Jiří Koutný; Alexander Meduna

Tree-controlled grammars with restrictions placed upon cuts and paths

Kybernetika, Vol. 48 (2012), No. 1, 165--175

Persistent URL: http://dml.cz/dmlcz/142069

KYBERNETIKA — VOLUME *48* (2012), NUMBER 1, PAGES 165–175

TREE-CONTROLLED GRAMMARS WITH RESTRICTIONS PLACED UPON CUTS AND PATHS

Jiří Koutný and Alexander Meduna

First, this paper discusses tree-controlled grammars with root-to-leaf derivation-tree paths restricted by control languages. It demonstrates that if the control languages are regular, these grammars generate the family of context-free languages. Then, in a similar way, the paper introduces tree-controlled grammars with derivation-tree cuts restricted by control languages. It proves that if the cuts are restricted by regular languages, these grammars generate the family of recursively enumerable languages. In addition, it places a binary-relation-based restriction upon these grammars and demonstrate that this additional restriction does not affect the generative power of these grammars.

Keywords: context-free grammars, tree-controlled grammars, restricted derivation trees, paths, cuts, language families

Classification: 68Q42

1. INTRODUCTION

Indisputably, the investigation of context-free grammars with restricted derivation trees represents an important trend in today's formal language theory as demonstrated by several publications on this subject (see [3, 5, 6, 7, 9]). In essence, these grammars generate their languages just like ordinary context-free grammars do; in addition, however, their derivation trees have to satisfy some prescribed conditions. The present paper continues with the investigation of this kind.

More specifically, in this paper, we restrict every root-to-leaf path in the derivation trees of context-free grammars by some control languages. We demonstrate that if these control languages are regular, the generative power of context-free grammars remains unchanged – that is, they characterize the family of context-free languages. This result is of some interest when compared to the study given in [3], which restricts tree levels rather than paths in this way and proves that the resulting grammars characterize the family of recursively enumerable languages. Let us also point out that our result significantly generalizes the study of [6], which only requires that there is at least one root-to-leaf path in derivation tree restricted by a regular language. Indeed, by [6, Prop. 2], if derivation trees are restricted so they have to contain at least one path in the given control regular language, then this restriction does not affect the generative power of context-free grammars. Our paper proves that this is true even if all paths are restricted in this way.

After establishing the path-related results sketched above, we study restrictions placed on tree cuts, and in this way, we actually open a new investigation area concerning the subject of this paper because all the other related studies discussed the restrictions placed on paths or levels, not cuts (see [3, 5, 6, 7, 9]).

More specifically, we introduce the notion of a tree-controlled grammar in which we restrict its derivation-tree cuts by a prescribed regular language so that for each derivation tree in the grammar there is a set X of tree cuts that cover all the tree and X is described by given regular language. Then, we consider all these grammars and prove that they characterize the family of recursively enumerable languages. Finally, we introduce a binary relation over the derivation-tree cuts in these grammars and prove that the family of languages generated by them is also identical with the family of recursively enumerable languages.

In the conclusion, we formulate some open problems concerning the future investigation of grammars with restricted paths and cuts.

2. PRELIMINARIES

This paper assumes that the reader is familiar with the graph theory (see [2]) and the theory of formal languages (see [8]), including the theory of regulated rewriting (see [4]). 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 the rules of the form $p : A \to x$ where p is a unique label, $A \in V - T$, $x \in V^*$, and $S \in V - T$ is the starting symbol. A derivation step in a context-free grammar G is defined for $u, v \in V^*$ and $p : A \to x \in P$ as $uAv \Rightarrow_G uxv\,[p]$.

A context-free grammar is referred to as ε-free provided that for each rule $p : A \to x \in P$, $x \in V^+$. A context-free grammar $G = (V, T, P, S)$ is *regular*, if and only if all its rules are of the form either $A \to a$, $A \to aB$, or $A \to a$, $A \to Ba$, but not both, where $A, B \in V - T$, $a \in T$.

A *context-sensitive* grammar is a quadruple $G = (V, T, P, S)$, where V, T, and S have the same meaning as in a context-free grammar and every rule in P is of the form $y \to x$ with $|y| \leq |x|$, or $y = S$, $x = \varepsilon$. A context-sensitive grammar $G = (V, T, P, S)$ is specified in *Penttonen normal form* if every rule in P is either of the form $AB \to AC$, $A \to BC$, or $A \to x$, where $A, B, C \in V - T$, $x \in T$. A derivation step in a context-sensitive grammar G is defined for $u, v \in V^*$ and $p : y \to x \in P$ as $uyv \Rightarrow_G uxv\,[p]$.

