How to Demonstrate Metalinearness and Regularity by Tree-Restricted General Grammars

Martin Havel
Brno University of Technology,
Faculty of Information Technology,
Brno, Czech republic
ihavelm@fit.vut.cz

Zbyněk Křivka
Brno University of Technology,
Faculty of Information Technology,
Brno, Czech republic
krivka@fit.vut.cz

Alexander Meduna
Brno University of Technology,
Faculty of Information Technology,
Brno, Czech republic
meduna@fit.vut.cz

This paper introduces derivation trees for general grammars. Within these trees, it defines context-dependent pairs of nodes, corresponding to rewriting two neighboring symbols using a non context-free rule. It proves that the language generated by a general grammar with linear core with a slow-branching derivation tree is k-linear if there is a constant u such that every sentence w in the generated language is the frontier of a derivation tree in which any pair of neighboring paths contains u or fewer context-dependent pairs of nodes. Next, it proves that the language generated by a general grammar with a regular core is regular if there is a constant u such that every sentence w in the generated language is the frontier of a derivation tree in which any pair of neighboring paths contains u or fewer context-dependent pairs of nodes. The paper explains that this result is a powerful tool for showing k-linear or regular languages. It sketches how to apply this tool in practice.

Submitted to:
NCMA 2024

1 Introduction

Formal language theory has always intensively struggled to establish conditions under which general grammars generate a proper subfamily of the family of recursively enumerable languages because results like this often significantly simplify proofs that some languages are members of the subfamily. Continuing with this important investigation trend in formal language theory, the present paper establishes another result of this kind based upon a restriction placed upon a graph-based representation of derivations in general grammars.

Concerning general grammars, which generate a proper subfamily of the family of recursively enumerable languages, some results of this kind have been achieved, too. First of all, [8] states that for a grammar, the set of terminal strings generated by left-to-right derivations is context-free. Second, [9] shows that the set of terminal strings generated by two-way derivations is context-free, which is further studied in [4]. Third, [3] demonstrates that a grammar generates a context-free language if the left-hand side of every rule contains only one nonterminal with terminal strings as the only context. Fourth, also [3] shows that if every rule of a general grammar has as its left context a string of terminal symbols at least as long as the right context, then the generated language is context-free. Fifth, [2] demonstrates that a grammar generates a context-free language if the right-hand side of every rule contains a string of terminals longer than any string of terminals between two nonterminals on the left-hand side. For k-linear grammar, there is no such study. For regularity, there is the publication [6], which shows regularity only in context-free languages.

Finally, Section 2.3.2 in [12] demonstrates context-freeness based on the tree restriction with context-dependency. We expand the importance of introduced context-dependency (see fig 1) to demonstrate metalinearness and regularity.

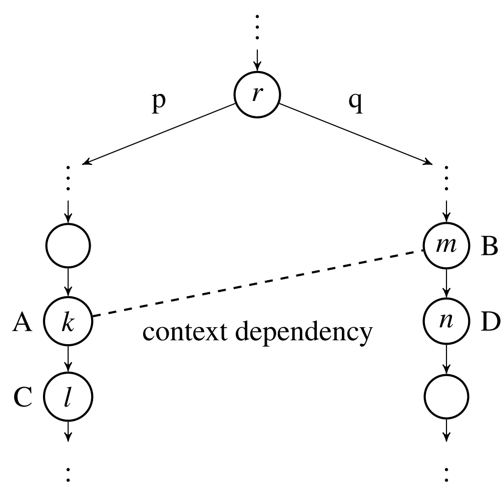

Figure 1: Illustration of context dependency in t

To give an insight into the new result achieved in the present paper, some terminology is first needed to be sketched. We introduce a general grammar G with linear core if any $p \in P$ has one of these forms,

$$AB \rightarrow CD, A \rightarrow BC, A \rightarrow xEy$$

where $A, B, C, D \in V - T$, $E \in (V - T) \cup \{\varepsilon\}$, $x, y \in T^*$ and, respectively, a general grammar G with regular core. We define the notion of a derivation tree t graphically representing a derivation in G by analogy with this notion in terms of a k-linear grammar (see Section 6.2 in [14]). However, in addition, we introduce context-dependent pairs of nodes in t as follows. In t, two paths are neighboring if no other path occurs between them. Let p and q be two neighboring paths in t. Let p contain a node k with a single child l, where k and l are labeled with A and C, respectively, and let q contain a node m with a single child n, where m and n are labeled with B and D, respectively. Let this four-node portion of t; consisting of k, l, m, and n; graphically represents an application of $AB \to CD$. Then k and m are a context-dependent pair of nodes (see Fig. 1).

2 Preliminaries

We assume that the reader is familiar with graph theory, including labeled ordered trees and their terminology (see [1, 5, 7]) as well as formal language theory (see [10, 13, 15]).

