is applicable also on bottom-up parsing methods, however, due to the space restrictions, it is left to the reader. However, from the ideas described above, one fundamental open question follows:

Open Question 1. Can $x \in_{III-n-path} L(G,R)$ for TC grammar (G,R) be decided in $O(|x|^k)$ with $k \in \mathbb{N}$, $n \geq 0$, and G ambiguous?

4 CONCLUSION

We conclude the paper by stating that for $L \in$ **III-n-path-TC** under assumption that L is generated by TC grammar (G,R) in which G has bounded ambiguity (i.e. G is unambiguous or m-ambiguous), there is parsing method working in polynomial time. This method can check whether or not the paths of the derivation tree t of $x \in L(G)$ belongs to control language R in the time of building of t. However, the natural question that still remains unanswered is whether or not this is true also if G is ambiguous. All these questions and answers play an important role when implementing syntax analyzer for **III-n-path-TC** in which some typical non-context-free languages belong (see [6]).

ACKNOWLEDGEMENT

This work was supported by the research plan MSM0021630528 and by MŠMT grant MEB041003.

REFERENCES

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

[2] J. Dassow, Gh. Păun, and A. Salomaa. Grammars with controlled derivations. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Volume II*, pages 101–154. Berlin: Springer, 1997.

[3] J. Dassow and Gh. Păun. *Regulated Rewriting in Formal Language Theory*. Springer, Berlin, 1989.

[4] J. Dassow and B. Truthe. Subregularly tree controlled grammars and languages. In *Automata and Formal Languages - 12th International Conference AFL 2008, Balatonfured*, pages 158–169. Computer and Automation Research Institute of the Hungarian Academy of Sciences, 2008.

[5] J. Koutný. Regular paths in derivation trees of context-free grammars. In *Proceedings of the 15th Conference and Competition STUDENT EEICT 2009 Volume 4*, pages 410–414. Faculty of Information Technology BUT, 2009.

[6] J. Koutný. On n-path-controlled grammars. In *Proceedings of the 16th Conference and Competition STUDENT EEICT 2010 Volume 5*, pages 176–180. Faculty of Information Technology BUT, 2010.

[7] S. Marcus, C. Martín-Vide, V. Mitrana, and Gh. Păun. A new-old class of linguistically motivated regulated grammars. In *CLIN*, pages 111–125, 2000.

[8] C. Martín-Vide and V. Mitrana. Further properties of path-controlled grammars. In *Formal Grammar / Mathematics of Language 2005*, pages 219–230. Edimburgh, 2005.

[9] Gh. Păun. On the generative capacity of tree controlled grammars. *Computing*, 21(3):213–220, 1979.