A *general* grammar is a quadruple $G = (V, T, P, S)$, where V, T, and S have the same meaning as in a context-free grammar and every rule in P is of the form $y \to x$ with $x \in V^*(V - T)V^*$, $y \in V^*$. A general grammar $G = (V, T, P, S)$ is specified in *Penttonen normal form* if every rule in P is either of the form $AB \to AC$, $A \to BC$, $A \to x$, or $A \to \varepsilon$, where $A, B, C \in V - T$, $x \in T$. A derivation step in a general grammar G is defined for $u, v \in V^*$ and $p : y \to x \in P$ as $uyv \Rightarrow_G uxv\,[p]$.

In the standard manner, for context-free grammar, context-sensitive grammar, and general grammar $G = (V, T, P, S)$, we introduce the relations $\Rightarrow^i_G$, $\Rightarrow^+_G$, and $\Rightarrow^*_G$ (see [8]). The language of a context-free grammar, a context-sensitive grammar, or a general grammar G is defined as $L(G) = \{x \in T^* : S \Rightarrow^*_G x\}$.

The family of regular languages, context-free languages, context-sensitive languages, and recursively enumerable languages is denoted by **REG**, **CF**, **CS**, and **RE**, respectively.

Let $G = (V, T, P, S)$ be a context-free grammar. Let ${}_G\Delta(x)$ denote the set of the derivation trees of x in G with $x \in V^*$. Let $t \in {}_G\Delta(x)$ be a *derivation tree*, then

- $root(t)$ denote the root node of t;
- a *level* of t is any sequence of all nodes with the same distance from the root of t;
- a *path* of t is any sequence of nodes with the first node equal to the root of t, last node equal to a leaf of t, and there is an edge in t between each two consecutive nodes of the sequence;
- a *cut* c of t is any sequence of nodes such that each path of t has precisely one node in c (see [1, chap. 2.4.1]).

Let $word(s)$ denote the string obtained by concatenating all symbols of the sequence of nodes s in a derivation tree.

Let $r : A \to B_1 \dots B_n \in P$. The rule tree corresponding to r is the derivation tree r_Δ of height 1 with $root(r_\Delta) = A$ and the sequence of leaves $B_1 \dots B_n$.

A *finite automaton* is a 5-tuple $M = (Q, V, R, s, F)$, where Q is a finite set of states, V is an input alphabet, R is a finite set of moves of the form $pa \to q$ with $p, q \in Q$, $a \in V \cup \{\varepsilon\}$, $s \in Q$ is start state, $F \subseteq Q$ is a set of final states. A *configuration of* M is a string $\chi \in QV^*$.

Let pax and qx be two configurations of M, where $p, q \in Q$, $a \in V^* \cup \{\varepsilon\}$, and $x \in V^*$. Let $r = pa \to q \in R$. The move of M, denoted by $\vdash$, from pax to qx according to r is defined as $pax \vdash qx\,[r]$.

A finite automaton $M = (Q, V, R, s, F)$ is referred to as *deterministic* provided

1. R contains no moves of the form $p \to q$ (ε-moves), and
2. for each $pa \to q \in R$, $R - \{pa \to q\}$ contains no rule of the form $pa \to q'$ for some $q' \in Q$.

In the standard manner, we introduce the relations $\vdash^n$, $\vdash^+$, and $\vdash^*$ (see [8]). The language accepted by a finite automaton is defined as $L(M) = \{w : w \in V^*, sw \vdash^* f, f \in F\}$.

3. DEFINITIONS

First, using the terminology of the previous section, we define the basic notions given in [3] and [6]. Then, we introduce new derivation-tree-based restrictions of tree-controlled grammars – the subject of investigation in this paper.

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.

Let (G, R) be a *TC* grammar.

(1) The *language that* (G, R) *generates under the levels control by* R is denoted by ${}_{level}L(G, R)$ and defined by the following equivalence:

For all $x \in T^*$, $x \in {}_{level}L(G, R)$ if and only if there is a derivation tree $t \in {}_G\Delta(x)$ such that for all levels s of t (except the last one), $word(s) \in R$.

This kind of derivation in *TC* grammars is studied in [3, 5], and [9].