A directed graph G is a pair $G = (V,E)$, where V is a finite set of nodes and $E \subseteq V^2$ is a finite set of edges. For a node $v \in V$ a number of edges of the form $(x,v) \in E$ and a number of edges of the form $(v,y) \in E$, for $x,y \in V$, is called an in-degree of v and an out-degree of v, respectively, and denoted by in-d(v), out-d(v). Let $(v_1,v_2,\ldots,v_n)$ be an n-tuple of nodes, for some $n \geq 1$, where $v_i \in V$, for $1 \leq i \leq n$, and there exists an edge $(v_k,v_{k+1}) \in E$, for every pair of nodes v_k,v_{k+1}, where $1 \leq k \leq n-1$, then, we call it a sequence of length n. Let $(v_1,v_2,\ldots,v_n)$ be a sequence of the length n, for some $n \geq 1$, where $v_i \neq v_j$, for $1 \leq i \leq n$, $1 \leq j \leq n$, then, we call the sequence a path. Let $(v_1,v_2,\ldots,v_n)$ be a path in G, for some $n \geq 1$, and $v_1 = v_n$, 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. Let V be an alphabet (finite non-empty set). V^* is the set of all strings over V. Algebraically, V^* represents the free monoid generated by V under the operation of concatenation. The unit 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 $w \in V^*$, $|w|$ denotes the length of w. The alphabet of w, denoted by alph(w), is the set of symbols appearing in w. Let $\mathscr{I}$ denotes the set of all positive integers.

Let $\Rightarrow$ be a relation over V^*. The transitive and transitive and reflexive closure of $\Rightarrow$ are denoted $\Rightarrow^+$ and $\Rightarrow^*$, respectively. Unless explicitly stated otherwise, we write $x \;\Rightarrow\; y$ instead of $(x,y) \in \Rightarrow$.

The families of context-free, context-sensitive and recursively enumerable languages are denoted by **CF**, **CS** and **RE**, respectively.

3 Definitions and Examples

Definition 1. *An* (oriented) tree *is a directed acyclic graph $G = (V,E)$, with a specified node $r \in V$ called the* root *such that* in-d$(r) = 0$, in-d$(x) = 1$, *and there exists a path $(v_1,v_2,\ldots,v_n)$, where $v_1 = r$, $v_n = x$, for some $n \geq 1$, for all $x \in V - \{r\}$. 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 and vice versa. A tree is called* labeled, *if there exist a set of labels $\mathscr{L}$ and a total mapping $l : V \to L$.*

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,…,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 $j < k$, n_j is* left sibling *of n_k and n_k is* right sibling *of n_j, $1 \leq j \leq r$, $1 \leq k \leq r$.*

Let t be a labeled ordered tree, and let t contains node o. Let $\alpha = (o,m_1,m_2,\ldots,m_r)$, and $\beta = (o,n_1,n_2,\ldots,n_s)$ be two paths in t, for some $r,s \geq 1$, such that o is the parent of m_1 and n_1, while

1. *m_1 is the direct left sibling of n_1;*

2. *m_i is a nonterminal child of m_{i-1}, while all its right siblings are terminal siblings, $2 \leq i \leq r-1$, n_j is a nonterminal child of n_{j-1}, while all its left siblings are terminal siblings, $2 \leq j \leq s-1$;*

3. *if m_r is a terminal node, then all its siblings are terminal nodes; otherwise, all its right siblings are terminal siblings;*

4. *if n_s is a terminal node, then all its siblings are terminal nodes; otherwise, all its left siblings are terminal siblings;*

Then, α and β are two nonterminal neighboring paths *in t, α is a* left nonterminal neighboring path *to β, and β is a* right nonterminal neighboring path *to α.*

Definition 2. *A* general grammar *(GG for short) G is a quadruple $G = (V,\ T,\ P,\ S)$, where V is a* total alphabet*, T is a* terminal alphabet*, P is a finite* set of ruless *of the form $x \rightarrow y$, where $x, y \in V^*$,* $\text{alph}(x) \cap (V-T) \neq \emptyset$*, $S \in V-T$ is a* start symbol. *For every $u, v \in V^*$ and $x \rightarrow y \in P$, $uxv \Rightarrow uyv[p]$ or simply $uxv \Rightarrow uyv$ is a* derivation step *of G from uxv to uyv by the rule $x \rightarrow y$, $\Rightarrow$ is the* direct derivation *relation. Let $w_0, w_1, \ldots, w_n \in V^*$, for some $n \geq 1$, such that $w_0 \Rightarrow w_2\,[p1] \Rightarrow \ldots \Rightarrow w_n\,[p_n]$, where $p_i \in P$, for all $i = 1, \ldots, n$, then, $w_0 \Rightarrow^n w_n$; based on $\Rightarrow^n$, we define $\Rightarrow^+$ and $\Rightarrow^*$.*

A language *of G is $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$. G is* propagating *if $A \rightarrow x \in P$ implies $x \neq \varepsilon$. G is* context-free *if $A \rightarrow x \in P$ implies $A \in V-T$. G is GG with* linear core*, as already introduced in introduction, if any $p \in P$ has one of these forms,*

$$AB \rightarrow CD, A \rightarrow BC, A \rightarrow xEy$$

where $A, B, C, D \in V-T$, $E \in (V-T) \cup \{\varepsilon\}$, $x, y \in T^*$. In what follows, unless explicitly stated otherwise, we automatically assume that every GG has a linear core.
Similary, G is GG with left linear core *if any $p \in P$ has one of these forms,*

$$AB \rightarrow CD, A \rightarrow BC, A \rightarrow xE$$

where $A, B, C, D \in V-T$, $E \in (V-T) \cup \{\varepsilon\}$, $x \in T^*$.
G is GG in the Kuroda normal form *(KNF) [11] if every rule is one of these forms:*

$$AB \rightarrow CD, A \rightarrow BC, A \rightarrow a, A \rightarrow \varepsilon$$

where $A, B, C, D \in V-T$, $a \in T$.

