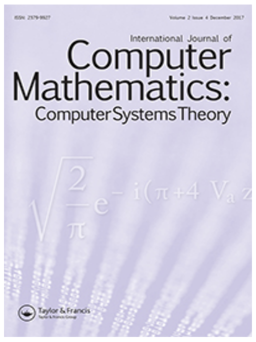

International Journal of Computer Mathematics: Computer Systems Theory

ISSN: 2379-9927 (Print) 2379-9935 (Online) Journal homepage: http://www.tandfonline.com/loi/tcom20

On tree-restricted regular-controlled context-free grammars

A. Meduna, O. Soukup & E. Csuhaj-Varjú

To cite this article: A. Meduna, O. Soukup & E. Csuhaj-Varjú (2017) On tree-restricted regular-controlled context-free grammars, International Journal of Computer Mathematics: Computer Systems Theory, 2:4, 147-163, DOI: 10.1080/23799927.2017.1388291

To link to this article: http://dx.doi.org/10.1080/23799927.2017.1388291

Accepted author version posted online: 09 Oct 2017.
Published online: 27 Oct 2017.

INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS: COMPUTER SYSTEMS THEORY, 2017
VOL. 2, NO. 4, 147–163
https://doi.org/10.1080/23799927.2017.1388291

On tree-restricted regular-controlled context-free grammars

A. Meduna[a], O. Soukup[a] and E. Csuhaj-Varjú[b]

[a]Brno University of Technology, Faculty of Information Technology, IT4Innovations Centre of Excellence, Brno, Czech Republic; [b]Department of Algorithms And Their Applications, Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary

ABSTRACT
This paper gives simple tree-based conditions under which regular-controlled context-free grammars generate context-free languages of finite index, so they cannot even generate all context-free languages. It defines the notion of path-changing derivation step which corresponds to performing two consecutive rewritings of nonterminal symbols present in the different branches of the derivation tree. It proves that if there exists a certain constant that limits the number of path-changing derivation steps, then, the regular-controlled grammar generates a context-free language of finite index. At the end, we generalize achieved result and provide some open problems for future study.

ARTICLE HISTORY
Received 18 January 2017
Revised 9 September 2017
Accepted 25 September 2017

KEYWORDS
Regular-controlled context-free grammars; restricted derivation trees; context-freeness of finite index

1. Introduction

The theory of formal languages has always struggled to establish conditions under which some grammars decrease their power so they characterize a well-known language family, properly contained in the language family generated by unrestricted versions of these grammars. This struggle comes as no surprise because conditions like this often significantly simplify proofs that some languages are members of the well-known language family in question. For instance, consider the famous workspace theorem for general grammars, which fulfils a crucially important role in the grammatically oriented theory of formal languages as a whole (see Theorem III.10.1 in [14]). This theorem represents a powerful tool to demonstrate that if a general grammar H generates each of its sentences by a derivation satisfying a prescribed condition (specifically, this condition requires that there is a positive integer k such that H generates every sentence y in the generated language $L(H)$ by a derivation in which every sentential form x satisfies $|x| \leq k|y|$), then $L(H)$ is a member of the context-sensitive language family. As a result, establishing conditions of this kind really represents a long-lasting and useful trend in formal language theory.

The present paper contributes to this trend by establishing conditions in terms of context-free grammars with derivations controlled by regular languages so the resulting conditions allow us to demonstrate that a language is a member of the family of context-free languages of finite index. To give an insight into this result, recall that context-free grammars whose derivations are controlled by regular languages [7] are stronger than their ordinary uncontrolled versions. In fact, regular-controlled grammars (described in Definition 3.8) are as powerful as matrix grammars – that is, they generate the family of matrix languages [10]. Surprisingly, the present paper demonstrates that under some very natural and simple conditions placed upon their derivation trees, this power significantly

CONTACT O. Soukup isoukup@fit.vutbr.cz Brno University of Technology, Faculty of Information Technology, IT4Innovations Centre of Excellence, Božetěchova 1/2, 612 66 Brno, Czech Republic

lessens. As a matter of fact, under the conditions, they generate only a certain subfamily of the family of context-free languages – context-free languages of finite index.

To give an insight into the tree-based conditions, consider a context-free grammar G with the following rules;

$$A \to BC, \quad B \to X, \quad C \to Y$$

and the next derivation

$$A \Rightarrow BC \Rightarrow XC \Rightarrow XY.$$

Let us look at the process of the derivation from the perspective of the derivation tree. During the first step, a branching node is introduced. Then, the derivation continues into the left branch. However, the last derivation step takes place within the different branch; which is the right one. We call this phenomenon path-change and discuss it in the terms of regular-controlled grammars.

In essence, we put restrictions on the number of path-changing derivation steps. It is proved that the language generated by a regular-controlled grammar is context-free of index k if there is a constant k such that every sentence w in the generated language is the frontier of a derivation tree corresponding to some derivation within which there were k or fewer path-changing derivation steps. Of course, this k is an upper bound and the minimal index is possibly lower.

Since the tree-based restrictions are independent of the control mechanism, the achieved result holds for well-known matrix grammars [1] as well.

The paper is organized as follows. First, Sections 2 and 3 give all the necessary terminology. Then, Section 4 establishes the main result of this paper concerning regular-controlled grammars. Finally, we generalize the result for other types of grammars and state some open problem areas for the future study.

2. Preliminaries

We assume that the reader is familiar with discrete mathematics, including graph theory [2, 4, 5] as well as formal language theory [9, 12, 14].

A *directed graph* G is a pair $G = (V, E)$, where V is a finite *set of nodes*, and $E \subseteq V \times V$ is a finite *set of edges*. For a node $v \in V$, the number of edges of the form $(x, v) \in E$, $x \in V$, is called an *in-degree* of v and denoted by in-d(v). For a node $v \in V$, the number of edges of the form $(v, x) \in E$, $x \in V$, is called an *out-degree* of v and denoted by out-d(v). Let $(v_0, v_1, \dots, v_n)$ be an n-tuple of nodes, for some $n \geq 0$, where $v_i \in V$, for $0 \leq i \leq n$, and there exists an edge $(v_k, v_{k+1}) \in E$, for every pair of nodes v_k,v_{k+1}, where $0 \leq k \leq n-1$, then, we call it a *sequence of the length n*. Let $(v_0, v_1, \dots, v_n)$ be a sequence of the length n, for some $n \geq 0$, where $v_i \neq v_j$, for $0 \leq i \leq n$, $0 \leq j \leq n$, $i \neq j$, then, we call the sequence a *path*. Let $(v_0, v_1, \dots, v_n)$ be a path in G, for some $n \geq 0$, except that $v_0 = v_n$, then, we call it a *cycle*. A graph G is *acyclic* iff it contains no cycle.

For a set W, card(W) denotes its *cardinality*. An *alphabet* is a finite nonempty set – elements are called *symbols*. Let V be an *alphabet*. V^* is the *set of all strings* over V. Algebraically, V^* represents the free monoid generated by V under the operation of concatenation. The identity of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$. Algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. For $w \in V^*$, $a \in V$, and $A \subseteq V$, $|w|$ denotes the *length of* w, $\#_a(w)$ denotes the *number of occurrences of the symbol a in w*, and $\#_A(w)$ denotes the *number of occurrences of the symbols from A in w*. The *alphabet of* w, denoted by alph(w), is the set of symbols appearing in w. For $u, v \in V^*$, $\bullet(u, v)$ denotes the *shuffle of the strings u and v* and is defined as $\bullet(u, v) = u_1v_1u_2v_2 \cdots u_kv_k$, $u_i, v_i \in V^*$, $1 \leq i \leq k$, for some $k \geq 0$, where $u_1u_2 \cdots u_k = u$ and $v_1v_2 \cdots v_k = v$.

Let $\Rightarrow$ be a relation over V^*. Define the ith power of $\Rightarrow$ as $\Rightarrow^i$, for $i \geq 0$. The transitive and the transitive-reflexive closure of $\Rightarrow$ are denoted by $\Rightarrow^+$ and $\Rightarrow^*$, respectively. Unless we explicitly stated otherwise, we write $x \Rightarrow y$ instead of $(x, y) \in \Rightarrow$ throughout.

The families of regular, context-free and context-sensitive languages are denoted by **REG**, **CF** and **CS**, respectively. Later we define **REG** and **CF**.