(2) The *language that* (G, R) *generates under the path control by* R is denoted by ${}_{path}L(G, R)$ and defined by the following equivalence:

For all $x \in T^*$, $x \in {}_{path}L(G, R)$ if and only if there is a derivation tree $t \in {}_G\Delta(x)$ such that there is path p of t with $word(p) \in R$.

This kind of derivation in *TC* grammars is studied in [6] and [7].

(3) The *language that* (G, R) *generates under the all-path control by* R is denoted by ${}_{all\text{-}path}L(G, R)$ and defined by the following equivalence:

For all $x \in T^*$, $x \in {}_{all\text{-}path}L(G, R)$ if and only if there is a derivation tree $t \in {}_G\Delta(x)$ such that for all paths s of t, $word(s) \in R$.

(4) The *language that* (G, R) *generates under the cuts control by* R is denoted by ${}_{cut}L(G, R)$ and defined by the following equivalence:

For all $x \in T^*$, $x \in {}_{cut}L(G, R)$ if and only if there is a derivation tree $t \in {}_G\Delta(x)$ and a set ${}_xM$ of its cuts such that

1. for each $c \in {}_xM$, $word(c) \in R$, and
2. ${}_xM$ covers the whole t.

In other words, (1.) states that ${}_xM$ contains only those cuts, which are described by R and the meaning of (2.) is that if n is a node of t, then there is $c \in {}_xM$ such that c contains n.

(5) Let $\preceq$ be a binary relation over a sequence ${}_xM$ of the cuts such that for each two cuts $c_1, c_2 \in {}_xM$, $c_1 \preceq c_2$ if and only if for each node n_2 of c_2

- either there is a node n_1 of c_1 such that n_2 is a son of n_1,
- or $n_1 = n_2$.

In other words, $n_1 \neq n_2$ implies n_2 is a son of n_1.

The *language that* (G, R) *generates under the ordered-cuts control by* R is denoted by ${}_{ord\text{-}cut}L(G, R)$ and defined by this equivalence:

For all $x \in T^*$, $x \in {}_{ord\text{-}cut}L(G, R)$ if and only if there is a derivation tree $t \in {}_G\Delta(x)$ and a sequence $c_{1x}, c_{2x}, \ldots, c_{n_x}$ of the cuts of t, for some $n_x \geq 1$, such that

1. for all $i = 1_x, 2_x, \ldots, n_x$, $word(c_i) \in R$,
2. $\{c_{1x}, c_{2x}, \ldots, c_{nx}\}$ covers the whole t, and
3. $c_{i_x} \preceq c_{(i+1)_x}$ for all $i = 1, 2, \ldots, n-1$.

In other words, (1.) states that a sequence of the cuts contains only those cuts, which are described by R, (2.) says that the set defined by a sequence of the cuts covers the whole t, and the meaning of (3.) is that the cuts in a sequence do not cross, although they can have some common nodes.

Let $X \in \{level, path, all\text{-}path, cut, ord\text{-}cut\}$ then **X-TC** $= \{{}_XL(G,R) | \ (G,R)$ is a TC grammar$\}$ and $\mathbf{X\text{-}TC}_{\varepsilon\text{-}free} = \{{}_XL(G,R) | \ (G,R)$ is a TC grammar in which G is ε-free$\}$.

Just like for ordinary context-free grammars, we can introduce ${}_{(G,R)}\Delta(x)$ for TC grammars, which represents a straightforward task left to the reader. Let (G,R) be a TC grammar where $G = (V,T,P,S)$, then ${}_{(G,R)}\Delta(x)$, $x \in V^*$, denotes the set of derivation trees with frontier x in G.

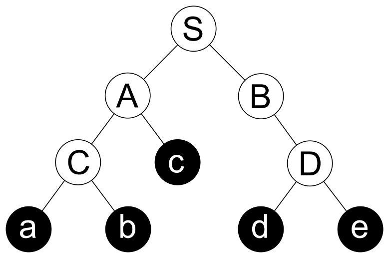

Fig. 1. An illustration of derivation-tree-based restrictions.

To illustrate (1) through (5) above, suppose that in a TC grammar (G,R), there is a derivation tree given in Figure 1, where *abcde* is a terminal string.