Definition 3. *Let $G = (V, T, P, S)$ be a general grammar in KNF. Let $w \in T*$ be a string derived from G. A derivation tree for w is a labeled tree τ such that:*

1. *The root of τ is labeled with S.*

2. *Each leaf of τ is labeled with a symbol from T.*

3. *Each internal node of τ is labeled with a symbol from V.*

4. *If an internal node v is labeled with $A \in V$ and has children labeled B_1, B_2, then there exists a rule $A \rightarrow B_1B_2$ in P and, respectively, for the rest of the rules in KNF.*

5. *The yield of τ (that is, the concatenation of the labels on its leaves) is w.*

Example 1. *The following graph (Fig. 2) represents a labeled ordered tree t for a general grammar in KNF. Since any two distinct nodes have different labels, we refer to their labels below. The root node $\hat{r}$ is a. It has no parent and two children b and c. Then b is a sibling of c and c is a sibling of b. The leftmost child of b is d, while the rightmost is e. The node d is a left sibling of e, and is the direct left sibling, which is e. The node d is the parent of h, but h has no child, so it is a leaf node. $horsm = \text{frontier}(t)$. Consider the node e. The nodes a and b are predecessors of e, while i, j, o, p, and r are descendants of e. The nodes c or d are not in predecessor relation with e, as they are neither predecessors of e, nor descendants of e. The sequence of nodes $bejpr$ is a path in t. The path $cglqs$ is neighboring to $bejpr$; unlike $cglqs$, eio or bdh.*

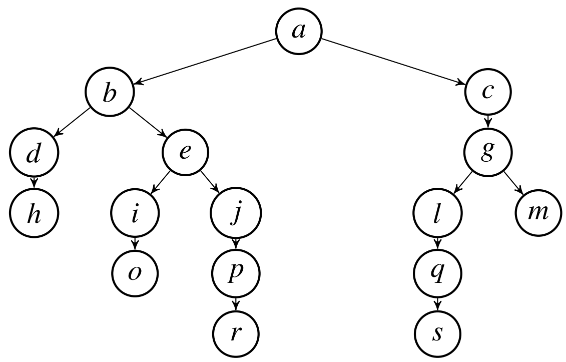

Figure 2: Labeled ordered tree t

Definition 4. *Let $G = (V, T, P, S)$ be a GG with linear core.*

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

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

     is obtained by replacing the ith non-ε-leaf in d labeled A, with the rule tree corresponding to p, $A\langle x\rangle$, where $i = |uA|$.
   - *(c) Let d be the derivation tree with $\text{frontier}(d) = uABv$ representing $X \Rightarrow^* uABv\,[\rho]$ and let $p\colon AB \to CD \in P$. The derivation tree that represents*

     $$X \Rightarrow^* uABv\,[\rho] \Rightarrow uCDv\,[p]$$

     is obtained by replacing the ith and $(i+1)$th non-ε-leaf in d labeled A and B with $A\langle C\rangle$ and $B\langle D\rangle$, respectively, where $i = |uA|$.
3. *A* derivation tree *in G is any tree t for which there is a derivation represented by t (see 2 in this definition).*

Note, after replacement in 2c, the nodes A and B are the parents of the new leaves C and D, respectively, and we say that A and B are context-dependent*, alternatively speaking, we say that there is a context dependency between A and B. In a derivation tree, two nodes are* context-independent *if they are not context-dependent.*

Then, for any $p\colon A \rightarrow x \in P$, ${}_G\triangle(p)$ *denotes the rule tree corresponding to p. 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 one of the derivation trees 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* ${}_G\triangle(A \Rightarrow^* x)$ *in what follows if there is no danger of confusion. Let* ${}_G\blacktriangle$ *denotes 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* $x \in L(G)$.

If a node is labeled with a terminal, it is called a terminal node*. If a node is labeled with a nonterminal, it is called a* nonterminal node*. If a node is labeled with a nonterminal and has two* nonterminal node *children, it is called a* branching nonterminal node*. Let* $\alpha = (o, m_1, m_2, \dots, m_r)$ *and* $\beta = (o, n_1, n_2, \dots, n_s)$ *be two neighboring paths, where* $r, s \geq 0$, α *is the left neighboring path to* β, *and* m_r *and* n_s *are terminal nodes. Then, there is a t-tuple* $\gamma = (g_1, g_2, \dots, g_t)$ *of nodes from* α *and t-tuple* $\delta = (h_1, h_2, \dots, h_t)$ *of nodes from* β, *where* $g_p < g_q$, *for* $1 \leq p < q \leq t$, $t < \min(r,s)$, *and* g_i *and* h_i *are context-dependent, for* $1 \leq i \leq t$. *Let* $\rho = p_1 p_2 \dots p_t$ *be a string of non-context-free rules corresponding to context dependencies between* γ *and* δ. *We call* ρ *the* right context of α *and the* left context of β *or the* context of α and β. *Consider a node* $m_i \in \alpha$, *where* $1 \leq i \leq r$, *and two* $(t-k+1)$*-tuples of nodes* $\sigma = (g_k, g_{k+1}, \dots, g_t)$ *and* $\varphi = (h_k, h_{k+1}, \dots, h_t)$, *where k is a minimal integer such that* $m_i < g_k$. *Then, a string of non-context-free rules* $\tau = p_k p_{k+1} \dots p_t$ *corresponding to context dependencies between* σ *and* φ *is called the* right descendant context of m_i, *for some* $1 \leq k \leq t$. *Analogously, we define the notion of the* left descendant context *of a node* n_j *in* β, *for some* $1 \leq j \leq s$.