3. Definitions and examples

Definition 3.1: An (*oriented*) tree is a directed acyclic graph $G = (V, E)$, with a specified node $\mathscr{R} \in V$ called the *root* such that in-d$(\mathscr{R}) = 0$, and for all $x \in V - \{\mathscr{R}\}$, in-d$(x) = 1$ and there exists a path $(v_0, v_1, \ldots, v_n)$, where $v_0 = \mathscr{R}$, $v_n = x$, for some $n \geq 1$. For $v, u \in V$, where $(v, u) \in E$, v is called a *parent* of u, u is called a child of v, respectively. For $v, u, z \in V$, where $(v, u), (v, z) \in E$, u is called a *sibling* of z. A node without any children is called a *leaf*.

Let $G = (V, E)$ be a tree. Define a partial order relation $<$ over V as follows. For a path $\alpha = (m_0, m_1, \ldots, m_k)$, where $m_0 = \mathscr{R}$, $m_i < m_k$, $0 \leq i \leq k-1$. Then, m_i is called a *predecessor* of m_k and m_k is called a *descendant* of m_i. Let x, y, and z be three nodes. If $x < y$ and $x < z$, we call x a *common predecessor* of y and z.

An *ordered tree* t is a tree, where for every set of siblings there exists a linear ordering. Let o has the children $n_1, n_2, \ldots, n_r$ ordered in this way, where $r \geq 1$. Then, n_1 is the *leftmost child* of o, n_r is the *rightmost child* of o and n_i is the *direct left sibling* of n_{i+1}, n_{i+1} is the *direct right sibling* of n_i, $1 \leq i \leq r-1$, and for $1 \leq j < k \leq r$, n_j is a *left sibling* of n_k and n_k is a *right sibling* of n_j.

An ordered tree is called *labelled*, if there exists a set of labels $\mathcal{L}$ and a total mapping $l : V \to \mathcal{L}$. Let t be a labelled ordered tree, then the string of labels of all leaves written in the left-to-right order is called the *frontier* of t and denoted by frontier(t). In what follows, we substitute a node of a tree by its label if there is no risk of confusion.

Definition 3.2: A *finite automaton* (an FA for short) is a quintuple $M = (Q, \Sigma, R, s, F)$, where Q is a finite set of *states*, Σ is an *input alphabet*, where $Q \cap \Sigma = \emptyset$, $R \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$ is a finite relation, called the set of *transitions*, $s \in Q$ is the *initial state*, $F \subseteq Q$ is the set of *final states*.

Instead of $(p, a, q) \in R$, we write $pa \to q \in R$. A *configuration* of M is any word from $Q\Sigma^*$. The relation of a *direct move*, denoted by $\vdash$, is defined over $Q\Sigma^*$ as follows: if pax, $qx \in Q\Sigma^*$, and $pa \to q \in R$, then $pax \vdash qx$ in M.

Let $\vdash^k$, $\vdash^*$, and $\vdash^+$ denote the kth power of $\vdash$, for some $k \geq 0$, the reflexive and transitive closure of $\vdash$, and the transitive closure of $\vdash$, respectively. The *language accepted by M* is denoted by $L(M)$ and defined as

$$L(M) = \{w \in \Sigma^* \mid sw \vdash^* f, f \in F\}.$$

As it is well known, the family of finite automata describes **REG** (see [8]).

Definition 3.3: A *context-free grammar* (a CFG for short) G is a quadruple $G = (N, T, P, S)$, where N is an alphabet of *nonterminals*, T is an alphabet of *terminals* such that $N \cap T = \emptyset$, $P \subseteq N \times (N \cup T)^*$ is a finite *set of rules*, and $S \in N$ is the *start symbol*. Instead of $p : (A, x) \in P$, where p is a unique *label*, we write $p : A \to x$. If no confusion arises, a rule and its label are interchangeable.

For every $u, v \in (N \cup T)^*$ and $p : A \to x \in P$, $\Rightarrow$ is the *direct derivation* relation over $(N \cup T)^*$ and we write $uAv \Rightarrow uxv[p]$ or simply $uAv \Rightarrow uxv$. For $n \geq 0$, $\Rightarrow^n$ denotes the nth power of $\Rightarrow$. Furthermore, $\Rightarrow^+$ and $\Rightarrow^*$ denote the transitive and the transitive-reflexive closure of $\Rightarrow$, respectively. Let $\mathscr{F}(G) = \{w \in (N \cup T)^* \mid S \Rightarrow^* w\}$ denote the *set of all sentential forms of* G. The *language of* G is $L(G) = \{w \in T^* \mid w \in \mathscr{F}(G)\}$. **CF** $= \{L(G) \mid G$ is CFG$\}$.

G is *propagating* if $A \to x \in P$ implies $x \neq \varepsilon$ or $A = S$ and S does not occur on the right-hand side of any rule in P, if $\varepsilon \in L(G)$.

Propagating CFGs characterize **CF** as well [12].

G is said to be in the *binary form* if any $p \in P$ has one of these forms,

$$A \to BC, \quad A \to x$$

where $A, B, C \in N$, $x \in (N \cup T)^*$, and $\#_N(x) \leq 1$. It can be easily shown that for any context-free grammar G there exists a context-free grammar G' in the binary form such that $L(G) = L(G')$. The proof is similar to the one that is used for constructing an equivalent Chomsky normal form grammar [3] to a context-free grammar.

G is of *index k* if for every $w \in L(G)$ there exists a derivation

$$x_0 \Rightarrow x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_n \Rightarrow w,$$

where $x_0 = S$ and $\#_N(x_i) \leq k$, for all $0 \leq i \leq n$, for some $n \geq 0$. L is a *context-free language of index k* if there exists a context-free grammar G of index k, where $L(G) = L$. The *family of all context-free languages of index k* generated by context-free grammars of index k is denoted by ${}_k\mathbf{CF}$. It was proved that families of context-free languages of finite index form an infinite hierarchy of language families above regular languages

$$\mathbf{REG} \subset {}_1\mathbf{CF} \subset {}_2\mathbf{CF} \subset {}_3\mathbf{CF} \subset \cdots \subset {}_\infty\mathbf{CF} = \mathbf{CF}.$$

Indeed, ${}_1\mathbf{CF}$ denotes the family of linear languages – and there are also context-free languages of an infinite index (for details see [6, 11, 13]).

Definition 3.4: Let t be a labelled ordered tree. A *left-bracketed representation of t* denoted by l-rep(t) can be obtained by applying the following recursive rules:

(1) If t has a root labelled $\mathscr{R}$ with subtrees $t_1, \dots, t_k$ ordered in this way, then l-rep$(t) = \mathscr{R}\langle$l-rep$(t_1), \dots,$ l-rep$(t_k)\rangle$.
(2) If t has a root labelled $\mathscr{R}$ with no direct descendants, then l-rep$(t) = \mathscr{R}$.

Definition 3.5: Let $G = (N, T, P, S)$ be a CFG in the binary form.

(1) For $p : A \to x \in P$, $A\langle x\rangle$ is the *rule tree* that represents p.
(2) The *derivation trees* representing derivations in G are defined recursively as follows:
    (a) One-node tree with a node labelled X is the derivation tree corresponding to $X \Rightarrow^0 X$ in G, where $X \in (N \cup T)$. Recall that if $X = \varepsilon$, we refer to the node labelled X as *ε-node* (*ε-leaf*); otherwise, we call it *non-ε-node* (*non-ε-leaf*).
    (b) Let d be the derivation tree representing $X \Rightarrow^* uAv[\rho]$ with frontier$(d) = uAv$, and let $p : A \to x \in P$. The derivation tree that represents

$$X \Rightarrow^* uAv[\rho] \Rightarrow uxv[p]$$

    is obtained by replacing the ith non-ε-leaf in d labelled A, with rule tree corresponding to p, denoted $A\langle x\rangle$, where $i = |uA|$.
(3) A *derivation tree* in G is any tree t for which there is a derivation represented by t (see item (2) in this definition).

For any $A \Rightarrow^* x\,[\rho]$ in G, where $A \in N$, $x \in V^*$, and $\rho \in P^*$, ${}_G\triangle(A \Rightarrow^* x[\rho])$ denotes the derivation tree corresponding to $A \Rightarrow^* x[\rho]$. Just like we often write $A \Rightarrow^* x$ instead of $A \Rightarrow^* x\,[\rho]$, we sometimes simplify ${}_G\triangle(A \Rightarrow^* x\,[\rho])$ to ${}_G\triangle(A \Rightarrow^* x)$ if there is no risk of confusion. Let ${}_G\blacktriangle$ denote the set of all derivation trees in G. Finally, by ${}_G\triangle_x \in {}_G\blacktriangle$, we mean a derivation tree whose frontier is x, where $x \in \mathscr{F}(G)$.