- In (1), to have *abcde* in ${}_{level}L(G,R)$, the strings S, AB, CcD have to be in R.
- In (2), to have *abcde* in ${}_{path}L(G,R)$, at least one of the strings $SACa$, $SACb$, SAc, $SBDd$, $SBDe$ has to be in R.
- In (3), to have *abcde* in ${}_{all\text{-}path}L(G,R)$, the strings $SACa$, $SACb$, SAc, $SBDd$, $SBDe$ have to be in R.
- In (4), to have *abcde* in ${}_{cut}L(G,R)$, for example the set ${}_xM = \{S, AD, CcB, abcB, Ade\}$ with $word(s) \in R$, for all $s \in {}_xM$, is correct. Note that, however, ${}_xM$ is not correct in terms of (5) since the cuts cross each other in ${}_xM$.
- In (5), to have *abcde* in ${}_{ord\text{-}cut}L(G,R)$, for example the sequence ${}_xM = S$, AB, CcB, $abcD$, $abcde$ with $word(s) \in R$, for each item s of ${}_xM$, is correct, since the cuts do not cross.

4. RESULTS

In this section, we prove that **CF** = **all-path-TC** and **RE** = **ord-cut-TC** = **cut-TC**.

Theorem 4.1. CF = **all-path-TC**

Proof. Let L be an **all-path-TC** language. We assume that L is generated by a TC grammar (G, R), where $G = (V, T, P, S)$ is a context-free grammar, R is a regular language over V, and (G, R) generates the language ${}_{all\text{-}path}L(G, R)$.

Next, we assume R is accepted by a deterministic finite automaton $M = (Q_M, V, R_M, s_M, F_M)$. Since the paths of a derivation tree of a context-free grammar are of the form xb with $x \in (V - T)^+, b \in T$, we assume that each $r \in R_M$ is of the form $pa \to q$ with either (a) $a \in V - T$ and $q \notin F_M$, or (b) $a \in T$ and $q \in F_M$.

Let G' be a context-free grammar $G' = (V', T, P', S')$, where $V' = Q \cup T$, $Q = \{\langle A, q_A\rangle : A \in V,\ q_A \in Q_M,\ qA \to q_A \in R_M$ for some $q \in Q_M\}$, $S' = \langle S, s_S\rangle, s_M S \to s_S \in R_M$, and P' is defined in the following way:

If

(1) $A \to B_1 B_2 \dots B_n \in P$, $n \geq 1$;

(2) $qA \to q_A \in R_M$, for some $q \in Q_M$;

(3) $q_A B_i \to q_{B_i} \in R_M$, for each B_i, $i = 1, 2, \dots, n$;

then add $\langle A, q_A\rangle \to \overline{B_1}\overline{B_2} \dots \overline{B_n}$ to P', where, for $i = 1, 2, \dots, n$,

- if $B_i \in V - T$, then $\overline{B_i} = \langle B_i, q_{B_i}\rangle$ with $q_A B_i \to q_{B_i} \in R_M$,
- if $B_i \in T$, then $\overline{B_i} = B_i$.

Without any loss of generality, we assume that $V \cap Q_M = \emptyset$. We define the function g from ${}_{G'}\Delta(y)$, $y \in (V')^*$, into ${}_{(G,R)}\Delta(x)$, $x \in V^*$, as

for all nodes labeled by $a \in T$, $g(a) = a$;
for all nodes labeled by $\langle A, q\rangle \in Q$, $g(\langle A, q\rangle) = A$.

To show that **all-path-TC** $\subseteq$ **CF**, we first prove the next claim.

Claim: $t \in {}_{(G,R)}\Delta(x)$, $x \in V^*$, if and only if $d \in {}_{G'}\Delta(y)$, $y \in (V')^*$, such that $g(d) = t$.

Only-If Part: That is, if $t \in {}_{(G,R)}\Delta(x)$, $x \in V^*$, then $d \in {}_{G'}\Delta(y)$, $y \in (V')^*$, such that $g(d) = t$. This is established by induction on the number of the rule trees, denoted by m, in $t \in {}_{(G,R)}\Delta(x)$, $x \in V^*$.

Basis: Let $m = 0$. Since $m = 0$ implies the zero-length derivation, the only rule tree in t contains only the node S that corresponds to the starting symbol of G. Clearly, the only rule tree in d is the node that corresponds to the starting symbol of G' – that is, $\langle S, s_S\rangle$ with $g(\langle S, s_S\rangle) = S$.

Induction Hypothesis: Suppose that the only-if part holds for all $t \in {}_{(G,R)}\Delta(uvw)$, $u, v, w \in V^*$, that contains m or fewer rule trees, for some $m \geq 0$.

