VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

GRAMMARS WITH RESTRICTED DERIVATION TREES

GRAMATIKY S OMEZENÝMI DERIVAČNÍMI STROMY

EXTENDED ABSTRACT OF PHD THESIS
ROZŠÍŘENÝ ABSTRAKT DISERTAČNÍ PRÁCE

AUTHOR Ing. JIŘÍ KOUTNÝ
AUTOR PRÁCE

SUPERVISOR prof. RNDr. ALEXANDER MEDUNA, CSc.
VEDOUCÍ PRÁCE

BRNO 2012

Contents

Chapter 1

Introduction

The *formal language theory* is an inherent part of the *theoretical computer science* particularly concerned with the study of the formal models. The formal models are mathematical objects used to describe the formal languages. The fundamental models include grammars and automata. The former are used to generate words and the latter accept them.

Grammars are the kind of the *rewriting models* that start from a specified symbol (i.e., start symbol). Then, the symbol is modified according to the given set of rewriting productions. Each production is composed of two components—the left-hand side and the right-hand side of a production. The application of a production on a word is done by rewriting a symbol equivalent to the left-hand side of a production by its right-hand side in the word. This process is known as a *derivation step.* During the computation of a derivation step, just one symbol is rewritten in the word. Given a start symbol of a grammar, a derivation step is computed repeatedly by applying the productions from the given set. Once the resulting word is composed of the symbols that cannot be rewritten anymore, the process of applying derivation steps ends and the resulting word belongs to the language of the grammar.

Essentially, the grammars are composed of a finitely many symbols that are rewritten by finitely many production in finitely many derivation steps. In this way, the grammars represent a finite description of even infinite languages. By the notion of infinite languages are meant those languages that contains infinitely many words. Since the most of the languages commonly used in practice are infinite, the grammars represent a powerful tool how to deal with them. In the formal language theory, there exists a huge variety of grammars which essentially differ in two domains. Specifically, in the complexity of the productions and in the way how to select appropriate production to be applied in a derivation step.

Generally, the complexity of rewriting productions can be seen from two angles—theoretical and practical.

- Theoretical viewpoint: As little as possible restrictions placed on the form of the rewriting productions in a rewriting model is desirable. More specifically, the more complex rewriting productions are, the larger class of languages may the model generate. In other words, to generate complex languages, complex productions are needed. By the notion of a form of a production, namely the number of the symbols on its left-hand side is meant.

- Practical viewpoint: Grammars are theoretical models that are implemented in many practical applications. From the perspective of cost-effective implementation, the simple rewriting productions are desirable. As simple-enough productions for effective implementation, those of the form with just one symbol on their left-hand side are considered. Such kind of productions are referred to as *context-free productions*, since they can be applied without any consideration of a context of currently rewritten symbol.

Non-regulated rewriting models like grammars and automata belong to the well-known core of the formal language theory and they are frequently used in practice. Indeed, automata including its variants underlie lexical analysers (see [3] and [36]), context-free grammars represent the basis of both top-down as well as bottom-up parsers (see [3] and [4]), etc. However, the power of the models with simple productions is indeed smaller than required for usage in many practical applications. On the other hand, the models that use only simple productions are usually easier to implement. As a background, it is desirable to extend context-free grammars as in many applications there are some natural phenomena which cannot be captured by context-free rewriting. More precisely, the motivation is based on the observation that many of the languages commonly used in practice, including programming and natural languages, are not context-free (see [15], [16], [41], [42], and [46]). Consequently, that means such languages cannot be generated by a grammar with only context-free productions. For these reasons, the idea whether or not it would be possible to use grammars with only context-free productions and increase the corresponding power in some other way—without changing the form of rewriting productions. Basically, this can be achieved by two fundamental approaches—using a kind of a regulation of rewriting or using more than one grammar with context-free productions in a model:

- Using a kind of a regulation of rewriting. By the notion of a *regulation*, the way how to select appropriate production to be applied is meant. Indeed, a situation is common in which, given a word, it is possible to apply several productions. Informally, the essential idea is represented by the observation that a regulation mechanism somehow prescribes the order of productions the grammar must follow. Therefore, many different kinds of such a regulation have been introduced in order to ensure selecting appropriate production. All of the resulting models based on a kinds of regulation are collectively referred to as *regulated rewriting models*.

- Using more than one grammar with context-free productions in a model. Roughly speaking, the main underlying idea is based on the observation that from the cooperation of several simple models, we can obtain more power than from each of them if they work separately. These systems were also thoroughly studied and the corresponding investigation area is referred to as the *theory of grammar systems*. However, we will deal with them only rarely in this work.

Informally, the goal of this work is to introduce a model that generates more than context-free languages and is usable in practice. From the theoretical viewpoint it means, the model should be able to generate namely the programming languages and the languages used in linguistics (e.g., multiple repetition, cross-dependencies, and copy-language). From the practical viewpoint, it should be possible to develop sophisticated parsing methods working in a polynomial time for the model.

One way to extend the power of context-free grammars is to consider *context-sensitive grammars* where the productions are more complex. Indeed, context-sensitive grammars contain the productions with even more than a single symbol on the left-hand side. However, despite their great power, generating complex languages by context-sensitive grammars actually leads to several fundamental problems making their practical usage problematic (see [6], [8], [35], [43], [51], and [52]). Specifically, for context-sensitive grammars, many problems are undecidable, it is difficult to describe the derivation by a graph structure, etc.

One of the many others approaches extending the power of context-free grammars is represented by *matrix grammars* introduced by Abraham in [1]. The fundamental underlying principle in a derivation step in matrix grammars is that not just one but a fixed number of context-free productions are required to be applied in a given order. This provides synchronization among different parts of a generated word and many non-context-free languages can be generated in this way (see [38], [45], and[53]).

There are lots of other well-known approaches for extending the power of context-free grammars which preserve the context-free nature of productions. Specifically, *Random Context Grammars* (see [50]), *Programmed Grammars* (see [44]), *Ordered Grammars* (see [14]), *Indian and Russian Parallel Grammars* (see [28]), *Indexed Grammar* (see [2]), and many others. However, these approaches do not represent the main topic of this work although some connections can probably be found.

1.1 Derivation Tree Restricted Models

One of the power-extending approaches is represented by the restrictions placed upon the derivation trees. Given a grammar, by the notion of a *derivation tree*, a graph structure depicting the application of productions on the start symbol up to the resulting word is meant. Indisputably, the investigation of context-free grammars with restricted derivation trees represents an important trend in today's formal language theory (see [7], [9], [11], [13], [27], [20], [22], [23], [25], [32], [33], [34], [37], [39], [48], and [49]). In essence, these grammars generate their languages just as ordinary context-free grammars do, but their derivation trees must satisfy some simple prescribed conditions.

The following two sections give an informal overview of the results related to the investigation of derivation-tree-restricted grammars. Based on this informal description, Chap. 2 being a strictly formal summary of the results presented here. The definitions needed just to present the results of the other authors are omitted in this work. However, the appropriate references for the definitions are given. Through this section, it is assumed the reader is familiar with the fundamentals of the formal language theory. Several results concerning the derivation-tree-restricted models related to L-systems (see [17], [18], [19], [31], [29], [30], and [40]) are not included in this work since the investigation presented in this short dissertation thesis focuses rather on the sequential (i.e., grammars) than parallel rewriting (i.e., L-systems).

1.1.1 Level Based Restriction

The idea of restrictions placed upon the derivation trees of context-free grammars is introduced by Culik and Maurer in [9] and the resulting grammars restricted in this way are referred to as *tree controlled grammars* (see Def. 2.1). In essence, the notion of a tree controlled grammar is defined as follows: take a context-free grammar, G, and a regular language, R. A word, w, generated by G belongs to the language defined by G and R if there is a derivation tree, t, for w in G such that all levels of t (except the last one) are described by R. Given a tree controlled grammar, (G, R), G and R are referred to as controlled grammar and control language, respectively. Culik and Maurer investigate several basic properties of tree controlled grammars—namely, the membership problem (see Th. 2.1) and the generative power (see Th. 2.2, Th. 2.3, Th. 2.4, Col. 2.5, and Th. 2.6).

Based on the original definition of a tree controlled grammar, Păun studies the modifications where many well-known types of both controlled grammars and control languages are considered in [39]. More precisely, Păun studies controlling the levels of the derivation trees of a regular grammar by several types of a control language (see Th. 2.7 and Col. 2.8), controlling the levels of the derivation trees of a context-free grammar without erasing productions by several types of a control language (see Th. 2.9, Th. 2.10, Col. 2.11, Th. 2.12, and Col. 2.13), and controlling the levels of the derivation trees of a context-free grammar by a finite language (see Th. 2.14 and Col. 2.15).

It is well-known that tree controlled grammars with a context-free grammar controlled by a regular language characterize the class of recursively enumerable languages (see Th. 2.5). Thus, the question arises whether or not it is possible to achieve the same generative power as tree controlled grammars have when the levels of the derivation trees are restricted by a subregular control language. This problem is studied by Dassow and Truthe in [13], where many types of subregular languages are considered. Dassow and Truthe study primarily controlling the levels of the derivation trees of a context-free grammar by two types of a language such that one is a subset of the other (see Lem. 2.16) and controlling the levels of the derivation trees of a context-free grammar by many different types of subregular languages (see Th. 2.17, Th. 2.18, Th. 2.19, Col. 2.20, Th. 2.21, Th. 2.22, Th. 2.23, and Th. 2.24). The same authors, Dassow and Truthe, also study hierarchies of subregularly tree controlled languages in [11] and [12]. They present controlling the levels of the derivation trees of a context-free grammar by the union of monoids (see Th. 2.25), by regular languages with restricted descriptional complexity (see Lem. 2.26, Th. 2.27, and Th. 2.28), and by the language accepted by a deterministic finite automaton with at most given number of states (see Th. 2.29).