If a node is labelled with a terminal, it is called a *terminal node*. If a node is labelled with a nonterminal, it is called a *nonterminal node*. If a leaf node is labelled by ε, it is called an *ε-node* and denotes an erasion of a nonterminal. If a node has more than one nonterminal child, it is called a *branching node*; otherwise, it is a *non-branching node*.

Consider a derivation in G of length $n \geq 2$, $S \Rightarrow^n w$. If the derivation is performed as

$$S \Rightarrow^{n-2} uAv \Rightarrow uxByv \Rightarrow uxzyv,$$

where $uxzyv = w$, for some $u, v, w, x, y, z \in (N \cup T)^*$ and $A, B \in N$, we call the nth step of the derivation *path-preserving* – indeed, in a resulting derivation tree the nodes corresponding to consecutively rewritten nonterminals A and B belong to the same path from $\mathscr{R}$. Otherwise, the derivation step is *path-changing*. By definition, the initial derivation step of any derivation is always path-preserving.

Example 3.6: Let $G = (\{S, A, B\}, \{a, b, c, d, e, f\}, P, S)$ be a CFG with

$$P = \{1: S \to aSb,\ 2: S \to AB,\\ 3: A \to cAd,\ 4: A \to \varepsilon,\\ 5: B \to eBf,\ 6: B \to \varepsilon\}.$$

Obviously, $L(G) = \{a^{n_1}c^{n_2}d^{n_2}e^{n_3}f^{n_3}b^{n_1} \mid n_1, n_2, n_3 \geq 0\}$ which is nonlinear context-free language of index 2. Consider the following derivation:

$$S \Rightarrow aSb \Rightarrow aABb \Rightarrow acAdBb \Rightarrow acAdeBfb \Rightarrow acdeBfb \Rightarrow acdefb\ [123546]$$

A graph representing ${}_G\triangle(S \Rightarrow^* acdefb[123546])$ is illustrated in Figure 1. Solid lines denote the edges of the tree, while the dashed lines denote the path of the derivation. If they overlap, the derivation step is path-preserving; otherwise, it is path-changing. By the definition, the initial derivation step is always path-preserving. Then, the graph demonstrates that there are precisely three path-changing derivation steps in $S \Rightarrow^* acdefb[123546]$.

Definition 3.7: A *matrix grammar* (*MG* for short) is a pair $H = (G, M)$, where

- $G = (N, T, P, S)$ is a context-free grammar;
- M is a finite language over the alphabet of rules ($M \subseteq P^*$).

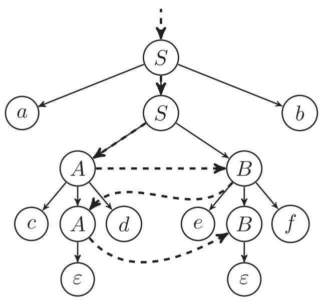

Figure 1. ${}_G\triangle(S \Rightarrow^* acdefb\,[123546])$.

For $x, y \in (N \cup T)^*$, $m \in M$, H performs a *direct derivation step* from x to y according to the *matrix* m denoted by $x \Rightarrow y[m]$, if and only if there are $x_0, x_1, \dots, x_n$ such that $x_0 = x$, $x_n = y$, and

- $x_0 \Rightarrow x_1[p_1] \Rightarrow x_2[p_2] \Rightarrow \cdots \Rightarrow x_n[p_n]$ in G, and
- $m = p_1 p_2 \dots p_n$, where $p_i \in P$, $1 \leq i \leq n$, for some $n \geq 1$.

The transitive and reflexive closure is defined and denoted as usual. Then,

$$\mathscr{L}(H) = \{x \in T^* \mid S \Rightarrow^* x\}$$

is the *language generated by H*. The *family of languages generated by matrix grammars* is called the *family of matrix languages* and introduced in [1].

Definition 3.8: A *regular-controlled grammar* (an RCG for short) H is a pair $H = (G, C)$, where *core grammar* $G = (N, T, P, S)$ is a context-free grammar and *control language* $C \subseteq P^*$ is a regular language. If $S \Rightarrow^* w[\alpha]$ in G and $\alpha\beta \in C$, for some $\alpha, \beta \in P^*$, we put $S \Rightarrow^* w\,[\alpha]$ in H or $S \Rightarrow^* w$ in H for short. The *language generated by H*, denoted by $L(H)$, is defined as

$$L(H) = \{w \in T^* \mid S \Rightarrow^* w[\alpha] \text{ in } G, \alpha \in C\}.$$

We define the *family of all regular-controlled languages*, denoted by **RC**, as

$$\mathbf{RC} = \{L \mid L = L(H) \text{ and } H \text{ is RCG}\}.$$

It was previously proved that **CF** ⊂ **RC** ⊂ **CS**—indeed, **RC** coincides with the family of matrix languages [10].

Let us demonstrate the notion of regular-controlled grammars.

Example 3.9: Let $H = (G, C)$ be an RCG with $G = (\{S, A, B\}, \{a, b\}, P, S)$, where

$$\begin{aligned} P = \{&1 : S \rightarrow AB, \\ &2 : A \rightarrow aA,\ 3 : B \rightarrow aB, \\ &4 : A \rightarrow bA,\ 5 : B \rightarrow bB, \\ &6 : A \rightarrow a,\ 7 : B \rightarrow a, \\ &8 : A \rightarrow b,\ 9 : B \rightarrow b\} \end{aligned}$$

and

$$C = 1\{23, 45\}^*\{67, 89\}.$$

Observe the control language. After applying the initial rule, there are always consecutive pairs of rules 23 and 45 applied. Eventually, the derivation finishes with application of rules 67 or 89. Then,

$$L(H) = \{ww \mid w \in \{a, b\}^+\},$$

which is the well-known non-context-free context-sensitive language.

Example 3.10: Let $H = (G, C)$ be an RCG with $G = (\{S, A, B\}, \{a, b, c, d\}, P, S)$, where

$$\begin{aligned} P = \{1 : S &\rightarrow aAcB, \\ &2 : A \rightarrow aA,\ 3 : B \rightarrow cB, \\ &4 : A \rightarrow bA,\ 5 : B \rightarrow dB, \\ &6 : A \rightarrow b,\ 7 : B \rightarrow d\} \end{aligned}$$

and

$$C = 1\{23\}^*\{45\}^*67.$$

Observe the construction of H. By the control language, after the application of the initial rule, first, zero or more consecutive applications of rules 23 are performed, second, zero or more consecutive applications of rules 45 are performed. Finally, the derivation finishes by the rules 67. Then,

$$L(H) = \{a^m b^n c^m d^n \mid m, n \geq 1\}$$

which is the well-known non-context-free context-sensitive language.

All the previous notation concerning CFGs and their derivation trees still applies for RCGs, however, not every derivation of a core grammar is legal according to the control language.

Theorem 3.11: *Let $H = (G, C)$ be an arbitrary RCG. Then, there exists RCG $H' = (G', C')$ with $L(H) = L(H')$, where G' is in the binary form.*

Proof: We introduce Algorithm 1 for conversion of RCG into a corresponding RCG in the binary form and prove its correctness.

Claim 1: Algorithm 1 is correct.

Proof: *Basic idea*. The initial steps 1 through 8 construct a template for the resulting grammar.

During 9 through 24 every rule which does not satisfy binary form is decomposed and replaced by three new rules. The first and the second rule is in the binary form. If the third rule does not satisfy the binary form, it enters the following iterative process, however, with shorter right-hand side than the replaced one; this ensures the finiteness of the procedure. Additionally, a new homomorphism is introduced to substitute the replaced rule in the control language with the sequence of newly introduced rules. This iterative process finally defines a finite hierarchy of homomorphisms.

Since **REG** is closed under homomorphism (see [14]), in the last 25th step of the algorithm a new control language is established by application of the defined homomorphisms. The complete rigorous proof is left to the reader. ■

Since for any RCG H we can construct RCG H' in the binary form, where $L(H) = L(H')$, the Theorem 3.11 holds. ■

For any RCG $H = (G, C)$, where G is in the binary form, we say that H is in the binary form. In what follows, unless explicitly stated otherwise, we automatically assume that every RCG is in the binary form.

