Transformation of Formal Models

Dissertation Thesis

BUT FIT

Petr Zemek

Faculty of Information Technology, Brno University of Technology

January 3, 2012

Petr Zemek

Transformation of Formal Models

Dissertation Thesis

January 3, 2012

Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic

Abstract

In this work, we declare the focus of our dissertation thesis by presenting the main topic, declare specific focus and goals, describe viable types of transformations, and give the current state of the art. A brief summary of the obtained results is given. We argue that the achieved results are important both from a theoretical and practical viewpoint.

Contents

1

Introduction

Formal languages fulfill a crucial role in a great number of scientific areas, ranging from linguistics through molecular genetics up to most fields of informatics [107]. Indeed, the theory of formal languages provides formal models to define and manipulate (formal) languages, which represent a suitable form of abstraction and representation of many real-world problems. Indisputably, grammars and automata belong among the most common language-defining models.

As a specific part, formal language theory studies *transformations of formal models*—the topic of this work. As presented next, transformations of formal models play a vital role both from a theoretical and practical viewpoint. We pay a special attention to algorithms that arrange formal models so they satisfy some prescribed desirable properties while the generated languages remain unchanged.

For example, consider transformations into normal forms. A formal model is in a *normal form* if all its rules have some prescribed uniform form. If the model posses more sets of rules, like in grammar systems (see [11]), conditions may also be placed upon the identity or disjointness of these sets. From a theoretical viewpoint, normal forms are useful in simplifying proofs of results [17, 73, 105]. From a practical viewpoint, their usage may result in more efficient construction of parsers [1, 76].

As another example, elimination of *erasing rules*—that is, rules with the empty string on their right-hand side—often fulfills a key role in formal grammars and their applications. From a practical point of view, some grammar-based methods, including many parsing algorithms, strictly require the absence of any erasing rules in the applied grammars [76]. From a theoretical point of view, the absence of erasing rules frequently simplifies the achievement of various results concerning grammars and their languages [17, 73, 105, 106].

Among yet other types of transformation belong conversions between related formal models, reduction of the size of their components, and generation of languages extended by some extra symbols that represent useful information. A detailed description of all the mentioned transformations is given in Chapter 2.

1.1 Goals

The goal of this work is to develop new transformations of formal models. Whenever appropriate, new formal models are introduced. Thereby, this work contributes to the theory of formal languages.

1.2 Focus

The variety of different formal models introduced over the past century is so huge that we cannot cover them all. Instead, we restrict our main attention to *regulated formal models* (see [17, 62, 95, 105]). In essence, regulated formal models represent classical models extended by additional mathematical mechanisms by which they control the use of rules during the generation of their languages.

In greater detail, we mainly focus on *regulated grammars* (see [17, 62, 95, 105]), *regulated automata* (see [15, 37, 47–49, 68, 95]), and *regulated grammar systems* (see [5, 12, 13, 23, 27, 55–58, 95, 101]). Nevertheless, several interesting transformations of classical and non-regulated formal models are also investigated.

1.3 Ways of Realization

The goals of this work are realized by publishing papers in international journals and conferences. In this way, all results undergo a critical reviewing process, which represent the very first feedback from the scientific community. Furthermore, this approach may help to disseminate the obtained results more rapidly.

1.4 Organization

This work is organized as follows. After this introductory chapter, Chapter 2 describes several types of transformation of formal models and gives a motivation for studying them. Then, Chapter 3 briefly presents the current state of the art. Chapter 4, which represents the core of the work, gives a brief overview of the achieved results. Chapter 5 concludes the work.

1.5 Prerequisites

We assume that the reader is familiar with formal language theory, including the theories of regulated rewriting and parsing. For detailed information and definitions of all unexplained notions, please refer to [1, 11, 17, 33, 36, 73, 76, 104, 105, 109, 111, 116, 118]).

2

Types of Transformation and Their Motivation

In this chapter, we describe several types of transformation of formal models and give a motivation for studying them.

2.1 Normal Forms

As already mentioned in the introduction, a model is in a *normal form* if all its rules have some prescribed uniform form. If the model posses more sets of rules, like in grammar systems (see [11]), conditions may also be placed upon the identity or disjointness of these sets. From a theoretical viewpoint, normal forms are useful in simplifying proofs of results [17, 73, 105]. From a practical viewpoint, their usage may result in more efficient construction of parsers [1, 76].

2.2 Elimination of Erasing Rules

Erasing rules are rules of the form $A \rightarrow \varepsilon$, where A is a nonterminal and ε is the empty string. Whenever a grammar do not contain any erasing rules, it is said to be *propagating.* The elimination of erasing rules often fulfills a key role in formal grammars and their applications. From a practical viewpoint, some grammar-based methods, including many parsing algorithms, strictly require the absence of any erasing rules in the applied grammars [76]. From a theoretical viewpoint, the absence of any erasing rules frequently simplifies the achievement of various results concerning grammars and their languages [17, 73, 105, 106].

2.3 Reduction

From a general point of view, *descriptional complexity* studies the generative power of formal models in relation to the size of their components. Descriptional complexity represents a vivid trend in today's formal language theory as demonstrated by several recent studies (see [14, 16, 21, 22, 24, 25, 34, 59, 65, 66, 69, 70, 98, 115]). As an important part, this trend aims to *reduce* the number of nonterminals or rules in grammars, and the number of states or rules in terms of automata.