Stiebe in [47] states that there is a tree controlled grammar for every linearly bounded queue automaton (see Lem. 2.30). Then, Stiebe proves that controlling the levels of the derivation trees of a context-free grammar by the language accepted by a minimal finite automaton with at most five states characterize the class of context-sensitive languages (see Th. 2.31). If, additionally, erasing productions in a controlled grammar are allowed, controlling the levels of the derivation trees of a context-free grammar by the language accepted by a minimal finite automaton with at most five states characterizes the class of recursively enumerable languages (see Th. 2.32).

Turaev, Dassow, and Selamat in [48] examine tree controlled grammars with bounded nonterminal complexity and demonstrate that nine/seven nonterminals in a tree controlled grammar are enough to generate any recursively enumerable language (see Th. 2.33 and Th. 2.34). Then, they establish that three nonterminals in a tree controlled grammar are enough to generate any regular language (see Th. 2.35) and any regular simple matrix language can be generated by a tree controlled grammar (see Th. 2.36) with three nonterminals. Finally, they demonstrate that three nonterminals in a tree controlled grammar are enough to generate any linear language (see Th. 2.37). The same authors in [49] state several further nonterminal-complexity-related properties of tree controlled grammars (see Lem. 2.38).

A strictly formal summary of the results concerning tree controlled grammars can be found in Chap. 2.1.

1.1.2 Path Based Restriction

As an attempt to increase the power of context-free grammars without changing the basic formalism and loosing some basic properties of context-free languages (decidability, efficient parsing, etc.), Marcus, Martín-Vide, Mitrana, and Păun in [32] study a new type of a restriction in a derivation: a derivation tree in a context-free grammar is accepted if it contains a path described by a control language. More precisely, they consider two context-free grammars, G and G'. A word, w, generated by G belongs to the language defined by G and G' if there is a derivation tree, t, for w in G such that there exists a path of t described by the language of G'. Based on this restriction, they introduce a *path controlled grammar* (see Def. 2.7) and study several properties of this model. Specifically, they study controlling a path of the derivation trees of several types of grammars by a regular language (see Th. 2.39) and controlling a path of the derivation trees of a regular grammar by a linear or context-free language (see Th. 2.41 and Th. 2.42). Then, they establish two kinds of pumping properties depending on the type of a controlled grammar (see Th. 2.44, and Th. 2.45), some consequences to the closure properties of path controlled grammars (see Th. 2.47, Th. 2.48, Th. 2.49), and a basic parsing property for path controlled grammars (see Th. 2.50). They also investigate the generative power of path controlled grammars (see Th. 2.43 and Col. 2.46). However, there exists a serious problem with the correctness of the proof they present.

As a continuation of the investigation of path-based restrictions, Martín-Vide and Mitrana study parsing properties of path controlled grammars (see Th. 2.51), closure properties of path controlled grammars (see Th. 2.52, Th. 2.53, and Th. 2.54), and several decision problems for path controlled grammars (see 2.55) in [33] and [34].

For a strictly formal summary of the results related to path controlled grammars, see Chap. 2.2.

1.2 Goals of the Thesis

As it clearly follows from the previous sections, level-based restriction is well-studied and the most of the important questions have been answered. On the other hand, in the case of path-based restriction many basic properties including the generative power have not been successfully investigated yet. Moreover, several other restrictions placed upon the derivation trees have not yet been introduced at all. Indeed, the restrictions placed upon the cuts of the derivation tree as well as upon several paths of the derivation trees represent completely new investigation areas. Thus, the goals of the doctoral thesis consist in three investigation areas.

- First, to introduce new investigation area represented by cut-based restrictions and establish the generative power of the model restricted in this way.
- Second, to establish several new results in the investigation of one-path-restricted grammars introduced in [32].
- Third, to generalize one-path-restricted model into several paths and investigate several its properties.

1.2.1 Preliminaries

For the state of the art as well as the new results presented in the following chapter, it is assumed that the reader is familiar with the graph theory (see [5]) and the theory of formal languages (see [35]), including the theory of regulated rewriting (see [10]). However, for better readability, consider the following preliminaries.

Definition 1.1. For a sequence, s, of the nodes of a derivation tree, the *word obtained by concatenating all symbols of* s is denoted as $\text{word}(s)$.

Definition 1.2 (Classes of combinational, definite, nilpotent, commutative, circular, suffix-closed, non-counting, power-separating, and ordered languages.)**.** The *class of combinational*, *definite*, *nilpotent*, *commutative*, *circular*, *suffix-closed*, *non-counting*, *power-separating*, and *ordered languages* are defined as **COMB**, **DEF**, **NIL**, **COM**, **CIRC**, **SUF**, **NC**, **PS**, **ORD** = $\{L$: L is a combinational, definite, nilpotent, commutative, circular, suffix-closed, non-counting, power-separating, and ordered language$\}$, respectively.

Definition 1.3 (Size of regular language)**.** Let R be a regular language. The *size* of R is denoted as $c(R)$ and defined as the number of states of a minimum-state finite automaton that accepts R. For any natural number $n \geq 1$, let $\mathbf{REG}_n$ be the class of all regular languages R such that $c(R) \leq n$.

Definition 1.4 (Target sets of monoids)**.** Let V be an alphabet. Then, **MON** denote the class of all languages of the form V^*. For any natural number $n \geq 1$, let $\mathbf{MON}_n$ be the class of all languages that can be represented in the form $V_1^* \cup V_2^* \cup \cdots \cup V_k^*$ with $1 \leq k \leq n$ where all V_i are alphabets, for $1 \leq i \leq k$.

Definition 1.5. The class of regular, linear, ε-free context-free, context-free, propagating scattered context grammars, matrix grammars, context-sensitive, and unrestricted grammars is denoted as **REG**, **LIN**, $\mathbf{CF}_\varepsilon$, **CF**, **PSC**, **MAT**, **CS**, **RE**, respectively.

Definition 1.6. The set of all regular, linear, ε-free context-free, context-free, context-sensitive, and unrestricted grammars is denoted as $\mathbb{G}_{\mathbf{REG}}$, $\mathbb{G}_{\mathbf{LIN}}$, $\mathbb{G}_{\mathbf{CF}_\varepsilon}$, $\mathbb{G}_{\mathbf{CF}}$, $\mathbb{G}_{\mathbf{CS}}$, and $\mathbb{G}_{\mathbf{RE}}$, respectively.

Chapter 2

State of the Art Survey

This chapter being the formal summary of the crucial derivation-tree-based restriction results. All proofs related to the presented results are omitted. However, appropriate reference where one can found the proof if required is always included.

2.1 Tree Controlled Grammars

This chapter being the formal description of level-based restriction placed upon the derivation trees as it was informally presented in the introduction of this work (see Sec. 1.1.1).

2.1.1 Definitions

In Sec. 1.1.1, *tree controlled grammars* are described informally. Here, we present the corresponding strictly formal definitions.

Definition 2.1 (Tree controlled grammar)**.** A *tree controlled grammar* is a pair (G, R) where $G = (V, T, P, S)$ is a controlled grammar and $R \subseteq V^*$ is a control language.

Definition 2.2 (Tree controlled language)**.** Let (G, R) be a tree controlled grammar. The *language generated by* (G, R) is denoted by $L(G, R)$ and defined by

$$L(G,R) = \{x : x \in L(G) \text{ and there exists a derivation tree of } x \text{ in } G \text{ such that each word obtained by concatenation of all symbols at any level of } t \text{ (except the last one) from left to right is in } R\}.$$

Definition 2.3 (Class of tree controlled languages)**.** For $\mathbf{X} \in \{\mathbf{CF}, \mathbf{CF}_\varepsilon, \mathbf{REG}\}$ and $\mathbf{Y} \in \{\mathbf{RE}, \mathbf{CS}, \mathbf{CF}, \mathbf{REG}, \mathbf{FIN}\} \cup \{\mathbf{MON}, \mathbf{NIL}, \mathbf{COMB}, \mathbf{ORD}, \mathbf{DEF}, \mathbf{COM}, \mathbf{NC}, \mathbf{CIRC}, \mathbf{PS}, \mathbf{SUF}\} \cup \{\mathbf{MON}_n, \mathbf{REG}_n : n \geq 1\}$, the *class of tree controlled languages* is denoted as $\mathbf{TC}(\mathbf{X}, \mathbf{Y})$ and defined as

$$\mathbf{TC}(\mathbf{X},\mathbf{Y}) = \{L(G,R) : (G,R) \text{ is a tree controlled grammar in which } G \in \mathbb{G}_\mathbf{X} \text{ and } R \in \mathbf{Y}\}.$$

Definition 2.4 (Nonterminal complexity of tree controlled grammar)**.** Let (G, R) be a tree controlled grammar. Clearly, $R = L(G')$ for some $G' = (V', T', P', S')$. A *nonterminal complexity of tree controlled grammar* is denoted as $\mathrm{Var}(G, R)$ and defined as $\mathrm{Var}(G, R) = \mathrm{card}(V - T) + \mathrm{card}(V' - T')$.

Definition 2.5 (Nonterminal complexity of tree controlled language)**.** Let $\min(\Sigma)$ denote the smallest number in a given set of natural numbers, Σ. Let $L \in \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{REG})$. An *$\varepsilon$-free nonterminal complexity of tree controlled language* is denoted as $\mathrm{Var}_\varepsilon(L)$ and defined as

$$\mathrm{Var}_\varepsilon(L) = \min(\{\mathrm{Var}(G, R) : (G, R) \text{ is a tree controlled grammar with an } \varepsilon\text{-free context-free grammar, } G, \text{ regular language, } R, \text{ and } L(G, R) = L\}).$$

Let $L \in \mathbf{TC}(\mathbf{CF}, \mathbf{REG})$. A *nonterminal complexity of tree controlled language* is denoted as $\mathrm{Var}(L)$ and defined as

$$\mathrm{Var}(L) = \min(\{\mathrm{Var}(G, R) : (G, R) \text{ is a tree controlled grammar with a context-free grammar, } G, \text{ regular language, } R, \text{ and } L(G, R) = L\}).$$

2.1.2 Results

The results presented in this section represent the state of the art of level-based restriction placed upon the derivation trees investigation area. For the informal description of the corresponding state of the art, see Sec. 1.1.1.