Algorithm 1 Conversion of RCG into the binary form

```
Input: An arbitrary RCG $H = (G, C)$, $G = (N, T, P, S)$.
Output: RCG $H'$ in the binary form with $L(H') = L(H)$.
 1: Construct $H' = (G', C')$, $G' = (N', T, P', S)$; $C' = P' = \emptyset$, $N' = N$.
 2: for all $r \in P$ do
 3:     if $r$ satisfies the binary form then
 4:         $P' \leftarrow P' \cup \{r\}$
 5:         $P \leftarrow P - \{r\}$
 6:     end if
 7: end for
 8: $i \leftarrow 0$
 9: while there exists $r : A \to w \in P$ do
10:     for $uXv = w$ and $\mathrm{alph}(u) \cap N = \emptyset$ and $X \in N$ do
11:         $N' \leftarrow \langle uX\rangle, \langle v\rangle$
12:         $P' \leftarrow P' \cup \{r_1 : A \to \langle uX\rangle\langle v\rangle, r_2 : \langle uX\rangle \to uX\}$
13:         if $\langle v\rangle \to v$ satisfies the binary form then
14:             $P' \leftarrow P' \cup \{r_3 : \langle v\rangle \to v\}$
15:         else
16:             $P \leftarrow P \cup \{r_3 : \langle v\rangle \to v\}$
17:         end if
18:     end for
19:     Define a new homomorphism $h_i$ over $P \cup P'$:
20:         $h_i(x) = r_1r_2r_3$, for $x = r$;
21:         $h_i(x) = x$, otherwise.
22:     $P \leftarrow P - \{r\}$
23:     $i \leftarrow i + 1$
24: end while // $P = \emptyset$
25: $C' \leftarrow \{w \mid w = h_i(h_{i-1}(\cdots h_1(h_0(x))\cdots)), x \in C\}$
```

Example 3.12: Let $H = (G, C)$ be an RCG with G from Example 3.6 and

$$C = \{1\}^*\{2\}\{3\}^*\{4\}\{5\}^*\{6\}.$$

Obviously, G is in the binary form. Even though $L(H) = L(G)$, the derivation $S \Rightarrow^* acdefb\,[123546]$ from Example 3.6 is not legal in H, since $123546 \notin C$. However, $123456 \in C$ and, thus,

$$S \Rightarrow aSb \Rightarrow aABb \Rightarrow acAdBb \Rightarrow acdBb \Rightarrow acdeBfb \Rightarrow acdefb[123456]$$

in H. A graph representing ${}_G\triangle(S \Rightarrow^* acdefb[123456])$ is illustrated in Figure 2.

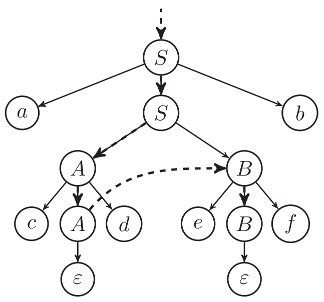

Figure 2. ${}_G\triangle(S \Rightarrow^* acdefb\,[123456])$.

Notice that there is only one path-changing derivation step despite the grammar is clearly of index 2. Indeed, a grammar with only linear rules is obviously of index 1 and no path-changing derivation steps occur in its derivations. For a grammar in the binary form the index as well as the minimal number of path-changing derivation steps increase by one for every branching derivation step, because every branch must be ones terminated in a successful derivation. Therefore, if there is a constant k limiting the number of path-changes, the index of a grammar is at most $k+1$.

4. Results

In the present section, we establish the main result of the study.

Theorem 4.1: *If there is a constant $k \geq 0$ and a regular-controlled grammar H in binary form such that, for every $w \in L(H)$, there exist a derivation of w in H with at most k path-changing derivation steps, then $L(H)$ is a context-free language, and moreover, it is of index $k+1$.*

An RCG satisfying restriction from Theorem 4.1 is said to be *k-restricted.*

Proof: Let $\overline{H} = (\overline{G}, \overline{C})$, $\overline{G} = (\overline{N}, T, \overline{P}, S)$, be an RCG in the binary form such that $L(\overline{H}) = L$ and let $k \geq 0$ be a constant such that for every $x \in L(\overline{H})$, there exists a derivation $S \Rightarrow^* x$ in $\overline{H}$ with k or fewer path-changing derivation steps.

Preliminary transformation. Construct $H = (G, C)$, $G = (N, T, P, S)$, as follows. Initially, set $C = \emptyset$, $N = \overline{N}$, and $P = \{r \mid r : A \to w \in \overline{P}, \#_{\overline{N}}(w) = 1\}$. Define the new homomorphism h over P as $h(x) = x$, for all $x \in P$. For every rule $r : A \to BC \in \overline{P}$, where $A, B, C \in \overline{N}$, add new nonterminal $\langle r \rangle$ to N and two new rules

$$r_1 : A \to \langle r \rangle C,\ r_2 : \langle r \rangle \to B$$

to P and redefine h so that $h(r) = r_1 r_2$. For every rule $r : A \to w \in \overline{P}$, where $A \in \overline{N}$ and $w \in T^*$, add new nonterminal $\langle r \rangle$ to N and two new rules

$$r_1 : A \to \langle r \rangle,\ r_2 : \langle r \rangle \to w$$

to P and redefine h so that $h(r) = r_1 r_2$. Finally set $C = h(\overline{C})$.

Claim 2: $L(\overline{H}) = L(H)$.

Proof: Since H is constructed so that every rule of the form $r : A \to BC$ or $r : A \to w$ is substituted by two always consecutively applied rules $r_1 : A \to \langle r \rangle C$ and $r_2 : \langle r \rangle \to B$ or $r_1 : A \to \langle r \rangle$ and $r_2 : \langle r \rangle \to w$, respectively, working equally, it is obvious that the claim holds. ■

Moreover, the preliminary transformation does not add any new branching and, thus, preserves k as a valid limit of path-changes.

The previous transformation aims to simplify the next construction proof. We avoid path-changes during branching and directly before leafs. Additionally, after every branching the derivation always continues with the left child node.

Construction. Let $M = (Q, P, R, s, F)$ be a finite automaton such that $L(M) = C$. Set

$$Q^{\leq k} = \bigcup_{i=0}^{k} Q^i, \overline{N} = \{\langle \overline{A}|q|r|s|t|f \rangle \mid A \in N; q \in Q; r, s, t \in Q^{\leq k}; f \in F \cup \{\varepsilon\}\},$$

$$N' = \{\langle A|q|r|s|t|f \rangle \mid A \in N; q \in Q; r, s, t \in Q^{\leq k},$$
$$f \in F \cup \{\varepsilon\}\} \cup \{\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon \rangle\} \cup \overline{N},$$

where $S' \notin N$. Construct a context-free grammar $G' = (N', T, P', \langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon \rangle)$. Set $P' = \emptyset$. Construct P' by performing (I) through (VI) given next.

(I) For all $x \in Q^{\leq k}$ and $f \in F$, add $\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \to \langle S|s|\varepsilon|\varepsilon|x|f\rangle$ to P';

(II) for all $r: A \to uBv \in P, qr \vdash p \in R, x, y, z \in Q^{\leq k}$, and $f \in F \cup \{\varepsilon\}$, where $B \in N, uv \in T^*$, add
  (i) $\langle A|q|x|y|z|f\rangle \to u\langle B|p|x|y|z|f\rangle v$,
  (ii) $\langle \overline{A}|q|x|y|z|f\rangle \to u\langle B|p|x|y|z|f\rangle v$ to P';

(III) for all $r: A \to uBv \in P$, $qr \vdash p \in R$, $g \in Q$, $x, y, z \in Q^{\leq k}$, and $f \in F \cup \{\varepsilon\}$, where $B \in N$, $uv \in T^*$, add $\langle A|g|gx|qy|z|f\rangle \to u\langle B|p|x|y|z|f\rangle v$ to P';

1. for all $r: A \to w \in P$ and $qr \vdash p \in Q$, where $w \in T^*$, add
  (i) $\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle \to w$,
  (ii) $\langle \overline{A}|q|p|\varepsilon|\varepsilon|\varepsilon\rangle \to w$ to P';