Definition 5. *A labeled ordered tree t is* slow-branching *if any of its pairs of nonterminal neighboring paths contains no more than two nonterminal nodes having two nonterminal children and there is no reachable terminal node from nodes of the path between the root and any branching nonterminal node. A slow-branching labeled ordered tree is of* degree *k if it contains k branching nonterminal nodes,* $k \geq 1$.

Example 2. *Let* $G = (N, T, P, S)$ *be a general grammar, where* $N = \{S, X, Y\ Z, A_1, A_2, B, C_1, C_2, D_1, D_2, E_1, E_2, F_1, F_2\}$, $T = \{a, b, c, 0, 1\}$, *and P contains the following rules:*

(1) $S \rightarrow A_1 X$	(12) $C_1 C_2 \rightarrow F_1 F_2$
(2) $X \rightarrow A_2 Y$	(13) $D_1 \rightarrow 0 D_1$
(3) $Y \rightarrow BZ$	(14) $D_2 \rightarrow D_2 1$
(4) $Z \rightarrow C_1 C_2$	(15) $E \rightarrow 0E1$
(5) $A_1 \rightarrow a A_1$	(16) $F_1 \rightarrow 0 F_1$
(6) $A_2 \rightarrow A_2 a$	(17) $F_2 \rightarrow F_2 1$
(7) $B \rightarrow bBc$	(18) $D_1 \rightarrow \varepsilon$
(8) $C_1 \rightarrow a C_1$	(19) $D_2 \rightarrow \varepsilon$
(9) $C_2 \rightarrow C_2 b$	(20) $E \rightarrow \varepsilon$
(10) $A_1 A_2 \rightarrow D_1 D_2$	(21) $F_1 \rightarrow \varepsilon$
(11) $B \rightarrow E$	(22) $F_2 \rightarrow \varepsilon$

A graph representing ${}_G\triangle(S \Rightarrow^* aaa0011a0011b)$ *is illustrated in Fig. 3.*

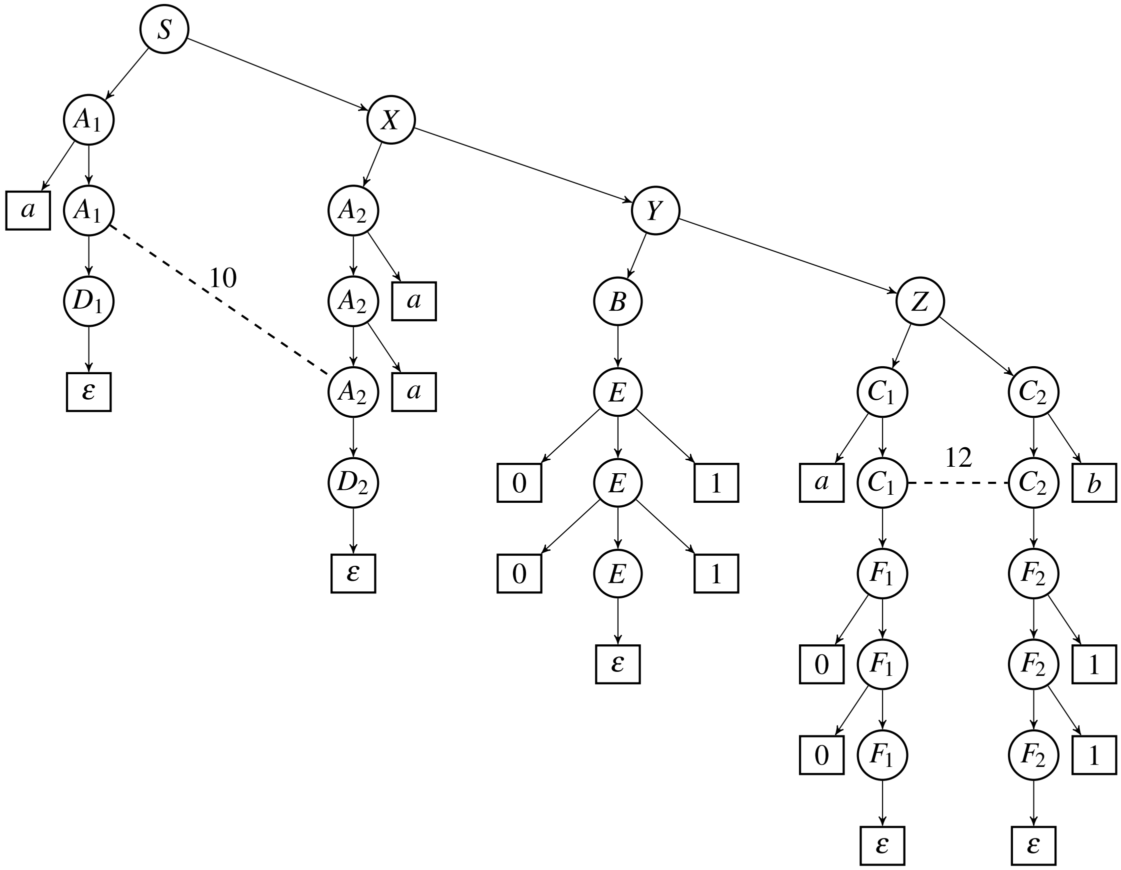

Figure 3: ${}_{G}\triangle_{aaa0011a0011b}$

Let us note that dashed lines and numbers contour only denote the context dependencies, and applied non-context-free rules, respectively, and are not the part of the derivation tree. The pairs of context-dependent nodes are linked with dashed lines, all the other nodes are context-independent.