2.4 Conversions Between Related Formal Models

As obvious, we can convert formal models to other equivalent language-defining devices. By using this kind of transformation, we can thus convert one type of language-defining devices to another type so the resulting type of these devices satisfy some prescribed properties needed under given circumstances concerning the defined language family. It thus comes as no surprise that these transformations are deeply appreciated both from a theoretical and practical viewpoint.

2.5 Generation of Extended Languages

Regulated formal models are sometimes modified so they generate their languages extended by some extra symbols that represent useful information related to the generated languages. For instance, these symbols may represent information used during parsing directed by these grammars. To give another example, shuffling or inserting some symbols into the generated languages may correspond to language operations that fulfill an important role in various modern fields of informatics, ranging from cryptography through various text algorithms to DNA computation [2, 108, 110].

3

State of the Art

In this chapter, we briefly present the current state of the art—that is, we give an overview of the existing knowledge concerning the described transformations. Note, however, that the aim of this chapter is not to give an exhaustive overview. Indeed, there are so many results so we only cover some of them. For lucidity, this chapter is organized in the same way as Chapter 2.

3.1 Normal Forms

In the classical terms, the well-known normal forms of context-free grammars include the *Chomsky normal form*, *Greibach normal form*, and *operator normal form* [105]. For context-sensitive and phrase-structure grammars, there are the *Kuroda* (see [51]) and *Penttonnen* (see [99]) normal forms. The *Geffert normal forms* of phrase-structure grammars (see [26]) are also well known.

In terms of regulated formal models, various special normal forms exist. For example, consider the *2-normal form* of matrix grammars (see [17]) and the *2-limited normal form* of propagating scattered context grammars (see [31] and Section 3.1 in [79]). The results from Section 3.3 can be also seen as normal forms.

3.2 Elimination of Erasing Rules

In the case of context-free grammars, it is possible to systematically remove all erasing rules while preserving the generated language (see Section 7.3. in [36]). However, whether erasing rules can be eliminated from regulated grammars in general represents a longstanding open problem. While this question has been answered for some types of regulated grammars (for example, one can remove erasing rules from permitting grammars without affecting their generative power [122]), it still remains unanswered[1] for some

---

[1] In Theorem 2.15 in [106], it is claimed that this is impossible in the case of regular-controlled grammars, matrix grammars, and programmed grammars. However, the given references do not contain a proof for this and in more recent publications, like [62, 121, 123], this is still considered to be an open problem.

other types of regulated grammars, like regular-controlled grammars, matrix grammars, programmed grammars, and random context grammars [62].

The removal of so-called *ε-transitions* from finite and pushdown automata (see [73, 105]) can be seen as a counterpart of the elimination of erasing rules in terms of automata theory.

3.3 Reduction

Using the aforementioned Geffert normal form, one can immediately imply that any recursively enumerable can be generated by a phrase-structure grammar with no more than three nonterminals (see [26]). In the case of regulated rewriting, two nonterminals suffice for scattered context grammars to generate any recursively enumerable language [14]. Matrix and programmed grammars are computationally complete with three nonterminals (see [22]). For other results in this direction, see [16, 21, 24, 25, 34, 59, 65, 66, 69, 70, 98, 115].

As another measure, one may reduce nondeterminism in various regulated grammars. For the case of programmed grammars, see [4, 7, 9].

Moving to automata theory, it is well known that pushdown automata with a sole state (accepting by empty pushdown) still characterize the family of context-free languages (see [76]), while for deterministic pushdown automata their computational power strictly increases with the number of states (see [33]). For a survey on the complexity results concerning finite automata, see [34, 35] and Chapter 2 in the first volume of [105].

To mention some results from a different area than grammars and automata, recall that every ET0L system can be converted into an equivalent system with only two tables (see [104]), and for the terminating mode, three components suffice for every CD grammar system to generate any ET0L language (see [11]).

3.4 Conversions Between Related Formal Models

Conversions between formal models arguably represent the most studied type of transformation. Indeed, the generative power of many formal models has been established by converting them to other language-defining devices whose power is known [107]. For example, consider regular expressions, regular grammars, finite automata, and proofs of their equivalence (see [73]). Turing completeness is usually established by simulating some Turing-machine-equivalent model [107]. For an overview of some results belonging to this category, see [11, 17, 33, 61, 73, 77, 79, 84, 104, 107, 109, 111, 118].

3.5 Generation of Extended Languages

As the first form of extended languages, one can consider generation of sentences followed by their parses. This topic is systematically studied in terms of matrix grammars in Section 7.2 in [17], which refers to these languages as *extended Szilard languages* (as pointed out by Wood on page 18 in [117], the notion of a *parse* represents a synonym of several other notions, including a *Szilard word*). A related study is given in [78] in terms of scattered context grammars (see also Chapter 7 in [79]). In [85], generation of derivation trees instead of parses is considered. More generally, Szilard languages and their properties have been vividly studied in recent four decades (see [20], references given therein, and [10]). Finally, see [32] for an extension of the notion of a Szilard language in terms of P systems.

Using “stuff symbols”, one may characterize the family of recursively enumerable languages even by propagating versions of regulated grammars. This has been done in terms of scattered context grammars in [19]. Also, [74, 75] gives a related result.