Theorem 2.1 (Th. 3.1 in [9])**.** *If (G, R) is a tree controlled grammar where G is unambiguous context-free grammar and R is a regular language, then there exists an algorithm which for any word, w, with $|w| = n$ determines in $O(n^2)$ steps if $w \in L(G, R)$.*

Theorem 2.2 (Th. 3.2 in [9])**.** *For every tree controlled grammar, (G, R), with ε-free context-free grammar, $G = (V, T, P, S)$, and regular language, R, the language $L(G, R)$ is recursive.*

Theorem 2.3 (Th. 3.3 in [9])**.** *A language, L, is regular, linear, context-free if and only if there is a context-free grammar, $G = (V, T, P, S)$, such that $L = L(G, T^*(V - T))$, $L = L(G, T^*(V - T)T^*)$, $L = L(G, V^*)$, respectively.*

Theorem 2.4 (Th. 3.5 in [9])**.** *Let Σ be an alphabet. Then, there exists a context-free grammar, $G = (V, \Sigma, P, S)$, such that for each recursively enumerable language, $L \subseteq \Sigma^*$, there exists a regular language, $R \subseteq V^*$, such that $L = L(G, R)$.*

Corollary 2.5 (Col. 3.1 in [9])**.** *Every recursively enumerable language can be generated by a tree controlled grammar, (G, R), such that $G = (V, \Sigma, P, S)$ is a context-free grammar, R is a regular language, and $P \subseteq (V - \Sigma) \times ((V - \Sigma)^* \cup \Sigma)$.*

Theorem 2.6 (Th. 3.6 in [9])**.** *Every recursively enumerable language can be generated by a tree controlled grammar, (G, R), with a context-free grammar, $G = (V, T, P, S)$ and a regular language, $R \subseteq (V - T)^*$.*

Theorem 2.7 (Th. 1 in [39])**.** *If G is a regular grammar, then for all $R \in \mathbf{X}$, where $\mathbf{X} \in \{\mathbf{RE}, \mathbf{CS}, \mathbf{CF}, \mathbf{REG}, \mathbf{FIN}\}$, $L(G, R) \in \mathbf{REG}$.*

Corollary 2.8 (Col. in Sec. 2 in [39])**.**

For all $\mathbf{X} \in \{\mathbf{RE}, \mathbf{CS}, \mathbf{CF}, \mathbf{REG}, \mathbf{FIN}\}$, $\mathbf{TC}(\mathbf{REG}, \mathbf{X}) = \mathbf{REG}$.

Theorem 2.9 (Th. 2 in [39])**.** $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{CS}) \subseteq \mathbf{CS}$.

Theorem 2.10 (Th. 3 in [39])**.** $\mathbf{X} \subseteq \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{X})$, *for* $\mathbf{X} \in \{\mathbf{RE}, \mathbf{CS}, \mathbf{CF}, \mathbf{REG}\}$.

Corollary 2.11 (Col. in Sec. 2 in [39]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{RE}) = \mathbf{RE}$ *and* $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{CS}) = \mathbf{CS}$.

Theorem 2.12 (Th. 4 in [39]). $\mathbf{CS} \subseteq \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{REG})$.

Corollary 2.13 (Col. in Sec. 2 in [39]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{X}) = \mathbf{CS}$*, for* $\mathbf{X} \in \{\mathbf{CS}, \mathbf{CF}, \mathbf{REG}\}$.

Theorem 2.14 (Th. 5 in [39]). $\mathbf{TC}(\mathbf{X}, \mathbf{FIN}) = \mathbf{FIN}$*, for* $\mathbf{X} \in \{\mathbf{CF}, \mathbf{CF}_\varepsilon\}$.

Corollary 2.15 (Col. in Sec. 2 in [39]).

Any language over one-letter alphabet in $\mathbf{TC}(\mathbf{CF}, \mathbf{FIN})$ *is regular.*

Lemma 2.16 (Lem. 4 in [13]). *If* $\mathbf{X} \subseteq \mathbf{Y} \subseteq \mathbf{REG}$*, then* $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{X}) \subseteq \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{Y})$*, for* $\mathbf{X}, \mathbf{Y} \in \{\mathbf{FIN}, \mathbf{MON}, \mathbf{NIL}, \mathbf{COMB}, \mathbf{ORD}, \mathbf{DEF}, \mathbf{COM}, \mathbf{NC}, \mathbf{CIRC}, \mathbf{PS}, \mathbf{SUF}\}$.

Theorem 2.17 (Th. 5 in [13]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{SUF}) = \mathbf{CS}$.

Theorem 2.18 (Th. 6 in [13]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{CIRC}) = \mathbf{CS}$.

Theorem 2.19 (Th. 7 in [13]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{ORD}) = \mathbf{CS}$.

Corollary 2.20 (Col. 8 in [13]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{NC}) = \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{PS}) = \mathbf{CS}$.

Theorem 2.21 (Th. 9 in [13]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{COM}) = \mathbf{MAT}_\varepsilon$ *where* $\mathbf{MAT}_\varepsilon$ *denotes the class of languages generated by matrix grammars in which all matrices contains no erasing productions.*

Theorem 2.22 (Th. 10 in [13]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{FIN}) = \mathbf{MAT}_{fin}$ *where* $\mathbf{MAT}_{fin}$ *denotes the class of languages generated by matrix grammars of finite index in which all matrices contains no erasing productions.*

Theorem 2.23 (Th. 12 in [13]).

$\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{FIN}) \subset \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{NIL})$ *and*
$\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{MON}) \subset \mathbf{TC}(\mathbf{NIL})$.

Theorem 2.24 (Th. 13 in [13]).

$$\mathbf{RE} = \mathbf{TC}(\mathbf{CF}, \mathbf{REG}) = \mathbf{TC}(\mathbf{CF}, \mathbf{SUF}) = \mathbf{TC}(\mathbf{CF}, \mathbf{ORD}) = \mathbf{TC}(\mathbf{CF}, \mathbf{NC}) = \\ = \mathbf{TC}(\mathbf{CF}, \mathbf{PS}) = \mathbf{TC}(\mathbf{CF}, \mathbf{COM}) = \mathbf{TC}(\mathbf{CF}, \mathbf{CIRC}),$$
$$\mathbf{MAT}_{fin} = \mathbf{TC}(\mathbf{CF}, \mathbf{FIN}) \subset \mathbf{TC}(\mathbf{CF}, \mathbf{NIL}) \subseteq \mathbf{RE},$$
$$\mathbf{TC}(\mathbf{CF}, \mathbf{MON}) \subset \mathbf{TC}(\mathbf{CF}, \mathbf{NIL}), \text{ and } \mathbf{TC}(\mathbf{CF}, \mathbf{MON}) \subset \mathbf{TC}(\mathbf{CF}, \mathbf{DEF}).$$

Theorem 2.25 (Prop. 4 in [12]).

$$\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{MON}_1) \subseteq \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{MON}_2) \subseteq \cdots \subseteq \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{MON}_j) \subseteq \ldots.$$

Lemma 2.26 (Lem. 10 in [12]).

$$\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{REG}_1) \subseteq \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{REG}_2) \ldots \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{REG}_n) \subseteq \ldots.$$

Theorem 2.27 (Th. 12 in [12]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{REG}_1) \subset \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{REG}_2)$.

Theorem 2.28 (Th. 13 in [12]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{COMB}) \subseteq \mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{REG}_2)$.

Theorem 2.29 (Th. 18 in [12]). *Every language that is generated by a context-sensitive grammar with exactly r non-context-free productions $p_1, p_2, \ldots, p_r$ and n_i symbols on the left-hand side of the production p_i, for $1 \leq i \leq r$, is also generated by a tree controlled grammar where the control set is accepted by a deterministic finite automaton with at most $2 + \Sigma_{i=1}^{r}(n_i - 1)$ states.*

Lemma 2.30 (Lem. 5 in [47]). *For every linearly bounded queue automaton (see Def. 3 in [47] for the definition), A, there is a tree controlled grammar, (G, R), such that $L(G, R) = L(A)$.*

Theorem 2.31 (Th. 6 in [47]). $\mathbf{TC}(\mathbf{CF}_\varepsilon, \mathbf{REG}_5) = \mathbf{CS}$.

Theorem 2.32 (Th. 7 in [47]). $\mathbf{TC}(\mathbf{CF}, \mathbf{REG}_5) = \mathbf{RE}$.

Theorem 2.33 (Th. 4 in [49]). *Every recursively enumerable language can be generated by a tree controlled grammar, (G, R), with context-free grammar, G, and a regular language, R, such that* $\mathrm{Var}(G, R) \leq 9$.

Theorem 2.34 (Th. 1 in [48]). *Every recursively enumerable language can be generated by a tree controlled grammar, (G, R), with context-free grammar, G, and regular language, R, such that* $\mathrm{Var}(G, R) \leq 7$.

Theorem 2.35 (Th. 2 in [48]). *For any regular language, L, there is a tree controlled grammar, (G, R), with context-free grammar, G, and a regular language, R, such that $L(G, R) = L$ and* $\mathrm{Var}(G, R) = 3$.

Theorem 2.36 (Th. 4 in [48]). *For any regular simple matrix grammar, G, (see [38] for the definition) there is a tree controlled grammar, (G, R), with context-free grammar, G, and a regular language, R, such that $L(G, R) = L(G)$ and* $\mathrm{Var}(G, R) = 3$.

Theorem 2.37 (Th. 5 in [48]). *For any linear language, L, there is a tree controlled grammar, (G, R), such that $L(G, R) = L$ and* $\mathrm{Var}(G, R) = 3$.

Lemma 2.38 (Lem. 2 in [49]). *For $n \geq 1$, let $L_n = \bigcup_{i=1}^{n}\{a_i^j : j \geq 1\}$. Then,* $\mathrm{Var}_\varepsilon(L_n) = n + 1$.

2.2 Path Controlled Grammars

In this chapter, based on the informal survey introduced in Sec. 1.1.2, we present the definitions and the results related to path controlled grammar.

2.2.1 Definitions

The following definitions are needed for presenting the state of the art in path-based restriction placed upon the derivation trees investigation area.