Induction Step: Consider any $t \in {}_{(G,R)}\Delta(uvw)$ that contains $m+1$ rule trees. Clearly, there is some subtree $t' \in {}_{(G,R)}\Delta(v)$ of t such that t' is a rule tree.

Next, we remove just one rule tree from t. If $root(t') = B$, then there is $t'' \in {}_{(G,R)}\Delta(uBw)$, where $u, w \in V^*, B \in V - T$, that contains m rule trees. There is also $r : B \to v \in P$ (and its rule tree r_Δ) and t'' is a subtree of t. Hence, by the induction hypothesis, there is also $d'' \in {}_{G'}\Delta(y)$ such that $g(d'') = t''$.

Since $uBw \Rightarrow uvw\,[r]$ in G, $q_B B_i \to q_{B_i} \in R_M$, for each B_i in v, $i = 1, 2, \ldots, |v|$. Therefore, there is $r' \in P'$ (and its rule tree r'_Δ) such that $g(r'_\Delta) = r_\Delta$. Thus, we obtain $d \in {}_{G'}\Delta(y)$ with $g(d) = t$.

If Part: That is, if $d \in {}_{G'}\Delta(y)$, $y \in (V')^*$, then $t \in {}_{(G,R)}\Delta(x)$, $x \in V^*$, such that $g(d) = t$. This is established by induction on the number of the rule trees, denoted by j, in $d \in {}_{G'}\Delta(y)$, $y \in (V')^*$.

Basis: Let $j = 0$. Since $j = 0$ implies the zero-length derivation, the only rule tree in d contains only the node $\langle S, s_S\rangle$ that corresponds to the starting symbol of G'. Clearly, the only rule tree in t contains only the node S that corresponds to the starting symbol of (G, R) and $g(\langle S, s_S\rangle) = S$.

Induction Hypothesis: Suppose that the if part holds for all $d \in {}_{G'}\Delta(uvw)$, $u, v, w \in (V')^*$, that contains j or fewer rule trees, for some $j \geq 0$.

Induction Step: Consider any $d \in {}_{G'}\Delta(uvw)$ that contains $j + 1$ rule trees. Clearly, there is some subtree $d' \in {}_{G'}\Delta(v)$ of d such that d' is a rule tree.

Next, we remove just one rule tree from d – that is, if $root(d') = \langle B, q\rangle$, then there is $d'' \in {}_{G'}\Delta(u\langle B, q\rangle w)$, where $u, w \in (V')^*, \langle B, q\rangle \in V' - T$, that contains j rule trees. There is also $r : \langle B, q\rangle \to v \in P'$ (and its rule tree r_Δ) and d'' is a subtree of d. Hence, by the induction hypothesis, there is also $t'' \in {}_{(G,R)}\Delta(x)$ such that $g(d'') = t''$.

Since $u\langle B, q\rangle w \Rightarrow uvw\,[r]$ in G', there is $r' \in P$ (and its rule tree r'_Δ) such that $g(r'_\Delta) = r_\Delta$. For each $\langle B_i, q_{B_i}\rangle$ in v, there is some $q \in Q_M$ such that $qB_i \to q_{B_i} \in R_M, i = 1, 2, \ldots, |v|$. Thus, we obtain $t \in {}_{(G,R)}\Delta(x)$ with $g(d) = t$.

We can now easily obtain **all-path-TC** $\subseteq$ **CF** as follows.

- Let $t \in {}_{(G,R)}\Delta(x)$, with $x \in T^*$. Clearly $x \in L(G, R)$, there is $d \in {}_{G'}\Delta(y)$ such that $g(d) = t$, and $x = y \in L(G')$. Thus $L(G, R) \subseteq L(G')$.
- Let $d \in {}_{G'}\Delta(y)$, with $y \in T^*$. Clearly $y \in L(G)$, there is $t \in {}_{(G,R)}\Delta(x)$ such that $g(d) = t$, and $y = x \in L(G, R)$. Thus, $L(G') \subseteq L(G, R)$.

Therefore, $L(G, R) = L(G')$ and thus **all-path-TC** $\subseteq$ **CF**.

Let L be a context-free language. Without any loss of generality, we assume that L is generated by a context-free grammar $G = (V, T, P, S)$. Let (G', R) be a TC grammar that generates ${}_{all\text{-}path}L(G', R) \in$ **all-path-TC**, where $G' = G$, $R = (V-T)^{+}T$. Clearly $L(G) = L(G', R)$, therefore **CF** $\subseteq$ **all-path-TC**. Thus, **all-path-TC** $=$ **CF**. □

Next, we prove **RE** $=$ **ord-cut-TC** $=$ **cut-TC**.