4

Results

In this chapter, we give an overview of the already achieved and planned results. Sources of these results are cited at the beginning of each section.

4.1 Non-Regulated Formal Models

The results in this section are based on [52, 97].

4.1.1 Alternative Elimination of Erasing Rules from Context-Free Grammars

We describe an alternative algorithm for removing erasing rules from context-free grammars. As opposed to the standard way of eliminating erasing rules from context-free grammars (see Section 5.1.3.2 in [73]), this method requires no predetermination of symbols that derive the empty string. The proposed algorithm is formally verified. In the conclusion of this section, the applicability of the algorithm to context-free grammars that work in a semi-parallel way is demonstrated. Furthermore, two open problem areas are formulated.

4.1.2 New Normal Form of Phrase-Structure Grammars

Traditionally, normal forms of phrase-structure grammars reduce the number of context-sensitive rules while introducing many context-free rules [105]. In this section, we go into an opposite direction. Indeed, we introduce a new normal form of phrase-structure grammar in which the number of context-free rules is linearly limited by the number of terminals of the generated language. More precisely, for every recursively enumerable language K, there exists a phrase-structure grammar G so that G generates K and contains only $2 + t$ context-free rules, where t is the number of terminals of K. (Of course, the remaining rules are context-sensitive.)

4.2 Regular-Controlled Grammars

The results in this section are based on [93, 97] and Sections 8.2 and 9.1 in [95].

4.2.1 Workspace Theorems

We establish a workspace theorem in terms of regular-controlled grammars (see [17, 106]). We prove that if for a regular-controlled grammar H, there is a positive integer k such that H generates every sentence $y \in L(H)$ by a derivation in which every sentential form x contains at most $(k-1)|x|/k$ occurrences of nonterminals that are erased throughout the rest of the derivation, where $|x|$ denotes the length of x, then the language of H is generated by a propagating regular-controlled grammar. An analogical workspace theorem is demonstrated for regular-controlled grammars with appearance checking. We provide an algorithm that removes all erasing rules from any regular-controlled grammar (possibly with appearance checking) that satisfies the workspace condition above without affecting the generated language. In the conclusion, we point out a relationship of the workspace theorems to other areas of formal language theory.

4.2.2 Generation of Sentences with Their Parses

We explain how to transform any regular-controlled grammar with appearance checking G to a propagating regular-controlled grammar with appearance checking G' whose language $L(G')$ has every sentence of the form $w\rho$, where w is a string of terminals in G and ρ is a sequence of rules in G', so that (i) $w\rho \in L(G')$ if and only if $w \in L(G)$ and (ii) ρ is a parse of w in G'. Consequently, for every recursively enumerable language K, there exists a propagating regular-controlled grammar with appearance checking G' with $L(G')$ of the above form so K results from $L(G')$ by erasing all rules in $L(G')$. In addition, analogical results are established (a) in terms of these grammars without appearance checking and (b) in terms of these grammars that make only leftmost derivations. In the conclusion, we point out some consequences implied by the results achieved in this section.

4.3 One-Sided Random Context Grammars

The results in this section are based on [87, 91, 92, 94, 96, 119] and Section 10.2 in [95].

The notion of a *one-sided random context grammar* is defined as a context-free-based regulated grammar, in which a set of *permitting symbols* and a set of *forbidding symbols* are attached to every rule, and its set of rules is divided into the set of *left random context rules* and the set of *right random context rules*. A left random context rule can rewrite a nonterminal if each of its permitting symbols occurs to the left of the rewritten symbol in the current sentential form while each of its forbidding symbols does not occur there. A right random context rule is applied analogically except that the symbols are examined to the right of the rewritten symbol.

We demonstrate that without erasing rules, one-sided random context grammars characterize the family of context-sensitive languages, and with erasing rules, these grammars characterize the family of recursively enumerable languages. In fact, these characterization results hold even if the set of left random context rules coincides with the set of right random context rules. Several special cases of these grammars are considered, and their generative power is established. In the conclusion, some important open problems are suggested to study in the future.

4.3.1 Leftmost Derivations

We study the generative power of one-sided random context grammars working in a leftmost way. More specifically, by analogy with the three well-known types of leftmost derivations in regulated grammars (see Section 1.4 in [17]), we introduce three types of leftmost derivations to one-sided random context grammars and prove the following three results. (I) One-sided random context grammars with type-1 leftmost derivations characterize the family of context-free languages. (II) One-sided random context grammars with type-2 and type-3 leftmost derivations characterize the family of recursively enumerable languages. (III) Propagating one-sided random context grammars with type-2 and type-3 leftmost derivations characterize the family of context-sensitive languages. In the conclusion, the generative power of random context grammars and one-sided random context grammars with leftmost derivations is compared.

4.3.2 Reduction of the Number of Nonterminals

We study the nonterminal complexity of one-sided random context grammars. More specifically, we prove that every recursively enumerable language can be generated by a one-sided random context grammar with no more than ten nonterminals. An analogical result holds for thirteen nonterminals in terms of these grammars with the set of left random context rules coinciding with the set of right random context rules. Furthermore, we introduce the notion of a *right random context nonterminal*, defined as a nonterminal that appears on the left-hand side of a right random context rule. We demonstrate how to convert any one-sided random context grammar G to an equivalent one-sided random context grammar H with two right random context nonterminals. An analogical conversion is given in terms of (1) propagating one-sided random context grammars and (2) *left random context nonterminals.* In the conclusion, two open problems are stated.