(V) for all $r: A \to w \in P$ and $qr \vdash f \in Q$, where $w \in T^*, f \in F$, add
  (i) $\langle A|q|\varepsilon|\varepsilon|\varepsilon|f\rangle \to w$,
  (ii) $\langle \overline{A}|q|\varepsilon|\varepsilon|\varepsilon|f\rangle \to w$ to P';

(VI) for all $r: A \to BC \in P$, $qr \vdash p \in R$, $B, C \in N$, $g \in Q$, $x_1x_2, y_1y_2, z_1z_2z_3z_4 \in Q^{\leq k}$, $f \in F \cup \{\varepsilon\}$, $f_1f_2 = f$, add
  (i) $\langle A|q| \bullet (x_1, x_2)| \bullet (y_1, y_2)|gz_1z_2z_3z_4|f\rangle \to \langle B|p| \bullet (x_1, gz_1)| \bullet (y_1, z_2)|z_3|f_1\rangle\langle \overline{C}|g| \bullet (x_2, z_2)| \bullet (y_2, z_1)|z_4|f_2\rangle$,
  (ii) $\langle \overline{A}|q| \bullet (x_1, x_2)| \bullet (y_1, y_2)|gz_1z_2z_3z_4|f\rangle \to \langle B|p| \bullet (x_1, gz_1)| \bullet (y_1, z_2)|z_3|f_1\rangle\langle \overline{C}|g| \bullet (x_2, z_2)| \bullet (y_2, z_1)|z_4|f_2\rangle$,
  (iii) $\langle A|q| \bullet (x_1, x_2)|g \bullet (y_1, y_2)|z_1z_2z_3z_4|f\rangle \to \langle B|p| \bullet (x_1, z_1)| \bullet (y_1, z_2)|z_3|f_1\rangle\langle \overline{C}|g| \bullet (x_2, z_2)| \bullet (y_2, z_1)|z_4|f_2\rangle$,
  (iv) $\langle \overline{A}|q| \bullet (x_1, x_2)|g \bullet (y_1, y_2)|z_1z_2z_3z_4|f\rangle \to \langle B|p| \bullet (x_1, z_1)| \bullet (y_1, z_2)|z_3|f_1\rangle\langle \overline{C}|g| \bullet (x_2, z_2)| \bullet (y_2, z_1)|z_4|f_2\rangle$ to P'.

Define the new morphism $\gamma : (N' \cup T)^* \to (N \cup T)^*$ such that for $\langle A|q|x|y|z|f\rangle \in N'$, $\gamma(\langle A|q|x|y|z|f\rangle) = A$, $\gamma(x) = x$ otherwise.

Basic idea. The context-free grammar G' is designed to simulate the derivations of H. Since in any derivation of H there are k or fewer path-changes, G' nondeterministically decides about all the path-changes during the initial derivation step. To satisfy the restrictions given by control language C, the automaton M, $L(M) = C$, is encoded in the rules of G'. While performing linear derivations, the consecutivity of states is ensured. When a new branching node is introduced, it is nondeterministically decided about path-changes between both subtrees of the derivation tree which are encoded in nonterminals and simulated by context-free rules.

Let us describe the composite nonterminal symbols in greater detail. For a symbol

$$\langle A|q|x|y|z|f\rangle$$

composed of symbol A, states q and f, and the stings of zero up to k states x, y, and z, we refer to A, q, x, y, z, and f as a first, second, third, fourth, fifth, and sixth component, respectively. The first component encodes nonterminal symbol itself, while the others encode states of the finite automaton M with $L(M) = C$. The second component encodes the current state of M. The third component holds the string of states from which there is a path-change underneath the current branch of the derivation tree, while the fourth component holds the string of states into which there is a path-change underneath the current branch of the derivation tree. The fifth component represents a string of branching states, which are to be set as the branching ones during the rest of the derivation. Finally, the sixth component encodes the final state of M to be reached.

Let us informally describe six classes of the rules of G':

(I) An initial rule of the form $\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \to \langle S|s|\varepsilon|\varepsilon|x|f\rangle$ rewriting the start symbol is applied only once at the beginning of any derivation. It nondeterministically generates x – a string of all states in which there is a path-change – and f – a final state of M to be reached – which are then saved in the fifth and sixth component of a nonterminal, respectively.

(II) The rules of the form $\langle A|q|x|y|z|f\rangle \rightarrow u\langle B|p|x|y|z|f\rangle v$ simulate consecutive path-preserving linear derivations which are designed to follow transitions in M or path-changes into the right child of a new branching node. The first component of a nonterminal represents nonterminal in G, while the second represents a state of M.

(III) The rules of the form $\langle A|g|gx|qy|z|f\rangle \rightarrow u\langle B|p|x|y|z|f\rangle v$ represent path-changes. Since the third component of a nonterminal represents a string of states in which the path-changes out of the subtree of the current node occur, a path-change may be performed only when the first symbol corresponds to the current state of M. The fourth component represent a string of states in which the path-changes into the subtree of the current node occur. Since there is a path-change out and the node is not terminal, if the derivation is successful, there once follows a path-change back simulated by the rule.
Notice that the rules (III) cannot rewrite noterminals from $\overline{N}$ generated by the rules (VI). They simulate path-change out and the following path-change back at once which, however, does not correspond to a path-change into the right subbranch of a new branching node.

(IV) The rules of the form $\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle \rightarrow w$ act slightly similarly to (III), however, a new node is terminal and M does not terminate yet, thus, a path-change out of the current branch must be performed, but there is no path-change back. Additionally, all the previously nondeterministically planned path-changes must be already done – the fourth and sixth component of a nonterminal is empty and the third contains precisely one state.

(V) A rule of the form $\langle A|q|\varepsilon|\varepsilon|\varepsilon|f\rangle \rightarrow w$ terminates the current derivation with respect to M, therefore, there follows no path-change. In every successful derivation there is always exactly one such rule applied.

(VI) The last class of the rules represents branching. Since G is in the binary form, every node has at most two children. Moreover, by the preliminary transformation of H it is ensured that the derivation follows by rewriting the left newly introduced nonterminal, thus, we do not consider other cases (e.g. path-changing while branching). To terminate the right branch, there must once occur a path-change into it which is planned while branching. A path-change may lead from the subtree of the left branch – (i)– (ii) – or is already planned – (iii)–(iv).
The third and fourth components of $\langle A|q|\bullet(x_1,x_2)|\bullet(y_1,y_2)|z_1z_2z_3z_4|f\rangle$ are nondeterministically divided into newly introduced branches, but the mutual order of the states is preserved, and some path-changes from the fifth component may be nondeterministically planned between both new branches. Finally, if $f \in F$, it is decided into which branch it is put.

We note that there is a lot of rules or nonterminals which possibly do not occur in any successful derivation. Moreover, a nondeterministic generation and distribution of path-changing states may result into blocking of a derivation. As we prove next, this, however, does not change the language of the grammar.

To clarify the construction part of the proof of Theorem 4.1, we provide Example A.8 in Appendix 1.

Claim 3: If $S \Rightarrow^m w$ in H, where $m \geq 0$ and $w \in (N \cup T)^*$, then $\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow^* w'$ in G', where $w' \in (N' \cup T)^*$ and $\gamma(w') = w$.

Proof: We prove the statement by induction on $m \geq 0$.

Basis Let $m=0$. That is, $S \Rightarrow^0 S$ in H. Clearly, $\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow \langle S|s|\varepsilon|\varepsilon|x|f\rangle$ in G', where $\gamma(\langle S|s|\varepsilon|\varepsilon|x|f\rangle) = S$, for some $x \in Q^{\leq k}$ and $f \in F$, so the basis holds.

Induction Hypothesis Suppose that there exists $n \geq 0$ such that Claim 3 holds for all m with $0 \leq m \leq n$.

Induction Step Let $S \Rightarrow^{n+1} w$ in H. Then, $S \Rightarrow^n v \Rightarrow w$, where $v \in (N \cup T)^*$, and there exists $r \in P$ such that $v \Rightarrow w\,[r]$. By the induction hypothesis, $\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow^* v'$, where $\gamma(v') = v$, in G'. Next, we consider the following five forms of r according to the construction of G'.