Definition 2.6 (Language of the paths). Let $G = (V, T, P, S)$ be a context-free grammar and $t \in {}_G\triangle(w)$ for some $w \in L(G)$. Then,

$path(t) = \{\mathrm{word}(p) : p \text{ is a path of } t\}$,
$path({}_G\triangle(w)) = \bigcup\{path(t) : t \in {}_G\triangle(w)\}$, and
$path(G) = \bigcup\{path({}_G\triangle(w)) : \text{for all } w \in L(G)\}$.

Definition 2.7 (Path controlled grammar)**.** A path controlled grammar is a pair (G, G') where $G = (V, T, P, S)$ is a controlled grammar and $G' = (V', V, S', P')$ is a controlling grammar.

Definition 2.8 (Path controlled language)**.** Let (G, G') be a path controlled grammar. The language generated by (G, G') is denoted by $L(G, G')$ and defined by

$$L(G, G') = \{w \in L(G) : \ path(w) \cap L(G') \neq \emptyset\}.$$

Definition 2.9 (Class of path controlled languages)**.** For $\mathbf{X}, \mathbf{Y} \in \{\mathbf{CF}, \mathbf{LIN}, \mathbf{REG}\}$, the *class of path controlled languages* is denoted as $\mathbf{PC}(\mathbf{X}, \mathbf{Y})$ and defined as

$$\mathbf{PC}(\mathbf{X}, \mathbf{Y}) = \{L(G, G') : (G, G') \text{ is a path controlled grammar in which } G \in \mathbb{G}_{\mathbf{X}} \text{ and } G' \in \mathbb{G}_{\mathbf{Y}}\}.$$

2.2.2 Results

This section being the summary of the fundamentals in the state of the art of path-based restriction placed upon the derivation trees investigation area. For the informal description of the corresponding state of the art, see Sec. 1.1.2.

Theorem 2.39 (Prop. 1 in [32])**.** *If G is a context-free grammar, then $path(G) \in \mathbf{REG}$.*

Theorem 2.40 (Prop. 2 in [32])**.** $\mathbf{X} = \mathbf{PC}(\mathbf{X}, \mathbf{REG})$, *for all* $\mathbf{X} \in \{\mathbf{REG}, \mathbf{LIN}, \mathbf{CF}\}$.

Theorem 2.41 (Prop. 3 in [32])**.** $\mathbf{PC}(\mathbf{REG}, \mathbf{X}) \subseteq \mathbf{X}$, *for all* $\mathbf{X} \in \{\mathbf{LIN}, \mathbf{CF}\}$.

Theorem 2.42 (Prop. 4 in [32])**.** *If L is a language in $\mathbf{X} \in \{\mathbf{LIN}, \mathbf{CF}\}$ without words of length one, then $L \in \mathbf{PC}(\mathbf{REG}, \mathbf{X})$.*

Theorem 2.43 (Prop. 6 in [32]; see the discussion in Chap. 3)**.** $\mathbf{PC}(\mathbf{CF}, \mathbf{CF}) \subseteq \mathbf{MAT}$.

Theorem 2.44 (Prop. 7 in [32])**.** *If $L \subseteq V^*$, $L \in \mathbf{PC}(\mathbf{CF}, \mathbf{CF})$, then there are two constants, p and q, such that each word, $z \in L$, with $|z| > p$ can be written in the form $z = u_1v_1u_2v_2u_3v_3u_4v_4u_5$, such that $0 < |v_1v_2v_3v_4| \leq q$ and $u_1v_1^iu_2v_2^iu_3v_3^iu_4v_4^iu_5 \in L$ for all $i \geq 1$.*

Theorem 2.45 (Prop. 8 in [32])**.** *If $L \subseteq V^*$, $L \in \mathbf{PC}(\mathbf{LIN}, \mathbf{LIN})$, then there are two constants, p and q, such that each word, $z \in L$, with $|z| > p$ can be written in the form $z = u_1v_1u_2v_2u_3v_3u_4v_4u_5$, such that $0 < |v_1v_2v_3v_4| \leq q$, $|u_1u_2u_3u_4| \leq q$, and $u_1v_1^iu_2v_2^iu_3v_3^iu_4v_4^iu_5 \in L$ for all $i \geq 1$.*

Corollary 2.46 (Conseq. in [32]; see the discussion in Chap. 3)**.** $\mathbf{PC}(\mathbf{CF}, \mathbf{CF}) \subset \mathbf{MAT}$.

Corollary 2.47 (Conseq. in [32])**.** $\mathbf{PC}(\mathbf{LIN}, \mathbf{LIN})$ *is not closed under concatenation.*

Theorem 2.48 (Prop. 9 in [32])**.** *For each language, $L \subseteq V^*$, $L \in \mathbf{PC}(\mathbf{LIN}, \mathbf{LIN})$, there are three linear languages $L_1 \subseteq V^*\{c\}V^*$, $L_2 = V^*\{c\}V^*$, and $L_3 = V^*$, where $c \notin V$, such that:*

- *$L \subseteq \{u_1u_2u_3u_4u_5 | \ u_1cu_5 \in L_1, u_2cu_4 \in L_2, u_3 \in L_3\}$.*
- *For each word, $u_1cu_5 \in L_1$ (for each word, $u_2cu_4 \in L_2$, for each word, $u_3 \in L_3$) there are a word, $u_2cu_4 \in L_2$, and a word, $u_3 \in L_3$ (a word, $u_1cu_5 \in L_1$, and a word, $u_3 \in L_3$, respectively, a word, $u_1cu_5 \in L_1$, and a word, $u_2cu_4 \in L_2$) such that $u_1u_2u_3u_4u_5 \in L$.*

Theorem 2.49 (Prop. 10 in [32]).

$\mathbf{CF} - \mathbf{PC}(\mathbf{LIN}, \mathbf{LIN}) \neq \emptyset$.
The inclusion $\mathbf{PC}(\mathbf{LIN}, \mathbf{LIN}) \subset \mathbf{PC}(\mathbf{CF}, \mathbf{CF})$ *is proper.*

Theorem 2.50 (Prop. 11 in [32]). *If* (G, G') *is a path controlled grammar with linear grammars* G *and* G' *such that* G *has a bounded ambiguity, the parsing of words in* $L(G, G')$ *can be done in a polynomial time.*

Theorem 2.51 (Prop. 1 in [33]). $\mathbf{PC}(\mathbf{CF}, \mathbf{CF})$ *is closed under the following operations: union, intersection with regular languages, left and right concatenation with context-free languages, substitution with* ε*-free context-free languages, non-erasing homomorphism. It is not closed under intersection.*

Theorem 2.52 (Prop. 2 in [33]). *The emptiness problem is decidable for path controlled grammars.*

Theorem 2.53 (Prop. 3 in [33]). *The finiteness problem is decidable for path controlled grammars.*

Theorem 2.54 (Prop. 4 in [33]). *One cannot algorithmically decide whether or not the language generated by a given path controlled grammar is context-free.*

Theorem 2.55 (Prop. 5 in [33]). *The language generated by a given path controlled grammar can be recognized in* $O(n^{10})$ *time.*

Chapter 3

New Results and Future Research Ideas

In this chapter, we briefly present the new definitions as well as the new results. However, due to maintain a reasonable number of pages of this short doctoral thesis, the results are presented detailed motivation and without proofs.

Since a restriction placed upon a level, a path, and a cut is, in essence, a restriction placed upon a derivation tree, we use a slightly modified but equivalent formulation of the definitions stated in [32], [33], and [34]. Consequently, aforementioned modifications allow us to investigate all derivation-tree-based restrictions using the same terminology—i.e., restriction on the levels (see [9], [13], [39], [48], and [49]), the paths (see [7], [20], [21], [22], [23], [24], [25], [26], [27], [32], [33], and [34]), or the cuts (see [27]). More precisely, all restrictions placed upon the derivation trees are covered by the general notion of *tree controlled grammar* that generates its language under several kinds of the restrictions.

Note that hereafter the notion *tree controlled grammar* is used in different meaning than in Chap. 2, see the following definitions of a *tree controlled grammar* and the definitions of the languages as well as the classes that tree controlled grammars generate under various kinds of restrictions that are introduced in the following three chapters.

First we reformulate the fundamental definitions so that all derivation-tree-based restrictions can be studied using the same notation. Then, this chapter summarizes the preliminaries common for all new results presented in the subsequent sections of this work.

Definition 3.1 (Tree controlled grammar)**.** A *tree controlled* grammar is a pair, (G, R), where $G = (V, T, P, S)$ is a controlled grammar, and R is a control language over V.

Definition 3.2 (Set of derivation trees)**.** Let (G, R) be a tree controlled grammar where $G = (V, T, P, S)$, then ${}_{(G,R)}\triangle(x)$, $x \in V^*$, denotes the set of the derivation trees with frontier x in G.

In the research presented through this part, we do not directly deal with level-based restriction placed upon the derivation trees. However, for the sake of completeness, note the following definitions related to level-based restriction placed upon the derivation trees.

Definition 3.3 (Language of tree controlled grammar under levels control)**.** Let (G, R) be a tree controlled grammar. The *language that* (G, R) *generates under the levels control by* R is denoted by ${}_{levels}L(G, R)$ and defined by the following equivalence:

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

Definition 3.4 (Class of tree controlled languages under levels control)**.** For some language classes $\mathbf{X}$ and $\mathbf{Y}$, the class of *tree controlled languages under the levels control* is defined as

$$\textbf{levels-TC}(\mathbf{X},\mathbf{Y}) = \{_{levels}L(G,R) : (G,R) \text{ is a tree controlled grammar in which } G \in \mathbb{G}_{\mathbf{X}} \text{ and } R \in \mathbf{Y}\}$$

Next, we summarize the most interesting results achieved in this work and point out some important open questions. Based on the *State of the Art* in the area of restrictions placed upon the derivation trees summarized in Sec. 1.1 and Chap. 2, this work deals in principle with three kinds of derivation-tree based restrictions, cut-based, path-based, and several-path-based.

3.1 Cut Based Restriction

In this section, we introduce new derivation-tree-based restrictions of tree-controlled grammars which are based on the restriction placed upon the cuts. Then, we introduce several properties of grammars with restriction placed upon the cuts. Specifically, we investigate the generative power.