4.3.3 Normal Forms

We give an overview of existing normal forms of one-sided random context grammars, and establish three new normal forms. More specifically, in the solely existing normal form, the set of left random context rules coincides with the set of right random context rules. The first one represents an analogy of the Chomsky normal form of context-free grammars. In the second new normal form, each rule has its permitting or forbidding context empty, i.e. no rule can both permit and forbid symbols. In the third normal form, the sets of left and right random context rules are disjoint. All normal forms are established in terms of one-sided random context grammars with and without erasing rules.

4.3.4 One-Sided Forbidding Grammars

In *one-sided forbidding grammars*, the set of rules is divided into the set of *left forbidding rules* and the set of *right forbidding rules.* A left forbidding rule can rewrite a nonterminal if each of its forbidding symbols is absent to the left of the rewritten symbol in the current sentential form while a right forbidding rule is applied analogically except that this absence is verified to the right. Apart from this, they work like ordinary forbidding grammars (see [17, 54, 62, 66, 100, 106, 114]).

We prove that one-sided forbidding grammars are equivalent to selective substitution grammars (see [18, 28–30, 39–45, 102, 103, 112, 113]). This equivalence is established in terms of grammars with and without erasing rules. Furthermore, the paper proves that one-sided forbidding grammars in which the set of left forbidding rules coincides with the set of right forbidding rules characterize the family of context-free languages. In the conclusion, the significance of the achieved results is discussed. More specifically, we argue that one-sided forbidding grammars can formally and elegantly simulate processing information in molecular genetics, including information concerning macromolecules, such as DNA, RNA, and polypeptides.

4.3.5 Generalized One-Sided Forbidding Grammars

In *generalized one-sided forbidding grammars*, each context-free rule has associated a finite set of *forbidding strings*, and the set of rules is divided into the set of *left forbidding rules* and the set of *right forbidding rules*. A left forbidding rule can rewrite a nonterminal if each of its forbidding strings is absent to the left of the rewritten symbol in the current sentential form while a right forbidding rule is applied analogically except that this absence is verified to the right. Apart from this, they work like any generalized forbidding grammar (see [64, 68, 72, 83, 84]).

We prove the following three results. (I) Generalized one-sided forbidding grammars in which each forbidding string consists of no more than two symbols characterize the family of recursively enumerable languages. (II) Generalized one-sided forbidding grammars in which the rules in one of the two sets of rules contain only ordinary context-free rules without any forbidding strings characterize the family of context-free languages. (III) Generalized one-sided forbidding grammars with the set of left forbidding rules coinciding with the set of right forbidding rules characterize the family of context-free languages.

4.4 Left Random Context ET0L Grammars

The results in this section are based on [89, 90, 120].

Consider ET0L grammars (see [84, 104]). Modify them so a set of *permitting symbols* and a set of *forbidding symbols* are attached to each of their rules, just like in random context grammars. A rule like this can rewrite a symbol if each of its permitting symbols occurs to the left of the rewritten symbol in the current sentential form while each of its forbidding symbols does not occur there. ET0L grammars modified in this way are referred to as *left random context ET0L grammars*.

We prove that these grammars characterize the family of recursively enumerable languages, and without erasing rules, they characterize the family of context-sensitive languages. We also introduce a variety of special cases of these grammars and establish their generative power. In the conclusion, we put all the achieved results into the context of formal language theory as a whole and formulate several open questions.

4.4.1 Reduction of the Number of Nonterminals

We study the nonterminal complexity of left random context E0L grammars, which represent a variant of left random context ET0L grammars. More specifically, we prove that every recursively enumerable language can be generated by a left random context E0L grammar with no more than seven nonterminals.

4.5 Programmed Grammars

The results in this section are based on [82] and Section 7.2 in [95].

4.5.1 Normal Forms

We establish three normal forms of programmed grammars. In the first form, each rule has a nonempty success field and a nonempty failure field. In the second form, the set of rules is divided into two disjoint subsets, P_1 and P_2. In P_1, the rules have precisely one rule in its success field and precisely one rule in its failure field. In P_2, each rule has no rule in its success field and no more than two rules in its failure field. Finally, in the third form, the right-hand side of each rule consists either of the empty string, a single terminal, a single nonterminal, or a two-nonterminal string.

4.5.2 Reduction of Nondeterminism

We discuss nondeterministic behavior of programmed grammar and its reduction. We prove that for every programmed grammar, there exists an equivalent programmed grammar where only a single rule has more than one successor. This result can be seen as a new normal form of programmed grammars. Furthermore, we establish an infinite hierarchy of language families resulting from the cardinality of successor sets. Open problem areas are formulated in the conclusion of this section.

4.6 Jumping Finite Automata

The results in this section are based on [88].

We propose a new investigation area in automata theory—*jumping finite automata.* These automata work like classical finite automata except that they read input words discontinuously—that is, after reading a symbol, they can jump over some symbols within the words and continue their computation from there. We establish several results concerning jumping finite automata in terms of commonly investigated areas of automata theory, such as decidability and closure properties. Most importantly, we achieve several results that demonstrate differences between jumping finite automata and classical finite automata. In the conclusion, we formulate several open problems and suggest future investigation areas.

4.7 Regulated Pure Grammar Systems

The results in this section are based on [86].