(1) Let $r : A \to u_1Bu_2 \in P$, for some $A, B \in N$, $u_1, u_2 \in T^*$, and $v \Rightarrow w\,[r]$ is a path-preserving derivation step or a path-changing derivation step into a node with some sibling. By the construction of G', there exists a rule $\langle A|q|x|y|z|f\rangle \to u_1\langle B|p|x|y|z|f\rangle u_2$ in P', where $qr \vdash p \in R$, $x, y, z \in Q^{\leq k}$, and $f \in F \cup \{\varepsilon\}$. Without any loss of generality, suppose q,x,y,z,f are correct. Then, there exists a derivation

$$v' \Rightarrow w'\ [\langle A|q|x|y|z|f\rangle \to u_1\langle B|p|x|y|z|f\rangle u_2]$$

in G', where $\gamma(w') = w$.

(2) Let $r : A \to u_1Bu_2 \in P$, for some $A, B \in N$, $u_1, u_2 \in T^*$, and $v \Rightarrow w[r]$ is a path-changing derivation step into a node without siblings. By the construction of G', there exists a rule $\langle A|g|gx|qy|z|f\rangle \to u_1\langle B|p|x|y|z|f\rangle u_2$ in P', where $g \in Q$, $qr \vdash p \in R$, $x, y, z \in Q^{\leq k}$, and $f \in F \cup \{\varepsilon\}$. Without any loss of generality, suppose g,q,x,y,z,f are correct. Then, there exists a derivation

$$v' \Rightarrow w'\ [\langle A|g|gx|qy|z|f\rangle \to u_1\langle B|p|x|y|z|f\rangle u_2]$$

in G', where $\gamma(w') = w$.

(3) Let $r : A \to x \in P$, for some $A \in N$, $x \in T^*$, and $\mathrm{alph}(w) \cap N \neq \emptyset$. By the construction of G', there exists a rule $\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle \to x$ in P', where $qr \vdash p \in R$. Without any loss of generality, suppose q is correct. Then, there exists a derivation

$$v' \Rightarrow w'\ [\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle \to x]$$

in G', where $\gamma(w') = w$.

(4) Let $r : A \to x \in P$, for some $A \in N$, $x \in T^*$, and $w \in T^*$. By the construction of G', there exists a rule $\langle A|q|\varepsilon|\varepsilon|\varepsilon|f\rangle \to x$ in P', where $qr \vdash f \in R, f \in F$. Without any loss of generality, suppose q is correct. Then, there exists a derivation

$$v' \Rightarrow w'\ [\langle A|q|\varepsilon|\varepsilon|\varepsilon|f\rangle \to x]$$

in G', where $\gamma(w') = w$.

(5) Let $r : A \to BC \in P$, for some $A, B, C \in N$. By the construction of G', there exists a rule $\langle A|q|x_1|y_1|z_1|f_1\rangle \to \langle B|p|x_2|y_2|z_2|f_2\rangle\langle C|g|x_3|y_3|z_3|f_3\rangle$ in P', where $qr \vdash p \in R$, $x_i, y_i, z_i \in Q^{\leq k}$, $f_i \in F \cup \{\varepsilon\}$, and without any loss of generality, suppose q,g,x_i,y_i,z_i,f_i are correct, for $1 \leq i \leq 3$. Then, there exists a derivation

$$v' \Rightarrow w'\ [\langle A|q|x_1|y_1|z_1|f_1\rangle \to \langle B|p|x_2|y_2|z_2|f_2\rangle\langle C|g|x_3|y_3|z_3|f_3\rangle]$$

in G', where $\gamma(w') = w$.

We covered all possible forms of p, so the claim holds. ■

Let us remark that assumption of correctness of nonterminals of G' results from the fact that the rules cover all possibilities—that is there is always a proper rule to be used.

Claim 4: Consider any $w \in T^*$, where $w \notin L(H)$. Then, $w \notin L(G')$.

Proof: We prove this by contradiction.

Assumption. Suppose there exists $w \in T^*$, where $w \notin L(H)$ and $w \in L(G')$.

(1) First, suppose $w \notin L(G)$. That is, there exists a derivation

$$\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow \langle S|s|\varepsilon|\varepsilon|x|f\rangle \Rightarrow^* u \Rightarrow v\,[A \to X] \Rightarrow^* w,$$

in G', where $\gamma(u) \not\Rightarrow \gamma(v)$ in G, for some $u, v \in (N' \cup T)^*$, $\langle S|s|\varepsilon|\varepsilon|x|f\rangle \in N'$, and $A \to X \in P'$. Then, $\gamma(A) \to \gamma(X) \notin P$. However, since by the construction of G' every non-initial rule $A \to X \in P'$ is introduced according to some $\gamma(A) \to \gamma(X) \in P$, this is a contradiction.

(2) Second, suppose $w \in L(G)$, however, for every derivation $S \Rightarrow^* w[d]$ in G, $d \notin C$. In the terms of M, there is no derivation $sd \vdash^* \overline{q}$, for any $\overline{q} \in Q$, or $sd \vdash \overline{q}$ and $\overline{q} \notin F$. Consider a derivation

$$\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow \langle S|s|\varepsilon|\varepsilon|\overline{x}|\overline{f}\rangle \Rightarrow^* w,$$

in G', for some $\langle S|s|\varepsilon|\varepsilon|\overline{x}|\overline{f}\rangle \in N'$, and a corresponding derivation $S \Rightarrow^* w\,[d]$ in G, for some $d \in R^*$.

(a) Suppose there is no derivation $sd \vdash^* \overline{q}$, for any $\overline{q} \in Q$. Then, there exist $u, v \in (N' \cup T)^*$ and $\langle A|q|x|y|z|f\rangle \rightarrow X \in P'$, where

$$\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow \langle S|s|\varepsilon|\varepsilon|\overline{x}|\overline{f}\rangle \Rightarrow^* u \Rightarrow v[\langle A|q|x|y|z|f\rangle \rightarrow X] \Rightarrow^* w,$$

in G', and a corresponding derivation

$$S \Rightarrow^* \gamma(u)\,[d_1] \Rightarrow \gamma(v)\,[r] \Rightarrow^* w[d_2]$$

in G, where $d_1rd_2 = d$, $r : A \rightarrow \gamma(X) \in P$, $sd_1 \vdash^* q$, and there is no derivation $sd_1r \vdash^* q'$, for any $q, q' \in Q$.

(i) Suppose $\gamma(u) \Rightarrow \gamma(v)[r]$ is a path-preserving derivation step. Then, $\langle A|q|x|y|z|f\rangle \rightarrow X$ is from (II) or (sec(1)) through (VI) depending on X. By the construction of G', the rule $\langle A|q|x|y|z|f\rangle \rightarrow X$ is introduced according to a transition $qr \vdash p \in R$, for some $p \in Q$. Therefore, however, $sd_1r \vdash^* p$ in M, which is a contradiction.

(ii) Suppose $\gamma(u) \Rightarrow \gamma(v)[r]$ is a path-changing derivation step. The states in which the path-changing derivation steps are always to be performed – represented by the string $\overline{x}$ – are nondeterministically generated by the initial derivation step

$$\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow \langle S|s|\varepsilon|\varepsilon|\overline{x}|\overline{f}\rangle$$

in the fifth component of $\langle S|s|\varepsilon|\varepsilon|\overline{x}|\overline{f}\rangle$. Therefore, $q \in \text{alph}(\overline{x})$. However, before the path-change can be simulated by G', the state must get to the third and fourth component of some nonterminals which can be done by the rule (VI) only. Then, by some rule

$$\langle A_1|q_1|x_1|y_1|z_1|f_1\rangle \rightarrow \langle A_2|q_2|x_2|y_2|z_2|f_2\rangle\langle A_3|q_3|x_3|y_3|z_3|f_3\rangle \in P',$$

where $\#_q(z_1) \geq \#_q(z_2) + \#_q(z_3) + 1$,

$$\begin{aligned}\langle S|s|\varepsilon|\varepsilon|\overline{x}|\overline{f}\rangle &\Rightarrow^* u_1\langle A_1|q_1|x_1|y_1|z_1|f_1\rangle u_2\\ &\Rightarrow u_1\langle A_2|q_2|x_2|y_2|z_2|f_2\rangle\langle A_3|q_3|x_3|y_3|z_3|f_3\rangle u_2,\end{aligned}$$