3.1.1 Definitions

Definition 3.5 (Language of tree controlled grammar under cuts control)**.** Let (G,R) be a tree controlled grammar. The *language that* (G,R) *generates under the cuts control by* R is denoted by $_{cut}L(G,R)$ and defined by the following equivalence:

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

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

Definition 3.6 (Class of tree controlled languages under cuts control)**.** The class of *tree controlled languages under the cuts control* is defined as

$$\textbf{cut-TC}(\mathbf{CF},\mathbf{REG}) = \{_{cut}L(G,R) : (G,R) \text{ is a tree controlled grammar in which } G \text{ is a context-free grammar and } R \in \mathbf{REG}\}$$

and the *class of tree controlled languages with* ε*-free controlled grammar under cuts control* is defined as

$$\textbf{cut-TC}_{\varepsilon}(\mathbf{CF}_{\varepsilon},\mathbf{REG}) = \{_{cut}L(G,R) : (G,R) \text{ is a tree controlled grammar in which } G \text{ is an } \varepsilon\text{-free context-free grammar and } R \in \mathbf{REG}\}.$$

Definition 3.7 (Ordering relation on the cuts)**.** Let $\preceq$ be a binary relation on a sequence, $_xM$, of the cuts such that for each two cuts, $c_1, c_2 \in {}_xM$, $c_1 \preceq c_2$ if and only if for each node, n_2, of c_2 either there is a node, n_1, of c_1 such that n_2 is a direct descendent of n_1, or $n_1 = n_2$. In other words, $n_1 \neq n_2$ implies n_2 is a direct descendent of n_1.

Definition 3.8 (Language of tree controlled grammar under ordered-cuts control)**.** Let (G, R) be a tree controlled grammar. The *language that* (G, R) *generates under the ordered-cuts control by* R is denoted by ${}_{ord\text{-}cut}L(G, R)$ and defined by the following equivalence:

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

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

Definition 3.9 (Class of tree controlled languages under ordered-cuts control)**.** The class of *tree controlled languages under the ordered cuts control* is defined as

$$\textbf{ord-cut-TC}(\textbf{CF}, \textbf{REG}) = \{{}_{ord\text{-}cut}L(G, R) : (G, R) \text{ is a tree controlled grammar in which } G \text{ is a context-free grammar and } R \in \textbf{REG}\}$$

and the *class of tree controlled languages with* ε*-free controlled grammar under ordered cuts control* is defined as

$$\textbf{ord-cut-TC}_\varepsilon(\textbf{CF}_\varepsilon, \textbf{REG}) = \{{}_{ord\text{-}cut}L(G, R) : (G, R) \text{ is a tree controlled grammar in which } G \text{ is an } \varepsilon\text{-free context-free grammar and } R \in \textbf{REG}\}.$$

3.1.2 Results

Concerning cut-based restriction placed upon the derivation trees, we have introduced two fundamental types of such kind of a restriction and thus, we have opened a new investigation area in derivation-tree-restricted models. Next, we have proved that both restrictions increase the generative power of context-free grammars so they characterize **RE**:

$$\textbf{ord-cut-TC}(\textbf{CF}, \textbf{REG}) = \textbf{cut-TC}(\textbf{CF}, \textbf{REG}) = \textbf{RE}.$$

An important open problem consists of the investigation of cut controlled grammars where ε-productions are forbidden. Consequently, the grammars restricted in this way should be placed into the relation with some other well-known language families, such as **CS**, and the deciding the question whether or not:

$$\textbf{ord-cut-TC}_\varepsilon(\textbf{CF}_\varepsilon, \textbf{REG}) = \textbf{cut-TC}_\varepsilon(\textbf{CF}_\varepsilon, \textbf{REG}) = \textbf{CS}.$$

Next open problem is the descriptional complexity of $\textbf{ord-cut-TC}(\textbf{CF}, \textbf{REG})$ and $\textbf{cut-TC}(\textbf{CF}, \textbf{REG})$. The results stated in above are based on the transformation of an unrestricted grammar in Pentonnen normal form. However, using Geffert normal form, the number of nonterminals in the resulting cut controlled grammar would be reduced.

Another future research idea is represented by the controlling the cuts of the derivation trees in which several types of subregular control languages are considered. In this way, the question whether or not a kind of a subregural language is enough to increase the generative power of controlled grammar properly. Consequently, the relation between the generative power of level-based and cut-based models restricted in this way would be founded out.

3.2 Path Based Restriction

In this section, we introduce a path-based restriction on tree-controlled grammars that is equivalent to the model introduced in Sec. 2.2. Next, we formally define the pseudoknot structure represented as a language. We introduce new results related to the *normal forms* and the presence of erasing productions in a controlled grammar. Then, this section presents a relationship between biology and the formal language theory in the form of word representation of pseudoknots generated by path controlled grammars. Last section of this part being a counterargument against the proof of the generative power of path controlled grammars that has been considered as correct so far.

3.2.1 Definitions

Definition 3.10 (Language of tree controlled grammar under path control)**.** Let (G, R) be a tree controlled grammar. The *language that* (G, R) *generates under the path control by* R is denoted by ${}_{path}L(G, R)$ and defined by the following equivalence:

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

Definition 3.11 (Class of tree controlled languages under path control)**.** For $\mathbf{X}, \mathbf{Y} \in \{\mathbf{LIN}, \mathbf{CF}\}$, the *class of tree controlled languages under the path control* is defined as

$$\textbf{path-TC}(\mathbf{X}, \mathbf{Y}) = \{{}_{path}L(G, R) : (G, R) \text{ is a tree controlled grammar in which } G \in \mathbb{G}_{\mathbf{X}} \text{ and } R \in \mathbf{Y}\}$$

and the *class of tree controlled languages with* ε*-free controlled grammar under path control* is defined as

$$\textbf{path-TC}_{\varepsilon}(\mathbf{CF}_{\varepsilon}, \mathbf{Y}) = \{{}_{path}L(G, R) : (G, R) \text{ is a tree controlled grammar in which } G \text{ is an } \varepsilon\text{-free context-free grammar and } R \in \mathbf{Y}\}.$$

Definition 3.12 (1^{st} normal form of a tree controlled grammar that generates the language under path control)**.** Let (G, R) be a tree controlled grammar that generates the language under path control by R, where $G = (V, T, P, S)$. (G, R) is in 1^{st} *normal form* if every production, $r : A \to x \in P$, is of the form $A \in V - T$ and $x \in T \cup (V - T) \cup (V - T)^2$.

Definition 3.13 (2^{nd} normal form of a tree controlled grammar that generates the language under path control)**.** Let (G, R) be a tree controlled grammar that generates the language under path control by R, where $G = (V, T, P, S)$. (G, R) is in 2^{nd} *normal form* if every production, $r : A \to x \in P$, is of the form $A \in V - T$ and $x \in T \cup ((V \cup \{E\}) - T)^2$ where $E \cap V = \emptyset$ and $E \to \varepsilon \in P$. The alphabet of G should now include E, with $E \notin V$.

Definition 3.14 (Pseudoknot)**.** Let Σ be an alphabet. The following languages over Σ are pseudoknots:

1. $\{xyx^Ry^R : \ x, y \in \Sigma^*\}$,
   $\{u_1xu_2yu_3x^Ru_4y^Ru_5 : \ x, y, u_i \in \Sigma^*, 1 \leq i \leq 5\}$,
2. $\{xyx^Rzz^Ry^R : \ x, y, z \in \Sigma^*\}$,
   $\{u_1xu_2yu_3x^Ru_4zu_5z^Ru_6y^Ru_7 : \ x, y, z, u_i \in \Sigma^*, 1 \leq i \leq 7\}$,
3. $\{xyx^Rzy^Rz^R : \ x, y, z \in \Sigma^*\}$,
   $\{u_1xu_2yu_3x^Ru_4zu_5y^Ru_6z^Ru_7 : \ x, y, z, u_i \in \Sigma^*, 1 \leq i \leq 7\}$,

4. $\{xyzx^Ry^Rz^R : \ x, y, z \in \Sigma^*\}$,
$\{u_1xu_2yu_3zu_4x^Ru_5y^Ru_6z^Ru_7 : \ x, y, z, u_i \in \Sigma^*, 1 \leq i \leq 7\}$.

Note that presented pseudoknots form obviously non-context-free languages.

3.2.2 Results

As a continuation of the investigation of path-based restrictions introduced in [32] and studied in [33] and [34], we have considered the impact of ε-productions in path controlled grammars to the generative power and we have stated that ε-productions can be removed from a path controlled grammar without affecting its language:

$$\textbf{path-TC}(\textbf{CF}, \textbf{CF}) = \textbf{path-TC}_{\varepsilon}(\textbf{CF}_{\varepsilon}, \textbf{CF}).$$

Next, we have established two Chomsky-like normal forms for path controlled grammars (see Def. 3.12 and Def. 3.13) and we have formulated algorithms that transform a path controlled grammar into its normal form:

- Let $L \in \textbf{path-TC}(\textbf{CF}, \textbf{CF})$. Then, there exists a tree controlled grammar, (G, R), in 1^{st} normal form such that $L = {}_{path}L(G, R)$.
- Let $L \in \textbf{path-TC}(\textbf{CF}, \textbf{CF})$. Then, there exists a tree controlled grammar, (G, R), in 2^{nd} normal form such that $L = {}_{path}L(G, R)$.

A future investigation idea consists of the modifying a general parsing methods that are based on Chomsky normal form such that it will be able to parse path controlled grammars in a polynomial time.

Another practical motivated idea is represented the relation between path controlled grammars and the theory of pseudoknots. We have demonstrated several typical pseudoknots used in biology represented by the words (see Def. 3.14) of non-context-free languages. We have demonstrated some pseudoknots belong to $\textbf{path-TC}(\textbf{LIN}, \textbf{LIN})$:

$\{xyx^Ry^R : \ x, y \in \Sigma^* \text{ for some } \Sigma\} \in \textbf{path-TC}(\textbf{LIN}, \textbf{LIN})$,
$\{xyx^Rzz^Ry^R : \ x, y, z \in \Sigma^* \text{ for some } \Sigma\} \in \textbf{path-TC}(\textbf{LIN}, \textbf{LIN})$,
$\{xyx^Rzy^Rz^R : \ x, y, z \in \Sigma^* \text{ for some } \Sigma\} \in \textbf{path-TC}(\textbf{LIN}, \textbf{LIN})$.