Theorem 4.2. RE $=$ **ord-cut-TC**

Proof. Let L be a recursively enumerable language. Without any loss of generality, we assume that L is generated by a general grammar $G = (V, T, P, S)$ in Pentonnen normal form (see Section 2). Let (G', R) be a TC grammar that generates ${}_{ord\text{-}cut}L(G', R) \in$ **ord-cut-TC**, where $G' = (V', T, P', S)$, $V' = V \cup Q$, $Q = \{\langle A, B, C\rangle : AB \to AC \in P\}$, and P' is defined in the following way:

1. for $A \to x \in P$, $A \in V - T$, $x \in \{\varepsilon\} \cup T \cup (V-T)^2$, add $A \to x$ to P';
2. for $AB \to AC \in P$, $A, B, C \in (V - T)$, add the set of two rules $\{B \to \langle A, B, C\rangle$, $\langle A, B, C\rangle \to C\}$ to P'.

Without any loss of generality, we assume that $Q \cap V = \emptyset$. The regular language R is defined as follows:

$$R = V^* \cup \{V^* A\langle A, B, C\rangle V^* : AB \to AC \in P, A \in V - T, \langle A, B, C\rangle \in Q\}.$$

We define the function h from $(V')^*$ into V^* by:

for all $C \in V$, $h(C) = C$,
for all $\langle A, B, C\rangle \in Q$, $h(\langle A, B, C\rangle) = C$.

To show that $L(G) = L(G', R)$, we first prove the next claim.

Claim. $S \Rightarrow^m w$, $w \in V^*$, in G, if and only if $S \Rightarrow^n v$, $v \in (V')^*$, in (G', R), where $w = h(v)$, $v \in R$, for $m, n \geq 0$.

Only-If Part: That is, if $S \Rightarrow^m w$, $w \in V^*$, in G, then $S \Rightarrow^* v$, $v \in (V')^*$, in (G', R), where $w = h(v)$, $v \in R$, for $m \geq 0$. This is established by induction on $m \geq 0$.

Basis: Let $m = 0$. The only w is S since $S \Rightarrow^0 S$ in G. Clearly, $S \Rightarrow^0 S$ in (G', R) with $S = h(S)$, and since $S \in V^*$, $S \in R$.

Induction Hypothesis: Let us suppose that the only-if part holds for all derivations of length m or less, for some $m \geq 0$.

Induction Step: Consider a derivation $S \Rightarrow^{m+1} x$ in G, $x \in V^*$. Since $m + 1 \geq 1$, there is some $y \in V^+$ and $p \in P$ such that $S \Rightarrow^m y \Rightarrow x\,[p]$ in G, and by the induction hypothesis, $S \Rightarrow^* y'$, $y' \in (V')^*$, in (G', R) with $h(y') = y$ and $y' \in R$. Next, as far as p is concerned, we distinguish two cases:

(1) p is of the form $AB \to AC$, $A, B, C \in V - T$,

(2) p is of the form $A \to \alpha$, $A \in V - T$, $\alpha \in \{\varepsilon\} \cup T \cup (V-T)^2$.

Let us discuss (1) through (2) in detail.

(1) Let p be of the form $AB \to AC$, $A, B, C \in V - T$. Then, $y' = y_1ABy_2 \in R$, $y_1, y_2 \in (V')^*$, and $B \to \langle A, B, C \rangle \in P'$ is applied in (G', R). Thus, we obtain $x' = y_1A\langle A, B, C\rangle y_2$, with $h(x') = x$ and since $x' \in V^*A\langle A, B, C\rangle V^*$, $x' \in R$. For each $\langle A, B, C\rangle \in Q$, there is $\langle A, B, C\rangle \to C \in P'$ with $h(\langle A, B, C\rangle) = h(C) = C$. Thus, $x' \Rightarrow z'$ with $h(z') = h(x') = x$, and since $z' \in V^*$, $z' \in R$.

(2) Let p be of the form $A \to \alpha$, $A \in V - T$, $\alpha \in \{\varepsilon\} \cup T \cup (V - T)^2$. Then, $y' = y_1Ay_2 \in R$, $y_1, y_2 \in (V')^*$, and $A \to \alpha \in P'$ is applied in (G', R). Thus, we obtain $x' = y_1\alpha y_2$ with $h(x') = x$, and since $x' \in V^*$, $x' \in R$.