Since $aaa0011a0011b = \mathrm{frontier}({}_{G}\triangle_{aaa0011a0011b})$*, all leaves are terminal nodes. Every other node is a nonterminal node.*

For a pair of neighboring paths $\alpha = SA_1A_1D_1\varepsilon$ *and* $\beta = SXA_2A_2A_2D_2\varepsilon$*, a string* $\rho = 10$ *is their context, it is the left context of* β *and the right context of* α*.*

4 Results

Theorem 1. *A language L is k-linear iff there is a constant* $k \geq 0$*, constant* $u \geq 0$ *and a general grammar G with linear core such that* $L = L(G)$ *and for every* $x \in L(G)$*, there is a slow-branching tree of degree* k ${}_{G}\triangle_{x} \in {}_{G}\blacktriangle$ *that satisfies:*

1. *any two nonterminal neighboring paths contain no more than u pairs of context-dependent nodes;*
2. *out of neighboring paths, any pair of nodes is context-independent.*

Proof. Construction. Consider any $u \geq 0$. Let $G = (V,T,P,S)$ be a GG such that $L(G) = L$. Set $N =$

$V - T$. Let $P_{cs} \subseteq P$ denote the set of all non-context-free rules of G. Set

$$N' = \{A_{l|r} \mid A \in N,\ l, r \in (P_{cs} \cup \{\varepsilon\})^u\}.$$

Construct a grammar $G' = (V', T, P', S_{\varepsilon|\varepsilon})$, where $V' = N' \cup T$. Set $P' = \emptyset$. Construct P' by performing I through III given next.

(I) For all $A \to xEy \in P, A \in N, E \in N \cup \{\varepsilon\}$, $x, y \in T^*$, and $l, r \in (P_{cs} \cup \{\varepsilon\})^u$, add $A_{l|r} \to xE_{l|r}y$ to P';

(II) for all $A \to BC \in P$, where $A, B, C \in N$, and $r, l, x \in (P_{cs} \cup \{\varepsilon\})^u$, add $A_{l|r} \to B_{l|x}C_{x|r}$ to P';

(III) for all $p\colon AB \to CD \in P, A, B, C, D \in N, x, z \in (P_{cs} \cup \{\varepsilon\})^u$, and $y \in (P_{cs} \cup \{\varepsilon\})^{u-1}$, add $A_{x|py} \to C_{x|y}$ and $B_{py|z} \to D_{y|z}$ to P'.

Basic idea. Notice nonterminal symbols. Since every pair of neighboring paths of G contains a limited number of context-dependent nodes, all of its context-dependencies are encoded in nonterminals. G' nondeterministically decides about all context-dependencies while introducing a new pair of neighboring paths by rules from (II). A new pair of neighboring paths is introduced with every application of

$$A_{l|r} \to B_{l|x}C_{x|r},$$

where x encodes a new descendant context. Context dependencies are realized later by context-free rules from (III).

Since P' contains no non-context-free rule and G' is context-free. Next, we prove $L(G) = L(G')$ by establishing Claims 1 through 3. Define the new homomorphism $\gamma\colon V' \to V$, $\gamma(A_{l|r}) = A$, for $A_{l|r} \in N'$, and $\gamma(a) = a$ otherwise.

Claim 1. *If $S \Rightarrow^m w$ in G, where $m \geq 0$ and $w \in V^*$, then $S_{\varepsilon|\varepsilon} \Rightarrow^* w'$ in G', where $w' \in V'^*$ and $\gamma(w') = w$.*

Proof. We prove this by induction on $m \geq 0$.

Basis. Let $m = 0$. That is $S \Rightarrow^0 S$ in G. Clearly, $S_{\varepsilon|\varepsilon} \Rightarrow^0 S_{\varepsilon|\varepsilon}$ in G', where $\gamma(S_{\varepsilon|\varepsilon}) = S$, so the basis holds.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that Claim 1 holds for all $0 \leq m \leq n$.

Induction Step. Let $S \Rightarrow^{n+1} w$ in G. Then, $S \Rightarrow^n v \Rightarrow w$, where $v \in V^*$, and there exists $p \in P$ such that $v \Rightarrow w\,[p]$. By the induction hypothesis, $S_{\varepsilon|\varepsilon} \Rightarrow^* v'$, where $\gamma(v') = v$, in G'. Next, we consider the following three forms of p.

(I) Let $p\colon A \to xEy \in P$, for some $A \in N$, $E \in N \cup \{\varepsilon\}$, $x, y \in T^*$. Without any loss of generality, suppose l and r are a left descendant context and a right descendant context of A. By the construction of G', there exists a rule $p'\colon A_{l|r} \to xE_{l|r}y \in P'$, where $A_{l|r} \in v'$. Then, there exists a derivation $v' \Rightarrow w'\,[p']$ in G', where $\gamma(w') = w$.

(II) Let $p\colon A \to BC \in P$, for some $A, B, C \in N$. Without any loss of generality, suppose l and r are a left descendant context and a right descendant context of A, and $x \in (P_{cs} \cup \{\varepsilon\})^u$ is a context of neighboring paths beginning at this node. By the construction of G', there exists a rule $p'\colon A_{l|r} \to B_{l|x}C_{x|r} \in P'$, where $A_{l|r}, B_{l|x}, C_{x|r} \in v'$. Then, there exists a derivation $v' \Rightarrow w'\,[p']$ in G', where $\gamma(w') = w$.