for some $u_1, u_2 \in (N' \cup T)^*$. Therefore, $q \in \text{alph}(x_2)$ or $q \in \text{alph}(x_3)$ and since these two cases are symmetric, without any loss of generality, let us consider only $q \in \text{alph}(x_2)$. Then, also $q_3 = q$ and

$$\langle A_3|q_3|x_3|y_3|z_3|f_3\rangle = \langle A_3|q|x_3|y_3|z_3|f_3\rangle$$

or $q \in \text{alph}(y_3)$ and to once get rid of it

$$\langle A_3|q_3|x_3|y_3|z_3|f_3\rangle \Rightarrow^* w_1\langle A|g|gx|qy|z|f\rangle w_2$$

for some $w_1, w_2 \in (N' \cup T)^*$. Either $\langle A_3|q|x_3|y_3|z_3|f_3\rangle$ or $\langle A|q|x|y|z|f\rangle$ represents the target of the path-change later denoted by Z. Additionally, G' must once get rid of q in

x_2, otherwise, the derivation is not successful. Hence,

$$\langle A_2|q_2|x_2|y_2|z_2|f_2\rangle \Rightarrow^* v_1 Y v_2,$$

where $Y = \langle A_4|q|qx_4|y_4|z_4|f_4\rangle$ which consequently allows an application of a rule (III) erasing q, for some $v_1, v_2 \in (N' \cup T)^*$ and $\langle A_4|q|qx_4|y_4|z_4|f_4\rangle \in N'$, or $Y \in T^*$ and

$$\begin{aligned}\langle A_2|q_2|x_2|y_2|z_2|f_2\rangle &\Rightarrow^* v_1\langle A'|q'|q|\varepsilon|\varepsilon|\varepsilon\rangle v_2[\langle A'|q'|q|\varepsilon|\varepsilon|\varepsilon\rangle \to Y]\\ &\Rightarrow v_1 Y v_2,\end{aligned}$$

for some $\langle A'|q'|q|\varepsilon|\varepsilon|\varepsilon\rangle \to Y \in P'$. Combining the previous observations and statements, we get

$$\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow^* u_1 v_1 Y v_2 w_1 Z w_2 u_2.$$

Since G' is a context-free grammar, without any loss of generality, we can suppose that the derivation follows M—that is, $sd_1 \vdash q \in M$. Then, the derivation

$$u_1 v_1 Y v_2 w_1 Z w_2 u_2 \Rightarrow u_1 v_1 Y v_2 w_1 X w_2 u_2$$

represents a path-changing derivation step changing the path of the derivation from Y to X. By the construction of G', the rule $\langle A|q|x|y|z|f\rangle \to X$ or $\langle A|g|gx|qy|z|f\rangle \to X$ by which the last derivation step is performed is introduced according to a transition $qr \vdash p \in R$, for some $p \in Q$. Therefore, however, $sd_1 r \vdash^* p$ in M, which is a contradiction.

(d) Suppose $sd \vdash^* \overline{q}$, where $\overline{q} \in Q - F$. In every successful derivation of G', there is precisely one rule (V) applied – G' must once get rid of $\overline{f}$ generated by the initial derivation step – which represents final accepting transition of M. Since G' is a context-free grammar, without any loss of generality, we can consider an application of such rule is always performed at the end of any successful derivation; this is also consistent with M and, thus, C. Then, however, $\overline{q} \in F$, which is a contradiction.

Since the assumption always results in contradiction, it is incorrect. ■

By Claim 3, if $S \Rightarrow^* w$ in H, then $\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow^* w'$ in G', where $\gamma(w') = w$. If $S \Rightarrow^* w$ in H and $w \in T^*$, then $w \in L(H)$. Since $\gamma(w') = w' = w$, for $w \in T^*$, $w' \in L(G')$. By Claim 4, $L(G') - L(H) = \emptyset$. Therefore, $L(H) = L(G')$. By Claim 2 $L(\overline{H}) = L(H)$ and Theorem 4.1 holds. ■

Corollary 4.2: *Let L be a context-free language of an infinite index. Then, there exists no k-restricted regular-controlled grammar H such that L(H) = L, for any* $1 \le k < \infty$.

Corollary 4.3: *A language L is context-free of index k if there is a constant* $k \ge 1$ *and a propagating regular-controlled grammar H in the binary form such that L = L(H) and for every* $x \in L(H)$, *there exists a derivation* $S \Rightarrow^* x$ *in H with k or fewer path-changing derivation steps.*

We introduced the binary form of regular-controlled grammars to simplify the proof of Theorem 4.1. Notice, however, it can be generalized for all k-restricted RCGs. A proof of the following theorem is, thus, left to the reader.

Theorem 4.4: *A language L is context-free of index k+1 if there is a constant* $k \ge 0$ *and a regular-controlled grammar H such that L = L(H) and for every* $x \in L(H)$, *there exists a derivation* $S \Rightarrow^* x$ *in H with k or fewer path-changing derivation steps.*

The control mechanism of regular-controlled grammars influences the order in which the core grammars apply their rules. However, the notion of path-change as well as the given restrictions are independent of this control mechanism and are related only to the core grammars and their derivation trees. Therefore, we can state the achieved result in a more general context.

Corollary 4.5: *A language L is context-free of index k if there is a constant $k \geq 1$ and a (propagating) matrix grammar H such that $L = L(H)$ and for every $x \in L(H)$, there exists a derivation $S \Rightarrow^* x$ in H with k or fewer path-changing derivation steps.*

Notice that phrase structure grammars, parallel grammars, random context grammars, and other regulated grammars generate their languages in various ways. This variety gives rise to the following open problem.

Open Problem 4.6: How to naturally restrict derivation trees of other types of grammars? How they influence their generative power?

Throughout the paper, we restrict the number of path-changes in a constant way. It is quite natural to think about more general restriction.

Open Problem 4.7: Consider non-constant restriction of path-changes, such as functions over the sentential form lengths. What is the generative power of regular-controlled grammars restricted in this way?

Acknowledgments

The authors deeply thank both anonymous referees for useful comments and suggestions regarding the first version of this paper.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by The Ministry of Education , Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science" FIT-S-14-2299 and LQ1602, the TAČR grant TE01020415, and the BUT grant FIT-S-17-3964.

References

[1] S. Abraham, *Some Questions of Language Theory*, in *Proceedings of the 1965 conference on Computational linguistics*. Association for Computational Linguistics, 1965, pp. 1–11.
[2] A. Aho and J. Ullman, *The Theory of Parsing, Translation, and Compiling*, Series in Automatic Computation, Prentice-Hall, 1972.
[3] N. Chomsky, *On certain formal properties of grammars*, Inf. Control 2 (1959), pp. 137–167.
[4] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms (Appendix B.5)*, McGraw-Hill, 2002.
[5] M.A. Harrison, *Introduction to Formal Language Theory*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1978.
[6] N.D. Jones, *A note on the index of a context-free language*, Inf. Control 16 (1970), pp. 201–202.
[7] C. Martín-Vide, V. Mitrana, and G. Păun (eds.), *Formal Languages and Applications*, Chapter 13, Springer, Berlin, 2004, pp. 249–274.
[8] A. Meduna, *Automata and Languages: Theory and Applications*, Springer, London, 2000.
[9] A. Meduna, *Formal Languages and Computation: Models and Their Applications*, Taylor & Francis, New York, 2014.
[10] A. Meduna and P. Zemek, *Regulated Grammars and Their Transformations*, Brno University of Technology, Brno, 2010.
[11] G. Păun, *On the index of grammars and languages*, Inf. Control 35 (1977), pp. 259–266.
[12] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages, Vol. 1: Word, Language, Grammar*, Springer-Verlag, New York, 1997.

[13] A. Salomaa, *On the index of a context-free grammar and language*, Inf. Control 14 (1969), pp. 474–477.
[14] A. Salomaa, *Formal Languages*, Academic Press, London, 1973.