We introduce and investigate a regulated version of pure grammar systems (see [3, 6, 8, 11]). *Regulated pure grammar systems* (1) contain n sets of rules (*components*), where n is a positive integer, (2) rules from each set can rewrite only symbols in the corresponding part of the n-tuple of strings (a *configuration*), (3) there is no distinction between nonterminals and terminals (just like in pure grammars, see [60, 63, 71] and page 242 in [105]), (4) they always rewrite the leftmost occurrence of a symbol, and (5) the use of their rules is regulated by a *control language*, as in regular-controlled grammars.

We prove the following three results. (I) Unregulated versions of these systems characterize only a proper subset of the family of context-free languages. (II) Regulated pure-grammar systems controlled by regular languages characterize the family of recursively enumerable languages. In fact, this characterization holds even if these systems have no more than two components. (III) Regulated pure-grammar systems over unary alphabets controlled by languages from the family of regular-controlled languages (see [17, 106]) generate only regular languages.

4.8 LL Random Context Grammars

The results in this section are based on [80].

We investigate LL versions of random context grammars. We prove that they generate the family of LL context-free languages (see [76]). Taking a finer look at this generation, we also demonstrate that the generation of languages by LL random context grammars is more succinct than that by LL context-free grammars. We also prove that for any random context grammar working in the leftmost way, it is decidable whether the grammar represents an LL random context grammar. A formulation of several open problems closes this section.

4.9 LL k-Linear Scattered Context Grammars

The results in this section are based on [81].

We introduce a new variant of a scattered context grammar (see [14, 24, 25, 31, 38, 46, 50, 53, 67, 79, 115]), called an LL leftmost k-linear scattered context grammar. It is an ordinary scattered context grammar without erasing rules, where (1) every scattered context rule is composed of k-linear rules, (2) if we take the first components of every rule, the resulting context-free grammar is an LL grammar, and (3) every rule is applied in a leftmost way. We study the generative power of this variant and its parsing properties, including time and space complexity. In the conclusion, several remarks regarding the achieved results are made.

5

Conclusion

In this work, we have declared the focus of our dissertation thesis by presenting the main topic (transformation of formal models), declared specific focus and goals, described viable types of transformations, and gave the current state of the art. A brief summary of the obtained results has also been given. We see that the achieved results are important both from a theoretical and practical viewpoint.

The results from [81, 82, 92, 96, 97] has been already published in international journals and international conferences. The remaining results [52, 80, 86–91, 93, 94, 119] are either in the reviewing process, or as unpublished manuscripts. Some further results, like regulated versions of finite automata and ordered versions of pushdown automata, are to be investigated.

Bibliography

1. A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, Boston, 2nd edition, 2006.
2. M. Amos. *DNA Computation*. PhD thesis, University of Warwick, England, 1997.
3. S. Aydin and H. Bordihn. Sequential versus parallel grammar formalisms with respect to measures of descriptional complexity. *Fundamenta Informaticae*, 55(3-4):243–254, 2003.
4. M. Barbaiani, C. Bibire, J. Dassow, A. Delaney, S. Fazekas, M. Ionescu, G. Liu, A. Lodhi, and B. Nagy. The power of programmed grammars with graphs from various classes. *Journal of Applied Mathematics & Computing*, 22(1–2):21–38, 2006.
5. M. Beek and J. Kleijn. Petri net control for grammar systems. In *Formal and natural computing*, pages 220–243, New York, NY, 2002. Springer-Verlag.
6. S. Bensch and H. Bordihn. Active symbols in pure systems. *Fundamenta Informaticae*, 76(3):239–254, 2007.
7. H. Bordihn. A grammatical approach to the LBA problem. In *New Trends in Formal Languages – Control, Cooperation, and Combinatorics*, pages 1–9, London, UK, 1997.
8. H. Bordihn, E. Csuhaj-Varjú, and J. Dassow. CD grammar systems versus L systems. In *Grammatical Models of Multi-Agent Systems*, volume 8 of *Topics in Computer Mathematics*, pages 18–32, Amsterdam, NL, 1999. Gordon and Breach Science Publishers.
9. H. Bordihn and M. Holzer. Programmed grammars and their relation to the LBA problem. *Acta Informatica*, 43(4):223–242, 2006.
10. L. Cojocaru, E. Mäkinen, and F. L. Tiplea. Classes of Szilard languages in NC^1. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 11th International Symposium*, pages 299–306, 2009.
11. E. Csuhaj-Varjú, J. Dassow, J. Kelemen, and G. Păun. *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*. Gordon and Breach, Yverdon, 1994.
12. E. Csuhaj-Varjú, J. Dassow, and G. Păun. Dynamically controlled cooperating/distributed grammar systems. *Information Sciences*, 69(1-2):1–25, 1993.
13. E. Csuhaj-Varjú and G. Vaszil. On context-free parallel communicating grammar systems: synchronization, communication, and normal forms. *Theoretical Computer Science*, 255(1-2):511–538, 2001.
14. E. Csuhaj-Varjú and G. Vaszil. Scattered context grammars generate any recursively enumerable language with two nonterminals. *Information Processing Letters*, 110:902–907, 2010.
15. E. Csuhaj-Varjú, T. Masopust, and G. Vaszil. Blackhole state-controlled regulated pushdown automata. In *Second Workshop on Non-Classical Models for Automata and Applications (NCMA 2010)*, pages 45–56, 2010.