(III) Let $p\colon AB \to CD \in P$, for some $A,B,C,D \in N$. By the assumption stated in Theorem 1, A and B occur in two neighboring paths denoted by α and β, respectively. Without any loss of generality, suppose that a context of α and β is a string $c \in (P_{cs} \cup \varepsilon)^u$, where $c = pcd$, and l is a left descendant context, r is a right descendant context of A, B, respectively. By the construction of G', there exist two rules

$$p'_l\colon A_{l|pcd} \to C_{l|cd},\ \ p'_r\colon B_{pcd|r} \to D_{cd|r} \in P',$$

where $A_{l|pcd}, C_{l|cd}, B_{pcd|r}, D_{cd|r} \in V'$. Then, there exists a derivation $v' \Rightarrow^2 w'\,[p'_l p'_r]$ in G', where $\gamma(w') = w$.

Notice (III). The preservation of the context is achieved by nonterminal symbols. Since the stored context is reduced symbol by symbol from left to right direction in both α and β, G' simulates the applications of non-context-free rules of G.

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

Claim 2. *Every $x \in L(G')$ can be derived in G' as follows.*

$$S_{\varepsilon|\varepsilon} = x_0 \Rightarrow^{d_1} x_1 \Rightarrow^{d_2} x_2 \Rightarrow^{d_3} \cdots \Rightarrow^{d_{h-1}} x_{h-1} \Rightarrow^{d_h} x_h = x,$$

for some $h \geq 0$, where $d_i \in \{1,2\}$, $1 \leq i \leq h$, so that

1. *if $d_i = 1$, then $x_{i-1} = uA_{l|r}v$, $x_i = uzv$, $x_{i-1} \Rightarrow x_i\ [A_{l|r} \to z]$, where $u,v \in V'^*$, $z \in \{E_{l|r}, B_{l|r}, C_{l|x}D_{x|r}, x, y\}$, for some $A_{l|r}, B_{l|r}, C_{l|x}, D_{x|r} \in N'$, $E_{l|r} \in (N' \cup \varepsilon)$, $x,y \in T^*$;*

2. *if $d_i = 2$, then $x_{i-1} = uA_{x|py}B_{py|z}v$, $x_i = uC_{x|y}D_{y|z}v$, and*

$$uA_{x|py}B_{py|z}v \Rightarrow uC_{x|y}B_{py|z}v\ [A_{x|py} \to C_{x|y}] \Rightarrow uC_{x|y}D_{y|z}v\ [B_{py|z} \to D_{y|z}],$$

for some $u,v \in V'^*$ and $A_{x|py}, B_{py|z}, C_{x|y}, D_{y|z} \in N'$.

Proof. Since G' is context-free, without any loss of generality in every derivation of G' we can always reorder applied rules to satisfy Claim 2. □

Claim 3. *Let $S_{\varepsilon|\varepsilon} \Rightarrow^{d_1} x_1 \Rightarrow^{d_2} \cdots \Rightarrow^{d_{m-1}} x_{m-1} \Rightarrow^{d_m} x_m$ in G' be a derivation that satisfies Claim 2, for some $m \geq 0$. Then, $S \Rightarrow^* w$ in G, where $\gamma(x_m) = w$.*

Proof. We prove this by induction on $m \geq 0$.

Basis. Let $m = 0$. That is $S_{\varepsilon|\varepsilon} \Rightarrow^0 S_{\varepsilon|\varepsilon}$ in G'. Clearly, $S \Rightarrow^0 S$ in G. Since $\gamma(S_{\varepsilon|\varepsilon}) = S$, the basis holds.

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

Induction Step. Let $S_{\varepsilon|\varepsilon} \Rightarrow^{d_1} x_1 \Rightarrow^{d_2} \cdots \Rightarrow^{d_{n-1}} x_{n-1} \Rightarrow^{d_n} x_n \Rightarrow^{d_{n+1}} x_{n+1}$ in G' be a derivation that satisfies Claim 2. By the induction hypothesis, $S \Rightarrow^* v$, $v \in V^*$, where $\gamma(x_n) = v$, in G. Divide the proof into two parts according to d_{n+1}.

(A) Let $d_{n+1} = 1$. By the construction of G', there exists a rule $p' \in P'$ such that $x_n \Rightarrow^{d_{n+1}} x_{n+1}\,[p']$. Next, we consider the following two forms of p'.

(I) Let $p'\colon A_{l|r} \to xE_{l|r}y \in P'$, for some $A \in N$, $E \in (N \cup \varepsilon)$, $x,y \in T^*$ and $l,r \in (P_{cs} \cup \{\varepsilon\})^u$. By the construction of G', rule p' was introduced by some rule $p\colon A \to xEy \in P$. Then, there exists a derivation $v \Rightarrow w\,[p]$, where $\gamma(x_{n+1}) = w$.

(II) Let $p' \colon A_{l|r} \to B_{l|x}C_{x|r} \in P'$, for some $A, B, C \in N$ and $l, r, x \in (P_{cs} \cup \{\varepsilon\})^u$. By the construction of G', rule p' was introduced by some rule $p \colon A \to BC \in P$. Then, there exists a derivation $v \Rightarrow w\,[p]$, where $\gamma(x_{n+1}) = w$.