Apparently, there is a huge variety of another pseudoknot structures in biology. For example, $\{xyzx^Ry^Rz^R : \ x, y, z \in \Sigma^*\}$ and it is an open question whether or not those pseudoknots can be generated by tree controlled grammars with linear components that generate the language under path control.

The last presented result deals with a reflection on the generative power of path controlled grammars that has been considered as well-known (see [32]) for more than last ten years. However, we have presented an argument against the correctness of the proof given in [32] that states $\textbf{path-TC}(\textbf{CF}, \textbf{CF}) \subseteq \textbf{MAT}$. We have concluded the counterargument by stating that the aforementioned inclusion still may hold; however, it cannot be proved in the way given in [32]. More precisely, we have found the language tha can be generated by a grammar with controlled path. However, this language cannot be generated by the grammar obtained by the construction introduced in [32]. Apparently, the generative power of path controlled grammar still represents an open problem.

3.3 Several Paths Based Restriction

This section being a generalization of path-restricted rewriting model to a restriction placed upon not just one but several paths. Then, it presents several properties of the model restricted in this way. Specifically, the generative power of a kind of all-paths-restricted rewriting model, closure and pumping properties in relation to the number of controlled paths, and the approximation of the generative power for n-path restricted model. The last section of this part presents an application related result concerning the parsing of path restricted languages.

3.3.1 Definitions

Definition 3.15 (Language of tree controlled grammar under all paths control)**.** Let (G, R) be a tree controlled grammar. The *language that* (G, R) *generates under the all-paths control by* R is denoted by ${}_{\textit{all-paths}}L(G, R)$ and defined by the following equivalence:

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

Definition 3.16 (Class of tree controlled languages under all paths control)**.** The *class of tree controlled languages under all paths control* is defined as

$$\textbf{all-path-TC}(\textbf{CF}, \textbf{REG}) = \{{}_{\textit{all-paths}}L(G, R) : (G, R) \text{ is a tree controlled grammar in which } G \text{ is a context-free grammar and } R \in \textbf{REG}\}.$$

Definition 3.17. Let (G, R) be a tree controlled grammar. The *language of tree controlled grammar under not common n-path control* by R, $n \geq 1$, is denoted by ${}_{\textit{nc-n-path}}L(G, R)$ and defined by the following equivalence:

For all $x \in T^*$, $x \in {}_{\textit{nc-n-path}}L(G, R)$ if there exists a derivation tree, $t \in {}_G\triangle(x)$, such that there is a set, Q_t, of n paths of t such that for each path, $p \in Q_t$, $\text{word}(p) \in R$.

Definition 3.18. For $\mathbf{X}, \mathbf{Y} \in \{\textbf{REG}, \textbf{LIN}, \textbf{CF}\}$, the *class of tree controlled languages under not-common n-path control* is defined as

$$\textbf{nc-n-path-TC}(\mathbf{X}, \mathbf{Y}) = \{{}_{\textit{nc-n-path}}L(G, R) : (G, R) \text{ is a tree controlled grammar with } G \in \mathbb{G}_{\mathbf{X}} \text{ and } R \in \mathbf{Y}\}.$$

Definition 3.19 (Common part of two paths)**.** Let p_1, p_2 be any different two paths of a derivation tree, t. Then, p_1 and p_2 contain at least one common node (the root of t, $\text{root}(t)$), and p_1 ends in a different leaf of t than p_2. Let $\text{cmn}(p_1, p_2)$ denote the maximal number of consecutive common nodes of p_1 and p_2.

Definition 3.20 (Common node of division of set of paths)**.** Let Q_t be a nonempty set of some paths of a derivation tree, t. The *paths of Q_t are divided in a common node of t* if and only if for some $k \geq 1$, $\text{cmn}(p_1, p_2) = k$ for every pair $(p_1, p_2) \in Q_t^2$. Let all paths of Q_t be divided in a common node of t. If $Q_t = \{p\}$, then $m_{Q_t} = |\,\text{word}(p)|$, otherwise $m_{Q_t} \geq 1$ denotes the maximal number of consecutive common nodes of all paths in Q_t.

Definition 3.21 (Language of tree controlled grammar under n-path control)**.** Let (G, R) be a tree controlled grammar. The *language of tree controlled grammar under n-path control* by R, $n \geq 1$, is denoted by ${}_{\textit{n-path}}L(G, R)$ and defined by the following equivalence:

For all $x \in T^*$, $x \in {}_{n\text{-}path}L(G, R)$ if there exists a derivation tree, $t \in {}_G\triangle(x)$, such that there is a set, Q_t, of n paths of t that are divided in a common node of t and for each path, $p \in Q_t$, word$(p) \in R$.

Definition 3.22 (Class of tree controlled languages under n-path control)**.** For $\mathbf{X}, \mathbf{Y} \in \{\mathbf{REG}, \mathbf{LIN}, \mathbf{CF}\}$, the *class of tree controlled languages under n-path control* is defined as

$$\textbf{n-path-TC}(\mathbf{X}, \mathbf{Y}) = \{{}_{n\text{-}path}L(G, R) : (G, R) \text{ is a tree controlled grammar with } G \in \mathbb{G}_{\mathbf{X}} \text{ and } R \in \mathbf{Y}\}.$$

Conventions

Hereafter, tree controlled grammars that generate the language under the n-path control are referred to as *n-path tree controlled grammars.*

Note that if we consider 0 controlled paths (i.e., $n = 0$ and consequently card$(Q_t) = 0$) in the definition of ${}_{n\text{-}path}L(G, R)$, then, clearly, the generative power of such a model equals **CF**.

Definition 3.23 (Special types of n-path controlled grammars)**.** Let (G, R) be a tree controlled grammar. Consider ${}_{n\text{-}path}L(G, R)$, for $n \geq 1$. If for each word, $z \in {}_{n\text{-}path}L(G, R)$, there exist a derivation tree, $t \in {}_G\triangle(z)$, a set of its paths, Q_t, $m_{Q_t} \geq 1$, and a partition, word$(p) = uvwxy$, for each path, $p \in Q_t$, satisfying the premise of the pumping lemma for linear languages such that it holds

$1 \leq m_{Q_t} \leq |u|$, then ${}_{n\text{-}path}L(G, R)$ is ${}_{I\text{-}n\text{-}path}L(G, R)$,
$|u| < m_{Q_t} \leq |uv|$, then ${}_{n\text{-}path}L(G, R)$ is ${}_{II\text{-}n\text{-}path}L(G, R)$,
$|uv| < m_{Q_t} \leq |uvw|$, then ${}_{n\text{-}path}L(G, R)$ is ${}_{III\text{-}n\text{-}path}L(G, R)$,
$|uvw| < m_{Q_t} \leq |uvwx|$, then ${}_{n\text{-}path}L(G, R)$ is ${}_{IV\text{-}n\text{-}path}L(G, R)$,
$|uvwx| < m_{Q_t} \leq |uvwxy|$, then ${}_{n\text{-}path}L(G, R)$ is ${}_{V\text{-}n\text{-}path}L(G, R)$.

Definition 3.24 (Classes of special types of n-path tree controlled languages)**.** For $i \in \{\mathbf{I, II, III, IV, V}\}$ and $n \geq 1$, the *class of i-n-path tree controlled languages* is defined as

$$i\textbf{-n-path-TC}(\mathbf{CF}, \mathbf{LIN}) = \{{}_{i\text{-}n\text{-}path}L(G, R) : (G, R) \text{ is tree controlled grammar in which } G \text{ is a context-free grammar and } R \in \mathbf{LIN}\}.$$

3.3.2 Results

It is well-know that path controlled grammars where the controlling grammar is regular characterize the same language class as its controlled grammar (see [32]) do. We have proved that the generative power of context-free grammars remains unchanged even if we restrict all paths in their derivation trees by regular languages:

$$\mathbf{CF} = \textbf{all-path-TC}(\mathbf{CF}, \mathbf{REG}).$$

Since for each context-free grammar, there is a regular language that describes all paths in all its derivation trees; and there is no regular language which increases its generative power when used to restrict the paths, if we consider tree controlled grammars (G, R) with $R \in \mathbf{REG}$, then, obviously, the generative power of such a model equals **CF** for any $n \geq 1$. Therefore, we investigate the properties of tree controlled grammar with non-regular control language. More specifically, we study tree controlled grammars that generates their languages under n-path control with linear control languages.

We have introduced a generalization of path controlled grammars so that they generate the language under the restriction placed on not just one but several paths. Consequently, we have found some subsets of n-path controlled grammars so their languages satisfy pumping premises similar to well-known premises stated by pumping lemmata for **CF**, **LIN**, and **REG**—more precisely:

- If $L \in \mathbf{I\text{-}n\text{-}path\text{-}TC}(\mathbf{CF}, \mathbf{LIN})$, $n \geq 1$, then there are two constants, $k, q \geq 0$, such that each word, $z \in L$, with $|z| \geq k$ can be written as $z = u_1v_1u_2v_2 \dots u_{4n}v_{4n}u_{4n+1}$ with $0 < |v_1v_2 \dots v_{4n}| \leq q$ and for all $i \geq 1$, $u_1v_1^iu_2v_2^i \dots u_{4n}v_{4n}^iu_{4n+1} \in L$.
- If $L \in \mathbf{III\text{-}n\text{-}path\text{-}TC}(\mathbf{CF}, \mathbf{LIN})$, $n \geq 1$, then there are two constants, $k, q \geq 0$, such that each word, $z \in L$, with $|z| \geq k$ can be written as $z = u_1v_1u_2v_2 \dots u_{2n+2}v_{2n+2}u_{2n+3}$ with $0 < |v_1v_2 \dots v_{2n+2}| \leq q$ and for all $i \geq 1$, $u_1v_1^iu_2v_2^i \dots u_{2n+2}v_{2n+2}^iu_{2n+3} \in L$.
- If $L \in \mathbf{V\text{-}n\text{-}path\text{-}TC}(\mathbf{CF}, \mathbf{LIN})$, $n \geq 1$, then there are two constants, $k, q \geq 0$, such that each word, $z \in L$, with $|z| \geq k$ can be written as $z = u_1v_1u_2v_2u_3v_3u_4v_4u_5$ with $0 < |v_1v_2v_3v_4| \leq q$ and for all $i \geq 1$, $u_1v_1^iu_2v_2^iu_3v_3^iu_4v_4^iu_5 \in L$.