16. E. Czeizler, E. Czeizler, L. Kari, and K. Salomaa. On the descriptional complexity of Watson-Crick automata. *Theoretical Computer Science*, 410(35):3250–3260, 2009.
17. J. Dassow and G. Păun. *Regulated Rewriting in Formal Language Theory.* Springer, New York, 1989.
18. A. Ehrenfeucht, H. C. M. Kleijn, and G. Rozenberg. Adding global forbidding context to context-free grammars. *Theoretical Computer Science*, 37:337–360, 1985.
19. A. Ehrenfeucht and G. Rozenberg. An observation on scattered grammars. *Information Processing Letters*, 9(2):84–85, 1979.
20. E.Mäkinen. A bibliography on Szilard languages. Department of Computer Sciences, University of Tampere. Available on URL: `http://www.cs.uta.fi/reports/pdf/Szilard.pdf`.
21. H. Fernau. Nonterminal complexity of programmed grammars. *Theoretical Computer Science*, 296(2):225–251, 2003.
22. H. Fernau, R. Freund, M. Oswald, and K. Reinhardt. Refining the nonterminal complexity of graph-controlled, programmed, and matrix grammars. *Journal of Automata, Languages and Combinatorics*, 12(1–2):117–138, 2007.
23. H. Fernau and M. Holzer. Graph-controlled cooperating distributed grammar systems with singleton components. *Journal of Automata, Languages and Combinatorics*, 7(4):487–503, 2002.
24. H. Fernau and A. Meduna. On the degree of scattered context-sensitivity. *Theoretical Computer Science*, 290(3):2121–2124, 2003.
25. H. Fernau and A. Meduna. A simultaneous reduction of several measures of descriptional complexity in scattered context grammars. *Information Processing Letters*, 86(5):235–240, 2003.
26. V. Geffert. Normal forms for phrase-structure grammars. *Theoretical Informatics and Applications*, 25(5):473–496, 1991.
27. F. Goldefus. Cooperating distributed grammar systems and graph controlled grammar systems with infinite number of components. In *Proceedings of the 15th Conference STUDENT EEICT 2009 Volume 4*, pages 400–404, Brno, CZ, 2009. Faculty of Information Technology BUT.
28. J. Gonczarowski, H. C. M. Kleijn, and G. Rozenberg. Closure properties of selective substitution grammars (part I). *International Journal of Computer Mathematics*, 14:19–42, 1983.
29. J. Gonczarowski, H. C. M. Kleijn, and G. Rozenberg. Closure properties of selective substitution grammars (part II). *International Journal of Computer Mathematics*, 14:109–135, 1983.
30. J. Gonczarowski, H. C. M. Kleijn, and G. Rozenberg. Grammatical constructions in selective substitution grammars. *Acta Cybernetica*, 6:239–269, 1984.
31. S. A. Greibach and J. E. Hopcroft. Scattered context grammars. *Journal of Computer and System Sciences*, 3(3):233–247, 1969.
32. M. A. Gutiérrez-Naranjo, M. J. Pérez-Jiménez, and A. Riscos-Núñez. Multidimensional Sevilla carpets associated with P systems. In *Proceedings of the ESF Exploratory Workshop on Cellular Computing (Complexity Aspects)*, pages 225–236, 2005.
33. M. Harrison. *Introduction to Formal Language Theory.* Addison-Wesley, Boston, 1978.
34. M. Holzer and M. Kutrib. Nondeterministic finite automata—recent results on the descriptional and computational complexity. In *Implementation and Applications of Automata*, volume 5148 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2008.
35. M. Holzer and M. Kutrib. Descriptional and computational complexity of finite automata—a survey. *Information and Computation*, 209(3):456–470, 2011.
36. J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Boston, 3rd edition, 2006.