(B) Let $d_{n+1} = 2$. Then, $x_n \Rightarrow^{d_{n+1}} x_{n+1}$ is equivalent to

$$u_1 A_{x|py} B_{py|z} u_2 \Rightarrow u_1 C_{x|y} B_{py|z} u_2\ [p'_1] \Rightarrow u_1 C_{x|y} D_{y|z} u_2\ [p'_2],$$

where $x_n = u_1 A_{x|py} B_{py|z} u_2$, $x_{n+1} = u_1 C_{x|y} D_{y|z} u_2$, and

$$p'_1 \colon A_{x|py} \to C_{x|y},\ p'_2 \colon B_{py|z} \to D_{y|z} \in P',$$

for some $u_1, u_2 \in V'^*$ and $A_{x|py}$, $B_{py|z}$, $C_{x|y}$, $D_{y|z} \in N'$. By the construction of G', rules p'_1 and p'_2 were introduced by some rule $p \colon AB \to CD \in P$, Then, there exists a derivation $v \Rightarrow w\,[p]$, where $\gamma(x_{n+1}) = w$.

We covered all possibilities, so the claim holds. □

Observe that the derivation trees of the constructed context-free G' remain slow-branching.

Claim 4. *The grammar G' is k-linear.*

Proof. In construction (III) we replace the rules of the form $AB \to CD$ with the rules of the form $A \to B$, where $A, B, C, D \in N$. Therefore, only the rules that are allowed to occur in the derivation G' before the rules of the form $A \to BC$ are the rules of the form $A \to B$. Rules of the form $A \to B$ before the rules of the form $A \to BC$ can be omitted by the trivial transformation of G', similar to the algorithm on elimination of unit productions from Section 5 in [11]. Therefore, the grammar G' is k-linear. □

By Claim 4 G' is k-linear. By Claims 1 and 3, $S \Rightarrow^* w$ in G iff $S_{\varepsilon|\varepsilon} \Rightarrow^* w'$ in G', where $\gamma(w') = w$. If $S \Rightarrow^* w$ in G and $w \in T^*$, then $w \in L(G)$. Since $\gamma(w') = w' = w$, for $w \in T^*$, $w' \in L(G')$. Therefore, $L(G) = L(G')$ and Theorem 1 hold. □

Consider Theorem 1. Observe that the 2nd condition is superfluous whenever G is propagating.

Theorem 2. *A language L is k-linear iff there is a constant $k \geq 0$, constant $u \geq 0$ and a propagating general grammar G with linear core such that $L = L(G)$ and for every $x \in L(G)$, there is a slow-branching tree of degree k $\triangle_x \in {}_G\blacktriangle$, where any two nonterminal neighboring paths contain no more than u pairs of context-dependent nodes.*

Proof. Prove this by analogy with the proof of Theorem 1. □

Theorem 3. *A language L is regular iff there is a constant $u \geq 0$ and a general grammar G with left linear core such that $L = L(G)$ and for every $x \in L(G)$, there is a tree $\triangle_x \in {}_G\blacktriangle$ that satisfies:*

1. *any two nonterminal neighboring paths contain no more than u pairs of context-dependent nodes;*
2. *out of neighboring paths, any pair of nodes is context-independent.*

Proof. Prove this by analogy with the proof of Theorem 1. □

Theorem 4. *A language L is regular iff there is a constant $u \geq 0$ and a propagating general grammar G with a linear left core such that $L = L(G)$ and for every $x \in L(G)$, there is a tree $\triangle_x \in {}_G\blacktriangle$, where any two nonterminal neighboring paths contain no more than u pairs of context-dependent nodes.*

Proof. Prove this by analogy with the proof of Theorem 1. □

5 Application

In this section, we explain how to apply the results achieved in the previous section in order to demonstrate the metalinearness (or regularity) of a language, L. As a rule, this demonstration follows the next three-step proof scheme for metalinearness.

1. Construct a general grammar G with linear core.
2. Prove $L(G) = L$.
3. Prove that G satisfies conditions from Theorem 2 or Theorem 1, depending on whether G is context-sensitive.

For regularity, we use a similar three-step scheme as following.

1. Construct a general grammar G with left linear core.
2. Prove $L(G) = L$.
3. Prove that G satisfies conditions from Theorem 3 or Theorem 4 depending on whether G is context-sensitive.

Reconsider the grammar G from Example 2. Following the proof scheme sketched above, we next prove that $L(G)$ is k-linear. Without any loss of generality, every terminal derivation of G can be divided into the following 5 phases, where each rule may be used only in a specific phase:

(a) 1–4 (b) 5–9 (c) 10–12 (d) 13–17 (e) 18–22

Next, we describe these phases in greater detail.

(a) First, we generate the following string by rules 1 though 4.

$$A_1A_2BC_1C_2$$

Possibly applicable rules from b and c may be postponed to the next phases without affecting the derivation, since the rules in the previous phases cannot rewrite the nonterminals of the following phases.

(b) The rules 5 through 9 are context-free rules and nonterminals on the left-hand side of the rule are the same as on the right-hand side of the rule. Therefore, they are grouped into (b), since they only generate terminals. Possibly applicable rules from (c) may be postponed for the phase (c) without affecting the derivation since the rules in the previous phases cannot rewrite nonterminals from the following phases.

$$a^*A_1A_2a^*b^*Bc^*a^*C_1C_2b^*.$$

(c) The rules 10 and 12 are non context-free rules. The rules 10 through 12 are all rules without generating terminals. For the same reason as in (a) rules 1 to 4 from the phases (d) and (e) can be postponed to respective phases.

$$a^*D_1D_2a^*b^*Ec^*a^*F_1F_2b^*.$$

(d) The rules 13 through 17 are alike rules in (b)

$$a^*0^*D_1D_21^*a^*b^*0^*E1^*c^*a^*0^*F_1F_21^*b^*.$$

(e) Since rules 18 and 22 are erasing rules and they can always be postponed until the end of any successful derivation.

$$a^*0^*1^*a^*b^*0^*1^*c^*a^*0^*1^*b^*.$$

Grammar G is obviously a general grammar with the linear core.