Observe that (1) through (2) cover all possible forms of p so that the only-if part holds true.

If Part: That is, if $S \Rightarrow^n v$, $v \in (V')^*$, in (G', R), then $S \Rightarrow^* w$, $w \in V^*$, in G where $w = h(v)$, $v \in R$, for $n \geq 0$. This is established by induction on $n \geq 0$.

Basis: For $n = 0$, the only v is S since $S \Rightarrow^0 S$ in (G', R), with $h(S) = S$ and since $S \in V^*$, $S \in R$. Clearly, $S \Rightarrow^0 S$ in G.

Induction Hypothesis: Let us suppose that the if part holds for all derivations of length n or less, for some $n \geq 0$.

Induction Step: Consider a derivation of the form $S \Rightarrow^{n+1} x'$ in (G', R), where $x' \in (V')^*$. Since $n+1 \geq 1$, there is some $y' \in V^+$ such that $S \Rightarrow^n y' \Rightarrow x'$ $[p]$ in (G', R) and $y' \in R$, and by the induction hypothesis, $S \Rightarrow^* y$ in G with $h(y') = y$. Next, as far as p is concerned, we distinguish three cases:

(1) p is of the form $B \to \langle A, B, C\rangle$, $B \in V'$, $\langle A, B, C\rangle \in Q$,

(2) p is of the form $\langle A, B, C\rangle \to C$, $\langle A, B, C\rangle \in Q$, $C \in V'$,

(3) p is of the form $A \to \alpha$, $A \in V'$, $\alpha \in \{\varepsilon\} \cup T \cup (V - T)^2$.

Let us discuss (1) through (3) in detail.

(1) Let p be of the form $B \to \langle A, B, C\rangle$, $B \in V'$, $\langle A, B, C\rangle \in Q$. Then, $y = y_1By_2$ and since $x' \in R$, y_1 is of the form $y_1 = z_1A$, for some $z_1 \in V^*$. Thus, $p : AB \to AC \in P$ is applied in G and $x = y_1Cy_2$ with $h(x') = x$, $y_1 = z_1A$.

(2) Let p be of the form $\langle A, B, C\rangle \to C$, $\langle A, B, C\rangle \in Q$, $C \in V'$. Then, $y = y_1Cy_2 = x$ with $h(x') = x$.

(3) Let p be of the form $A \to \alpha$, $A \in V'$, $\alpha \in \{\varepsilon\} \cup T \cup (V - T)^2$. Then, $y = y_1Ay_2$ and $p : A \to \alpha$ is applied in G. Thus, $x = y_1\alpha y_2$ with $h(x') = x$.

Observe that (1) through (3) cover all possible forms of p so that the if part holds true.

The proof of the inclusion **RE** $\subseteq$ **ord-cut-TC** can be easily obtained from the claim above. From the definition of a cut, the following properties straightforwardly follow:

(1) Every sentential form is a special case of a cut. Therefore, let ${}_wM$ be the sequence of all sentential forms corresponding to the derivation tree of any $w \in L(G)$.

(2) ${}_wM$ covers each node of the derivation tree of w in G at least once. Thus, ${}_wM$ covers the derivation tree of any $w \in L(G)$.

(3) Considering the order of sentential forms of $w \in L(G)$ in the derivation $S \Rightarrow^* w$ in G, ${}_xM$ satisfies condition 3. stated in the definition of ${}_{\textit{ord-cut}}L(G, R)$.

Thus, $S \Rightarrow^* w$ in G if and only if $S \Rightarrow^* w$ in (G', R), $w \in T^*$. Therefore, $L(G) = {}_{\textit{ord-cut}}L(G', R)$ and consequently **RE** $\subseteq$ **ord-cut-TC**.

Clearly, **ord-cut-TC** $\subseteq$ **RE** and, therefore, **RE** $=$ **ord-cut-TC**. □

Theorem 4.3. RE = cut-TC

Proof. Clearly, **cut-TC** $\subseteq$ **RE**. Obviously, by the definitions (4) and (5) in the previous section, **ord-cut-TC** $\subseteq$ **cut-TC**. Thus **RE** $\subseteq$ **cut-TC** follows from **RE** $\subseteq$ **ord-cut-TC** (see Theorem 4.2). Therefore, **RE** $=$ **cut-TC**. □

Corollary 4.4. ord-cut-TC = cut-TC

Proof. This corollary straightforwardly follows from **RE** $=$ **ord-cut-TC** (Theorem 4.2) and **RE** $=$ **cut-TC** (Theorem 4.3). □

5. CONCLUSION

In this concluding section, we summarize the achieved results and point out some important open questions.