37. M. Jantzen, M. Kudlek, and G. Zetzsche. Finite automata controlled by Petri nets. In *Proceedings of the 14th Workshop; Algorithmen und Werkzeuge für Petrinetze*, number Technical Report Nr. 25/2007, pages 57–62. Universität Koblenz-Landau, 2007.
38. O. Jirák. Table-driven parsing of scattered context grammar. In *Proceedings of the 16th Conference and Competition EEICT 2010*, pages 171–175. Faculty of Information Technology, Brno University of Technology, 2010.
39. H. C. M. Kleijn. *Selective Substitution Grammars Based on Context-Free Productions.* PhD thesis, Leiden University, Netherlands, 1983.
40. H. C. M. Kleijn. Basic ideas of selective substitution grammars. In A. Kelemenová and J. Kelemen, editors, *Trends, Techniques, and Problems in Theoretical Computer Science*, volume 281 of *Lecture Notes in Computer Science*, pages 75–95. Springer, 1987.
41. H. C. M. Kleijn and G. Rozenberg. Context-free like restrictions on selective rewriting. *Theoretical Computer Science*, 16:237–269, 1981.
42. H. C. M. Kleijn and G. Rozenberg. A general framework for comparing sequential and parallel rewriting. In *Mathematical Foundations of Computer Science*, pages 360–368, 1981.
43. H. C. M. Kleijn and G. Rozenberg. On the role of selectors in selective substitution grammars. In *Fundamentals of Computation Theory*, volume 117, pages 190–198, 1981.
44. H. C. M. Kleijn and G. Rozenberg. Sequential, continuous and parallel grammars. *Information and Control*, 48(3):221–260, 1981.
45. H. C. M. Kleijn and G. Rozenberg. On the generative power of regular pattern grammars. *Acta Informatica*, 20:391–411, 1983.
46. D. Kolář. Scattered context grammars parsers. In *Proceedings of the 14th International Congress of Cybernetics and Systems of WOCS*, pages 491–500. Wroclaw University of Technology, 2008.
47. D. Kolář and A. Meduna. Regulated pushdown automata. *Acta Cybernetica*, 2000(4):653–664, 2000.
48. D. Kolář and A. Meduna. One-turn regulated pushdown automata and their reduction. *Fundamenta Informaticae*, 2001(21):1001–1007, 2001.
49. D. Kolář and A. Meduna. Regulated automata: From theory towards applications. In *Proceeding of 8th International Conference on Information Systems Implementation and Modelling (ISIM'05)*, pages 33–48, 2005.
50. D. Kolář and Š. Květoňová. Process modeling & optimization of complex systems by scattered context grammars. In *Proceedings of the International Conference on Engineering Computational Technology*, pages 1–12. Valencia, ES, 2010.
51. S. Y. Kuroda. Classes of languages and linear-bounded automata. *Information and Control*, 7(2):207–223, 1964.
52. Z. Křivka, A. Meduna, and P. Zemek. A new normal form for phrase-structure grammars. Manuscript.
53. J. Křoustek, S. Židek, D. Kolář, and A. Meduna. Exploitation of scattered context grammars to model VLIW instruction constraints. In *Proceedings of the 12th Biennial Baltic Electronics Conference*, pages 165–168. Talinn, EE, 2010.
54. M. V. Lomkovskaya. On some properties of c-conditional grammars (in Russian). *Nauchno-Tekhnicheskaya Informatsiya*, 2(1):16–21, 1972.
55. R. Lukáš and A. Meduna. Multigenerative grammar systems. *Schedae Informaticae*, 2006(15):175–188, 2006.
56. R. Lukáš and A. Meduna. Multigenerative grammar systems. In *Proceedings of 1st International Workshop – WFM*, pages 19–26, 2006.
57. R. Lukáš and A. Meduna. General multigenerative grammar systems. In *WFM'07: Information Systems and Formal Models ISIM*, pages 205–212. Silesian University, Opava, 2007.

58. R. Lukáš and A. Meduna. Multigenerative grammar systems and matrix grammars. *Kybernetika*, 46(1):68–82, 2010.
59. M. Madhu. Descriptional complexity of rewriting P systems. *Journal of Automata, Languages and Combinatorics*, 9(2–3):311–316, 2004.
60. E. Mäkinen. A note on pure grammars. *Information Processing Letters*, 23(5):271–274, 1986.
61. J. C. Martin. *Introduction to Languages and the Theory of Computation.* McGraw-Hill, New York, 3rd edition, 2002.
62. C. Martín-Vide, V. Mitrana, and G. Păun, editors. *Formal Languages and Applications*, chapter 13, pages 249–274. Springer, Berlin, 2004.
63. P. Martinek. Limits of pure grammars with monotone productions. *Fundamenta Informaticae*, 33(3):265–280, 1998.
64. T. Masopust. Generalized forbidding grammars with linear productions. In *Third Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2007)*, pages 121–126, Znojmo, CZ, 2007.
65. T. Masopust. Descriptional complexity of multi-parallel grammars. *Information Processing Letters*, 108(2):68–70, 2008.
66. T. Masopust. On the descriptional complexity of scattered context grammars. *Theoretical Computer Science*, 410(1):108–112, 2009.
67. T. Masopust. Bounded number of parallel productions in scattered context grammars with three nonterminals. *Fundamenta Informaticae*, 99(4):473–480, 2010.
68. T. Masopust and A. Meduna. Descriptional complexity of grammars regulated by context conditions. In *LATA '07 Pre-proceedings. Reports of the Research Group on Mathematical Linguistics 35/07, Universitat Rovira i Virgili*, pages 403–411, 2007.
69. T. Masopust and A. Meduna. On descriptional complexity of partially parallel grammars. *Fundamenta Informaticae*, 87(3):407–415, 2008.
70. T. Masopust and A. Meduna. Descriptional complexity of three-nonterminal scattered context grammars: An improvement. In *Proceedings of 11th International Workshop on Descriptional Complexity of Formal Systems*, pages 235–245. Otto-von-Guericke-Universität Magdeburg, 2009.
71. H. A. Maurer, A. Salomaa, and D. Wood. Pure grammars. *Information and Control*, 44(1):47–72, 1980.
72. A. Meduna. Generalized forbidding grammars. *International Journal of Computer Mathematics*, 36(1-2):31–38, 1990.
73. A. Meduna. *Automata and Languages: Theory and Applications.* Springer, London, 2000.
74. A. Meduna. Coincidental extension of scattered context languages. *Acta Informatica*, 39(5):307–314, 2003.
75. A. Meduna. Erratum: Coincidental extension of scattered context languages. *Acta Informatica*, 39(9):699, 2003.
76. A. Meduna. *Elements of Compiler Design.* Auerbach Publications, Boston, 2007.
77. A. Meduna and T. Kopeček. *Conditional Grammars and Their Reduction.* Faculty of Information Technology, Brno University of Technology, 2008.
78. A. Meduna and J. Techet. Canonical scattered context generators of sentences with their parses. *Theoretical Computer Science*, 2007(389):73–81, 2007.
79. A. Meduna and J. Techet. *Scattered Context Grammars and their Applications.* WIT Press, Southampton, 2010.
80. A. Meduna, L. Vrábel, and P. Zemek. LL random context grammars. Submitted to *Kybernetika.*
81. A. Meduna, L. Vrábel, and P. Zemek. LL leftmost k-linear scattered context grammars. In *AIP Conference Proceedings*, volume 1389, pages 833–836, Kassandra, Halkidiki, GR, 2011. American Institute of Physics.