Appendix 1. Theorem 4.1 : Proof : construction: illustration

Example A.8: Consider RCG $H = (G, C)$ from Example 3.12 and let $k = 1$. Recall $G = (\{S, A, B\}, \{a, b, c, d, e, f\}, P, S)$, $C = \{1\}^*\{2\}\{3\}^*\{4\}\{5\}^*\{6\}$, and

$$\begin{aligned} P = \{&1 : S \to aSb,\ 2 : S \to AB,\\ &3 : A \to cAd,\ 4 : A \to \varepsilon,\\ &5 : B \to eBf,\ 6 : B \to \varepsilon\}. \end{aligned}$$

Construct $H' = (\overline{G}, \overline{C})$ according to the preliminary transformation of the proof of Theorem 4.1 with $\overline{G} = (\{S, A, B, \langle 2\rangle, \langle 4\rangle, \langle 6\rangle\}, \{a, b, c, d, e, f\}, \overline{P}, S)$,

$$\begin{aligned} \overline{P} = \{&1 : S \to aSb,\ 2_1 : S \to \langle 2\rangle B,\ 2_2 : \langle 2\rangle \to A,\\ &3 : A \to cAd,\ 4_1 : A \to \langle 4\rangle,\ 4_2 : \langle 4\rangle \to \varepsilon,\\ &5 : B \to eBf,\ 6_1 : B \to \langle 6\rangle, 6_2 : \langle 6\rangle \to \varepsilon\}, \end{aligned}$$

and $\overline{C} = \{1\}^*\{2_1\}\{2_2\}\{3\}^*\{4_1\}\{4_2\}\{5\}^*\{6_1\}\{6_2\}$. Define an FA $M = \{\{s, s_q, q, q_p, p, p_f, f\},\ \{1, 2_1, 2_2, 3, 4_1, 4_2, 5, 6_1, 6_2\}, R, s, \{f\}\}$ with

$$\begin{aligned} R = \{&s1 \vdash s,\ s2_1 \vdash s_q,\ s_q2_2 \vdash q,\\ &q3 \vdash q,\ q4_1 \vdash q_p,\ q_p4_2 \vdash p,\\ &p5 \vdash p,\ p6_1 \vdash p_f,\ p_f6_2 \vdash f\}. \end{aligned}$$

Next, we define a CFG simulating H', however, to make it as readable as possible, we list only essential nonterminals and rules; despite this example is quite simple, the grammar contains thousands of them, but only very few nonterminals are reachable and terminating (see [9]) and very few rules applicable in any derivation. Define $G' = (N', \{a, b, c, d, e, f\}, P', \langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle)$ with

$$\begin{aligned} N' = \{&\langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle,\ \langle S|s|\varepsilon|\varepsilon|p|f\rangle, \langle\langle 2\rangle|s_q|p|\varepsilon|\varepsilon|\varepsilon\rangle,\ \langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle,\\ &\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle,\ \langle\langle 4\rangle|q_p|p|\varepsilon|\varepsilon|\varepsilon\rangle, \langle\langle 6\rangle|p_f|\varepsilon|\varepsilon|\varepsilon|f\rangle\} \end{aligned}$$

$$\begin{aligned} P' = \{&\dot{0} : \langle S'|s|\varepsilon|\varepsilon|\varepsilon|\varepsilon\rangle \to \langle S|s|\varepsilon|\varepsilon|p|f\rangle,\\ &\dot{1} : \langle S|s|\varepsilon|\varepsilon|p|f\rangle \to a\langle S|s|\varepsilon|\varepsilon|p|f\rangle b,\\ &\dot{2} : \langle S|s|\varepsilon|\varepsilon|p|f\rangle \to \langle\langle 2\rangle|s_q|p|\varepsilon|\varepsilon|\varepsilon\rangle\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle,\\ &\dot{3} : \langle\langle 2\rangle|s_q|p|\varepsilon|\varepsilon|\varepsilon\rangle \to \langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle,\\ &\dot{4} : \langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle \to c\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle d,\\ &\dot{5} : \langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle \to \langle\langle 4\rangle|q_p|p|\varepsilon|\varepsilon|\varepsilon\rangle,\\ &\dot{6} : \langle\langle 4\rangle|q_p|p|\varepsilon|\varepsilon|\varepsilon\rangle \to \varepsilon,\\ &\dot{7} : \langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle \to e\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle f,\\ &\dot{8} : \langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle \to \langle\langle 6\rangle|p_f|\varepsilon|\varepsilon|\varepsilon|f\rangle,\\ &\dot{9} : \langle\langle 6\rangle|p_f|\varepsilon|\varepsilon|\varepsilon|f\rangle \to \varepsilon,\\ &\cdots\}. \end{aligned}$$

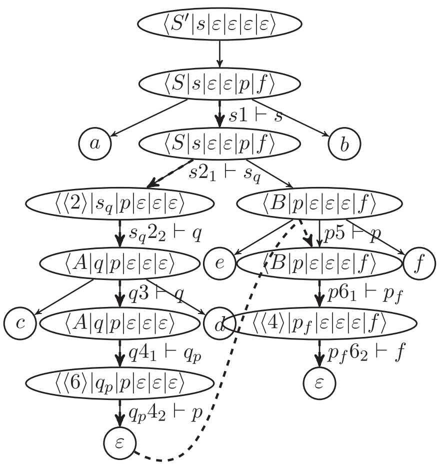
$\langle S'|s|\varepsilon|\varepsilon|\varepsilon\rangle$
$\langle S|s|\varepsilon|\varepsilon|p|f\rangle$
a
$s1 \vdash s$
$\langle S|s|\varepsilon|\varepsilon|p|f\rangle$
b
$s2_1 \vdash s_q$
$\langle\langle 2\rangle|s_q|p|\varepsilon|\varepsilon|\varepsilon\rangle$
$\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle$
$s_q2_2 \vdash q$
$p5 \vdash p$
$\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle$
e
$\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle$
f
$q3 \vdash q$
$p6_1 \vdash p_f$
c
$\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle$
d
$\langle\langle 4\rangle|p_f|\varepsilon|\varepsilon|\varepsilon|f\rangle$
$q4_1 \vdash q_p$
$p_f6_2 \vdash f$
$\langle\langle 6\rangle|q_p|p|\varepsilon|\varepsilon|\varepsilon\rangle$
ε
$q_p4_2 \vdash p$
ε

Figure A1. $_{G'}\triangle(\langle S'|s|\varepsilon|\varepsilon|\varepsilon\rangle \Rightarrow^* acdefb[\dot{0}\dot{1}\dot{2}\dot{3}\dot{4}\dot{5}\dot{6}\dot{7}\dot{8}\dot{9}])$.

For easier referencing, we add a unique label to each rule. Consider the derivation 123456 in G from Example 3.12. The corresponding derivation in G' is as follows.

$$\begin{aligned}
\langle S'|s|\varepsilon|\varepsilon|\varepsilon\rangle &\Rightarrow \langle S|s|\varepsilon|\varepsilon|p|f\rangle \quad [\dot{0}]\\
&\Rightarrow a\langle S|s|\varepsilon|\varepsilon|p|f\rangle b \quad [\dot{1}]\\
&\Rightarrow a\langle\langle 2\rangle|s_q|p|\varepsilon|\varepsilon|\varepsilon\rangle\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle b \quad [\dot{2}]\\
&\Rightarrow a\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle b \quad [\dot{3}]\\
&\Rightarrow ac\langle A|q|p|\varepsilon|\varepsilon|\varepsilon\rangle d\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle b[\dot{4}]\\
&\Rightarrow ac\langle\langle 4\rangle|q_p|p|\varepsilon|\varepsilon|\varepsilon\rangle d\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle b \quad [\dot{5}]\\
&\Rightarrow acd\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle b \quad [\dot{6}]\\
&\Rightarrow acde\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle fb \quad [\dot{7}]\\
&\Rightarrow acde\langle\langle 6\rangle|p_f|\varepsilon|\varepsilon|\varepsilon|f\rangle fb \quad [\dot{8}]\\
&\Rightarrow acdefb \quad [\dot{9}]
\end{aligned}$$

However, it also corresponds to $s12_12_234_14_256_16_2 \vdash^* f$ in M and, thus, to H'. Notice step $\dot{0}$, where $\langle S|s|\varepsilon|\varepsilon|p|f\rangle$ is generated. It encodes that the grammar must once simulate a path-change in state p and apply terminating rule entering final state f – which is step $\dot{9}$ – with respect to M. In branching step $\dot{2}$, state p is put to the third component of $\langle\langle 2\rangle|s_q|p|\varepsilon|\varepsilon|\varepsilon\rangle$ which encodes that it once must be reached in the left branch – this is done in step $\dot{6}$ – and to the second component of $\langle B|p|\varepsilon|\varepsilon|\varepsilon|f\rangle$ simulating that the derivation continues from the same state with respect to M. Figure A1 demonstrates how G' follows M.