A natural question that still remains open is whether or not there are similar pumping properties for the languages of $\mathbf{II\text{-}n\text{-}path\text{-}TC}(\mathbf{CF}, \mathbf{LIN})$ and $\mathbf{IV\text{-}n\text{-}path\text{-}TC}(\mathbf{CF}, \mathbf{LIN})$.

We have also proved some closure properties, that is

- for $n \geq 1$, $i \in \{$**I, II, III, IV, V**$\}$ and $T_G, T_R \in \{\mathbf{REG}, \mathbf{LIN}, \mathbf{CF}\}$, $\mathbf{n\text{-}path\text{-}TC}(T_G, T_R)$, i**-n-path-TC**(T_G, T_R) are closed under intersection with regular languages, union, and non-erasing homomorphism,
- for $n \geq 1$, $\mathbf{I\text{-}n\text{-}path\text{-}TC}(\mathbf{CF}, \mathbf{LIN})$, $\mathbf{III\text{-}n\text{-}path\text{-}TC}(\mathbf{CF}, \mathbf{LIN})$, and $\mathbf{V\text{-}n\text{-}path\text{-}TC}(\mathbf{CF}, \mathbf{LIN})$ are not closed under concatenation, intersection, and complement.

Since n-path controlled grammars are a natural generalization of grammars with just one path controlled, we have studied several properties that are well-known for path controlled grammars in the case of controlling given n paths. Most importantly, we have tried to establish the generative power for n path controlled grammar. We have found the approximation of the generative power that can be applied also on grammars with just one path controlled. However, this approximation does not say too much since it is well-known that **PSC** is not closed under erasing homomorphism. Thus, we have informally concluded that: *„We either have the power to check what we need but not to remove it (using* **PSC***) or vice versa (using* **MAT***).“* More precisely, we have stated the following:

Let $L \in \mathbf{n\text{-}path\text{-}TC}(\mathbf{CF}, \mathbf{CF})$, for $n \geq 1$. Then there exists $L' \in \mathbf{PSC}$ with $L = h(L')$, for some homomorphism h.

Finally, we have studied several parsing properties of path-based restriction which is indisputably one of the most important language-class-characterizing property from the practical viewpoint. Formally, we have studied a polynomial time parsing possibilities and we have stated that:

- For a tree controlled grammar, (G, R) with an unambiguous context-free grammar, G, and a linear control language, R, the membership $x \in {}_{nc\text{-}n\text{-}path}L(G, R)$, $n \geq 1$, is decidable in $O(|x|^k)$, for some $k \geq 0$.

- For a tree controlled grammar, (G, R), where G is a context-free grammar and $R \in$ **LIN**, there is a tree controlled grammar, (G', R'), such that G' does not contain unit productions and ${}_{nc\text{-}n\text{-}path}L(G, R) = {}_{nc\text{-}n\text{-}path}L(G', R')$, $n \geq 1$.
- For tree controlled grammar (G, R) where G is m-ambiguous LR grammar, $m \geq 1$, and an unambiguous language $R \in$ **LIN** , the membership $x \in {}_{nc\text{-}n\text{-}path}L(G, R)$, $n \geq 1$, is decidable in $O(|x|^k)$, for some $k \geq 0$.

The significant disadvantage of n-path tree controlled grammars is that the number of n paths satisfying the properties of Def. 3.21 is strictly limited by the length of the right-hand sides of the productions of underlying context-free grammar. That is, given a general context-free grammar, G, and a linear language, R, controlling the paths, the membership of a certain language might be decidable. However, given the same context-free language as $L(G)$ as a context-free grammar, H, in Chomsky normal form together with R, we might not be able to find suitable path restriction to obtain the same language. On the other hand, the derivation trees of tree controlled grammars that generates their languages under n-path control by a linear language are constructed exactly as in context-free grammars and, in addition, we have to check some of their paths. Thus, there is actually great possibility to use well-known parsing methods for context-free languages to construct the derivation trees and to check their paths. However, in this viewpoint, n-path tree controlled grammars seems to be a quite fragile formalism since it requires a context-free grammar to have a production with at least n nonterminals on the right-hand side which ensures the division of n paths in a common node. Moreover, it means that any attempt to use a parsing method that transforms a context-free grammar into Chomsky normal form will basically destroy any path restriction with $n \geq 3$. Moreover, several nice properties of context-free grammars have been lost—e.g., decomposition based on pumping lemma for linear languages is potentially ambiguous and thus, the membership problem for i-**n-path-TC**, $i \in \{$**I, II, III, IV, V**$\}$, is potentially ambiguous also.

There are still many questions to be answered, namely generative power of grammars with path or paths controlled non-regularly, further closure properties, decision properties, etc. However, there are several other more general variants of path-based restriction. Indeed, a modification of the formalism such that the paths do not have to be divided in a common node of a derivation tree, or a variant where a path in a tree does not have to start in the root and end in a leaf of the tree have not been investigated yet.

Abstract

This doctoral thesis studies theoretical properties of grammars with restricted derivation trees. After presenting the state of the art concerning this investigation area, the research is focused on the three main kinds of the restrictions placed upon the derivation trees. First, it introduces completely new investigation area represented by cut-based restriction and examines the generative power of the grammars restricted in this way. Second, it investigates several new properties of path-based restriction placed upon the derivation trees. Specifically, it studies the impact of *erasing productions* on the generative power of grammars with restricted path and introduces two corresponding normal forms. Then, it describes a new relation between grammars with restricted path and some pseudoknots. Next, it presents a counterargument to the generative power of grammars with controlled path that has been considered as well-known so far. Finally, it introduces a generalization of path-based restriction to not just one but several paths. The model generalized in this way is studied, namely its pumping, closure, and parsing properties.

Keywords

tree controlled grammars, level controlled grammars, path controlled grammars, paths controlled grammars, cut controlled grammars, ordered cut controlled grammars, regulated rewriting, restricted derivation trees

Bibliography

[1] S. Abraham. Some questions of language theory. In *Proceedings of the 1965 conference on Computational linguistics*, 1965.

[2] A. V. Aho. Indexed grammars - an extension of context-free grammars. *Journal of the ACM*, 15:25, 1968.

[3] A. V. Aho, R. Sethi, and J. D. Ullman. *Compilers principles, techniques, and tools.* Addison-Wesley, 1986.

[4] A. V. Aho and J. D. Ullman. *The theory of parsing, translation, and compiling.* Prentice-Hall, Inc., 1972.

[5] A. Bondy. *Graph Theory.* Springer, 2010.

[6] R. V. Book. On the structure of context-sensitive grammars. *International Journal of Computer and Information Sciences*, 2:11, 1973.

[7] M. Čermák, J. Koutný, and A. Meduna. Parsing based on n-path tree-controlled grammars. *Theoretical and Applied Informatics*, 2011:16, 2012.

[8] A. B. Cremers. Normal forms for context-sensitive grammars. *Acta Informatica*, 3:15, 1973.

[9] K. Čulik and H. A. Maurer. Tree controlled grammars. *Computing*, 19:11, 1977.

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

[11] J. Dassow, R. Stiebe, and B. Truthe. Two collapsing hierarchies of subregularly tree controlled languages. *Theoretical Computer Science*, 410:11, 2009.

[12] J. Dassow and B. Truthe. On two hierarchies of subregularly tree controlled languages. In *10th International Workshop on Descriptional Complexity of Formal Systems, DCFS 2008, Charlottetown, Prince Edward Island, Canada, July 16-18, 2008*, 2008.

[13] J. Dassow and B. Truthe. Subregularly tree controlled grammars and languages. In *Automata and Formal Languages - 12th International Conference AFL 2008, Balatonfured*, 2008.

[14] I. Friš. Grammars with partial ordering of the rules. *Information and Control*, 12:11, 1968.

[15] J. Higginbotham. English is not a context-free language. *Linguistic Inquiry*, 15:10, 1984.

[16] A. K. Joshi. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions. Technical report, University of Pennsylvania, 1985.

[17] J. Koutný. L-systems and their usage in practice. Bachelor thesis, Faculty of Information Technology, Brno University of Technology, 2006.

[18] J. Koutný. L-systems and their applications. Master's thesis, Faculty of Information Technology, Brno University of Technology, 2008.

[19] J. Koutný. L-systémy a jejich aplikace. In *Proceedings of the 14th Conference STUDENT EEICT 2008*, 2008.

[20] J. Koutný. Regular paths in derivation trees of context-free grammars. In *Proceedings of the 15th Conference STUDENT EEICT 2009 Volume 4*, 2009.

[21] J. Koutný. On n-path-controlled grammars. In *Proceedings of the 16th Conference STUDENT EEICT 2010 Volume 5*, 2010.

[22] J. Koutný. Syntax analysis of tree-controlled languages. In *Proceedings of the 17th Conference STUDENT EEICT 2011 Volume 3*, 2011.

[23] J. Koutný. On path-controlled grammars and pseudoknots. In *Proceedings of the 18th Conference STUDENT EEICT 2012 Volume 3*, 2012.

[24] J. Koutný, Z. Křivka, and A. Meduna. On grammars with controlled paths. Acta Cybernetica, submitted.

[25] J. Koutný, Z. Křivka, and A. Meduna. Pumping properties of path-restricted tree-controlled languages. In *7th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, 2011.

[26] J. Koutný and A. Meduna. On normal forms and erasing rules in path controlled grammars. Schedae Informaticae, in press.

[27] J. Koutný and A. Meduna. Tree-controlled grammars with restrictions placed upon cuts and paths. *Kybernetika*, 48:11, 2012.