Only rules in the step (a) include branching of nonterminals, no terminals are generated and the branching in the step (a) is a slow-branching. Therefore, the slow-branching condition is fulfilled.

Let us now show that for any $x \in L(G)$, there is ${}_G\triangle_x \in {}_G\blacktriangle$, where any two neighboring paths contain no more than a 1 pair of context-dependent nodes.

Every pair of context-dependent nodes in ${}_G\triangle_x$ corresponds to one non-context-free rule in $S \Rightarrow^* x$. Consider the five phases sketched above. Observe that all phases except (c) contain only non context-free rules, so we only have to investigate (c). On the other hand, (c) contain no rule of the form $A \to BC$, thus the number of neighboring paths remains unchanged.

In (c) rule 10 and 12 introduce context dependency between two pairs of neighboring paths. After the application of these two rules, we cannot reach the nonterminals again on the left-hand side of rules 10 and 12. Therefore, these context-dependencies can occur only once between a pair of neighboring paths.

No other non-context-free rule is applied; therefore, no other context-dependent pair of nodes can occur. Then, every pair of neighboring paths may contain at most one context-dependent pair of nodes introduced in phase (c).

Since G is a general grammar with linear core, where for every $x \in L(G)$, there is ${}_G\triangle_x \in {}_G\blacktriangle$, where any two neighboring paths contain no more than 1 pair of context-dependent nodes, by Theorem 1, $L(G)$ is k-linear.

Corollary

Next, we show that the proposed grammars with linear core have the same generative power as GGs.

Theorem 5. *A language L is recursively enumerable iff $L = L(G)$, where G is a general grammar with linear core.*

Proof. Every language L generated by a general grammar with linear core G is recursively enumerable, because every general grammar with linear core can be trivially converted to KNF. In other direction, every KNF G is a GG with linear core by Definition 2.

□

Theorem 6. *A language L is context-sensitive iff $L = L(G)$, where G is a propagating general grammar with linear core.*

Proof. Every language L generated by propagating general grammar with linear core G is context sensitive, because each rule in P, where $P \in G$, is a form of $x \to y$ and $|x| \le |y|$.

□

Open problem

Let us consider a more lenient definition of slow-branching tree as follows.

Definition 6. *A labeled ordered tree t is* slow-branching *if any of its pairs of nonterminal neighboring paths contains no more than two nonterminal nodes having two nonterminal children. A slow-branching labeled ordered tree is of* degree *k if it contains k branching nonterminal nodes,* $k \geq 1$.

It is obvious that the newly provided Definition 6 is insufficient to prove that a grammar restricted by a slow-branching derivation tree is k-linear. However, it is possible to apply different restrictions to the Definition 6 with its own advantages or demonstrate similar result to Theorem 1 to prove that it is k-linear. Such a discovery would require further studies.

Acknowledgments

This work was supported by the BUT grant FIT-S-23-8209.

References

[1] A.V. Aho & J.D. Ullman (1972): *The Theory of Parsing, Translation, and Compiling*. Prentice-Hall, Series in Automatic Computation.

[2] Brenda S. Baker (1974): *Non-context-free grammars generating context-free languages*. Information and Control 24(3), pp. 231–246.

[3] Ronald V. Book (1972): *Terminal Context in Context-Sensitive Grammars*. SIAM J. Comput. 1(1), p. 20–30.

[4] Ronald V Book (1973): *On the structure of context-sensitive grammars*. International Journal of Computer & Information Sciences 2, p. 129–139.

[5] T.H. Cormen, C.E. Leiserson & R.L. Rivest (2002): *Introduction to Algorithms (Appendix B.5)*. McGraw-Hill.

[6] A. Ehrenfeucht, D. Haussler & G. Rozenberg (1983): *On regularity of context-free languages*. Theoretical Computer Science 27(3), pp. 311–332.

[7] M. A. Harrison (1978): *Introduction to Formal Language Theory*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[8] G.H. Matthews (1964): *A note on asymmetry in phrase structure grammars*. Information and Control 7(3), pp. 360–365.

[9] G.H. Matthews (1967): *Two-way languages*. Information and Control 10(2), pp. 111–119.

[10] A. Meduna (2014): *Formal Languages and Computation: Models and Their Applications*. Taylor & Francis, New York. ISBN: 978-1466513457.

[11] Alexander Meduna (2000): *Automata and languages: theory and applications*. Springer-Verlag, Berlin, Heidelberg.

[12] Alexander Meduna & Ondřej Soukup (2017): *Modern Language Models and Computation: Theory with Applications*. Springer US, doi:10.1007/978-3-319-63100-4.

[13] G. Rozenberg & A. Salomaa (1997): *Handbook of Formal Languages, Vol. 1: Word, Language, Grammar*. Springer, New York.

[14] Grzegorz Rozenberg & Arto Salomaa, editors (1997): *Handbook of Formal Languages, Volume 1: Word, Language, Grammar*. Springer.

[15] A. Salomaa (1973): *Formal Languages*. Academic Press, London.