As a generalization of *TC* grammars that generate the language under path-based control introduced in [6], we have proved that the generative power of context-free grammars remains unchanged even if we restrict all paths in their derivation trees by regular languages.

Then, we have introduced two types of cut-based restrictions on the derivation trees of context-free grammars, and we have proved that both of them increase the generative power of context-free grammars so they characterize **RE**.

A crucially important open problem area consists of the determination of the generative power of these grammars without ε-rules. In other words, future investigations concerning the subject of this paper should try to place **ord-cut-TC**$_{\varepsilon\text{-}free}$ and **cut-TC**$_{\varepsilon\text{-}free}$ into the relation with some other well-known language families, such as **CS**.

Obviously, **CS** $\subseteq$ **ord-cut-TC**$_{\varepsilon\text{-}free}$ can be established by analogy with demonstrating **RE** $\subseteq$ **ord-cut-TC** in the proof of Theorem 4.2, in which covering the whole derivation tree by sentential forms is considered. Indeed, the only difference between Penttonen normal form for general grammars and context-sensitive grammars is that in the former, the rules of the form $A \rightarrow \varepsilon$ are allowed; while in the later, they are not. An open problem is whether **ord-cut-TC**$_{\varepsilon\text{-}free}$ $\subseteq$ **CS** holds, which would mean **ord-cut-TC**$_{\varepsilon\text{-}free}$ $=$ **CS**.

Clearly, **CS** $\subseteq$ **cut-TC**$_{\varepsilon\text{-}free}$ can be demonstrated similarly as establishing **RE** $\subseteq$ **cut-TC** in the proof of Theorem 4.3, in which **ord-cut-TC** $\subseteq$ **cut-TC** is considered. Obviously, based upon a similar argument, **ord-cut-TC**$_{\varepsilon\text{-}free}$ $\subseteq$ **cut-TC**$_{\varepsilon\text{-}free}$ holds. An open problem is whether **cut-TC**$_{\varepsilon\text{-}free}$ $\subseteq$ **CS** holds, which would imply **cut-TC**$_{\varepsilon\text{-}free}$ = **CS**.

ACKNOWLEDGEMENT

The authors thank the anonymous referee of an earlier version of our paper for invaluably useful comments, especially for the suggestion to consider ε-rules in cut-restricted TC grammars.

This work was supported by the research plan MSM0021630528, the BUT FIT grant FIT-S-11-2, MSM grant MEB041003 of the Ministry of Education, Youth and Sports of the Czech Republic, and GA ČR grant GD102/09/H042.

(Received July 23, 2010)

REFERENCES

[1] A. V. Aho and J. D. Ullman: The theory of parsing, translation, and compiling. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1972.

[2] A. Bondy: Graph Theory. Springer, 2010.

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

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

[5] J. Dassow and B. Truthe: Subregularly tree controlled grammars and languages. In: Automata and Formal Languages – 12th Int. Conference AFL 2008, Proceedings (E. Csuhaj-Varjú, Z. Ésik, eds.), Balatonfüred, Hungary 2008, pp. 158–169.

[6] S. Marcus, C. Martín-Vide, V. Mitrana, and Gh. Păun: A new-old class of linguistically motivated regulated grammars. In: Computational Linguistics in the Netherlands 2000, Selected Papers from the Eleventh clin Meeting (W. Daelemans, K. Sima'an, J. Veenstra, J. Zavrel, eds.), Rodopi, Amsterdam 2001, pp. 111-125.

[7] C. Martín-Vide and V. Mitrana: Further properties of path-controlled grammars. In: Proceedings of FG-MoL 2005, The 10th Conference on Formal Grammar and The 9th Meeting on Mathematics of Language, Edinburgh 2005, pp. 221–232.

[8] A. Meduna: Automata and Languages: Theory and Applications. Springer Verlag, 2005.

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

Jiří Koutný, DIFS FIT BUT, Božetěchova 2, 612 66 Brno. Czech Republic.
e-mail: ikoutny@fit.vutbr.cz

Alexander Meduna, DIFS FIT BUT, Božetěchova 2, 612 66 Brno. Czech Republic.
e-mail: meduna@fit.vutbr.cz