[28] M. K. Levitina. On some grammars with rules of global replacement. *Scientific-technical information (Nauchno-tehnichescaya informacii), Series 2*, page 5, 1972.

[29] A. Lindenmayer. Mathematical models for cellular interactions in development, I & II. *Journal of Theoretical Biology*, 18:36, 1968.

[30] A. Lindenmayer. Developmental systems without cellular interactions, their languages and grammars. *Journal of Theoretical Biology*, 30:30, 1971.

[31] A. Lindenmayer and G. Rozenberg. Developmental systems and languages. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1972.

[32] S. Marcus, C. Martín-Vide, V. Mitrana, and Gh. Păun. A new-old class of linguistically motivated regulated grammars. In *Walter Daelemans, Khalil Sima'an, Jorn Veenstra, Jakub Zavrel (Eds.): Computational Linguistics in the Netherlands 2000, Selected Papers from the Eleventh CLIN Meeting, Tilburg*, 2000.

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

[34] C. Martin-Vide and V. Mitrana. Decision problems on path-controlled grammars. *International Journal of Foundations of Computer Science*, 18:11, 2007.

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

[36] A. Meduna. *Elements of Compiler Design*. Taylor and Francis Informa plc, 2008.

[37] P. Navrátil. Parsing based on regulated grammars. Master's thesis, Faculty of Information Technology, Brno University of Technology, 2003.

[38] G. Păun. On the generative capacity of simple matrix grammars of finite index. *Information Processing Letters*, 7:3, 1978.

[39] Gh. Păun. On the generative capacity of tree controlled grammars. *Computing*, 21:8, 1979.

[40] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag, 1990.

[41] G. K. Pullum. On two recent attempts to show that english is not a CFL. *Computational Linguistics*, 10:5, 1984.

[42] G. K. Pullum and G. Gazdar. Natural languages and context-free languages. *Linguistics and Philosophy*, 4:34, 1982.

[43] Jr. R. L. Cannon. An algebraic technique for context-sensitive parsing. *International Journal of Computer and Information Sciences*, 5:20, 1976.

[44] D. J. Rosenkrantz. Programmed grammars and classes of formal languages. *Journal of the ACM*, 16:25, 1969.

[45] A. Salomaa. Matrix grammars with a leftmost restriction. *Information and Control*, 20:7, 1972.

[46] S. Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:11, 1985.

[47] R. Stiebe. On the complexity of the control language in tree controlled grammars. In *Colloquium on the Occasion of the 50th Birthday of Victor Mitrana*, 2008.

[48] S. Turaev, J. Dassow, and M. H. Selamat. Language classes generated by tree controlled grammars with bounded nonterminal complexity. In *Descriptional Complexity of Formal Systems - 13th International Workshop, DCFS 2011, Gießen/Limburg, Germany, July 25-27, 2011. Proceedings*, 2011.

[49] S. Turaev, J. Dassow, and M. H. Selamat. Nonterminal complexity of tree controlled grammars. *Theoretical Computer Science*, 412:7, 2011.

[50] A. P. J. van der Walt. Random context grammars. In *Proceedings of Symposium on Formal Languages*, 1970.

[51] B. Wegbreit. A generator of context-sensitive languages. *Journal of Computer and System Sciences*, 3:6, 1969.

[52] W. A. Woods. Context-sensitive parsing. *Communications of the ACM*, 13:9, 1970.

[53] G. Zetzsche. Toward understanding the generative capacity of erasing rules in matrix grammars. *International Journal of Foundations of Computer Science*, 22:16, 2011.

Curriculum Vitae

Personal Data

Name:	Jiří Koutný
E-mail:	ikoutny@fit.vutbr.cz
Homepage:	www.fit.vutbr.cz/~ikoutny
Date of Birth:	3^{rd} May 1984

Experiences

Microprocessor Models and Documentation, Codasip Division
ApS Brno
2011 — Present, Brno, Czech Republic

Technical Staff
Brno University of Technology Faculty of Information Technology
2008 — Present, Brno, Czech Republic

Technical Literature Translations
Computer Press
2008 — 2011, Brno, Czech Republic

External Redactor
Mlada fronta
1999 — 2011, Brno, Czech Republic

Decompiler Group
AVG Technologies
November 2011, Brno, Czech Republic

Technical Literature Translations
Zoner Press
2007 — 2009, Brno, Czech Republic

Junior Developer
Kentico Software
June 2006 — December 2006, Brno, Czech Republic

Study/Research/Teaching Stays

University of Texas
June 2012 (1 week) Austin, Texas, U.S.

University of Valladolid
May 2012 (1 week) Segovia, Spain

Universite Paris-Est Marne-la-Valle
April 2012 (4 days) Paris, France

University of Debrecen
April 2012 (4 days) Debrecen, Hungary

Kyoto University
November 2011 — December 2011 (3 weeks) Kyoto, Japan

University of Debrecen
August 2011 (1 week) Debrecen, Hungary

Hungarian Academy of Sciences
May 2011 — June 2011 (3 weeks) Budapest, Hungary

Universite Paris-Est Marne-la-Valle
December 2010 (1 week) Paris, France

University of Debrecen
September 2010 (1 week) Debrecen, Hungary

University of Western Ontario
August 2010 (1 week) London, Canada

Institut de Gestion — Universite de La Rochelle
June 2010 — July 2010 (1 week) La Rochelle, France

University of Valladolid
May 2010 (1 week) Segovia, Spain

Academy of Sciences of the Czech Republic
May 2010 (1 week) Brno, Czech Republic

Vrij Technisch Instituut Izegem
April 2002 (2 weeks) Izegem, Belgium

Education

Brno University of Technology Faculty of Information Technology
PhD. (in progress), Formal Language Theory (2008 — 2012)
Doctoral Degree Programme: Computer Science and Engineering
PhD Thesis: Grammars with Restricted Derivation Trees

Brno University of Technology Faculty of Information Technology
Ing., Computer Graphics and Multimedia (2006 — 2008)
Master Degree Study Programme: Information Technology
Study Specialisation: Computer Graphics and Multimedia
MSc Thesis: L Systems and Their Applications

Brno University of Technology Faculty of Information Technology
Bc., Information Technology (2003 — 2006)
Bachelor Degree Study Programme: Information Technology
BSc Thesis: L Systems and Their Practical Usage

Secondary Technical School in Zlin
Graduation, Electrical Engineering (1999 — 2003)
Study Programme: Electrical Engineering
Study Specialisation: Electronical Computer Systems

Honors and Awards

EEICT 2012 Conference and Competition: 1st place
EEICT 2011 Conference and Competition: 2st place
EEICT 2010 Conference and Competition: 1st place
EEICT 2009 Conference and Competition: 1st place
EEICT 2008 Conference and Competition: 2st place

Memberships

Formal Model Research Group at FIT BUT
2008 — Present

Czechoslovakia Section of Institute of Electrical and Electronics Engineers
2009, 2010, 2012

List of scientific publications in FIT BUT

1. M. Čermák, J. Koutný, and A. Meduna. On n-language classes hierarchy. Acta Cybernetica, submitted.

2. M. Čermák, J. Koutný, and A. Meduna. Parsing based on n-path tree-controlled grammars. *Theoretical and Applied Informatics*, 2011:16, 2012.

3. J. Koutný, Z. Křivka, and A. Meduna. On grammars with controlled paths. Acta Cybernetica, submitted.

4. J. Koutný and A. Meduna. On normal forms and erasing rules in path controlled grammars. Schedae Informaticae, in press.

5. J. Koutný and A. Meduna. Tree-controlled grammars with restrictions placed upon cuts and paths. *Kybernetika*, 48:11, 2012.

6. J. Koutný, Z. Křivka, and A. Meduna. Pumping properties of path-restricted tree-controlled languages. In *7th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, 2011.

7. J. Koutný. Syntax analysis of tree-controlled languages. In *Proceedings of the 17th Conference STUDENT EEICT 2011 Volume 3*, 2011.

8. J. Koutný. On n-path-controlled grammars. In *Proceedings of the 16th Conference STUDENT EEICT 2010 Volume 5*, 2010.

9. J. Koutný. On path-controlled grammars and pseudoknots. In *Proceedings of the 18th Conference STUDENT EEICT 2012 Volume 3*, 2012.

10. J. Koutný. Regular paths in derivation trees of context-free grammars. In *Proceedings of the 15th Conference STUDENT EEICT 2009 Volume 4*, 2009.

11. J. Koutný. L-systémy a jejich aplikace. In *Proceedings of the 14th Conference STUDENT EEICT 2008*, 2008.

List of project participation in FIT BUT/BUT

Advanced recognition and presentation of multimedia data
FIT VUT, FIT-S-11-2, 2011-2013, running
Research leader: Doc. Dr. Ing. Pavel Zemcik

Context-free languages and pushdown automata
MŠMT, MEB041003, 2010-2011, completed
Research leader: prof. RNDr. Alexander Meduna, CSc.

Recognition and presentation of multimedia data
FIT VUT, FIT-S-10-2, 2010, 2010, completed
Research leader: Doc. Dr. Ing. Pavel Zemcik

Vyuka pokrocilych temat teoreticke informatiky na mezinarodni urovni
FRVŠ MŠMT, FR2581/2010/G1, 2010, completed
Research leader: **Ing. Jiri Koutny**

Security-Oriented Research in Information Technology
CEZ MŠMT, MSM0021630528, 2009, running
Research leader: **Prof. Ing. Tomas Hruska**

Mathematical and Engineering Approaches to Developing Reliable and Secure Concurrent and Distributed Computer Systems
GAČR, GD102/09/H042, 2009-2012, running
Research leader: prof. RNDr. Milan Ceska, CSc.

Sit na podporu spoluprace technicky a podnikatelsky zamerenych univerzit s podniky v Jihomoravskem kraji
CZ.1.07/2.4.00/12.0017, 2009-2012, running
April 2011 — April 2011 (1 month) Brno, Czech Republic
July 2011 — August 2011 (2 months) Brno, Czech Republic
Research leader: doc. RNDr. Miloslav Svec, CSc.