82. A. Meduna, L. Vrábel, and P. Zemek. On nondeterminism in programmed grammars. In *13th International Conference on Automata and Formal Languages*, pages 316–328, Debrecen, HU, 2011. Computer and Automation Research Institute, Hungarian Academy of Sciences.
83. A. Meduna and M. Švec. Descriptional complexity of generalized forbidding grammars. *International Journal of Computer Mathematics*, 80(1):11–17, 2003.
84. A. Meduna and M. Švec. *Grammars with Context Conditions and Their Applications*. Wiley, New Jersey, 2005.
85. A. Meduna and S. Židek. Scattered context grammars generating sentences followed by derivation trees. *Theoretical and Applied Informatics*, 2011(2):97–106, 2011.
86. A. Meduna and P. Zemek. Controlled pure grammar systems. Manuscript.
87. A. Meduna and P. Zemek. Generalized one-sided forbidding grammars. Submitted to *International Journal of Computer Mathematics*.
88. A. Meduna and P. Zemek. Jumping finite automata. Submitted to *International Journal of Foundations of Computer Science*.
89. A. Meduna and P. Zemek. Left random context ET0L grammars. Submitted to *Fundamenta Informaticae*.
90. A. Meduna and P. Zemek. Nonterminal complexity of left random context E0L grammars. Submitted to *Information Processing Letters*.
91. A. Meduna and P. Zemek. Nonterminal complexity of one-sided random context grammars. Submitted to *Acta Informatica*.
92. A. Meduna and P. Zemek. On one-sided forbidding grammars and selective substitution grammars. *International Journal of Computer Mathematics*. In press.
93. A. Meduna and P. Zemek. On the generation of sentences with their parses by propagating regular-controlled grammars. Submitted to *Theoretical Computer Science*.
94. A. Meduna and P. Zemek. One-sided random context grammars with leftmost derivations. Submitted to *LNCS Festschrifts Series "Languages Alive – Essays Dedicated to Jürgen Dassow on the Occasion of his 65th Birthday"*.
95. A. Meduna and P. Zemek. *Regulated Grammars and Their Transformations*. Brno University of Technology, Brno, CZ, 2010.
96. A. Meduna and P. Zemek. One-sided random context grammars. *Acta Informatica*, 48(3):149–163, 2011.
97. A. Meduna and P. Zemek. Workspace theorems for regular-controlled grammars. *Theoretical Computer Science*, 412(35):4604–4612, 2011.
98. F. Okubo. A note on the descriptional complexity of semi-conditional grammars. *Information Processing Letters*, 110(1):36–40, 2009.
99. M. Penttonen. One-sided and two-sided context in formal grammars. *Information and Control*, 25(4):371–392, 1974.
100. M. Penttonen. ET0L-grammars and N-grammars. *Information Processing Letters*, 4(1):11–13, 1975.
101. G. Păun. On the synchronization in parallel communicating grammar systems. *Acta Informatica*, 30(4):351–367, 1993.
102. G. Rozenberg. Selective substitution grammars (towards a framework for rewriting systems). Part 1: Definitions and examples. *Elektronische Informationsverarbeitung und Kybernetik*, 13(9):455–463, 1977.
103. G. Rozenberg. On coordinated selective substitutions: Towards a unified theory of grammars and machines. *Theoretical Computer Science*, 37:31–50, 1985.
104. G. Rozenberg and A. Salomaa. *Mathematical Theory of L Systems*. Academic Press, Orlando, 1980.
105. G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages, Vol. 1: Word, Language, Grammar*. Springer, New York, 1997.

106. G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages, Vol. 2: Linear Modeling: Background and Application.* Springer, New York, 1997.
107. G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages, Volumes 1 through 3.* Springer, New York, 1997.
108. W. Rytter and M. Crochemore. *Text Algorithms.* Oxford University Press, New York, 1994.
109. A. Salomaa. *Formal Languages.* Academic Press, London, 1973.
110. J. Seberry and J. Pieprzyk. *Cryptography: An Introduction to Computer Security.* Prentice-Hall, New Jersey, 1989.
111. M. Sipser. *Introduction to the Theory of Computation.* PWS Publishing Company, Boston, 2nd edition, 2006.
112. R. Siromoney and V. R. Dare. On infinite words obtained by selective substitution grammars. *Theoretical Computer Science*, 39:281–295, 1985.
113. R. Siromoney and K. G. Subramanian. Selective substitution array grammars. *Information Sciences*, 25(1):73–83, 1981.
114. A. P. J. van der Walt and S. Ewert. A shrinking lemma for random forbidding context languages. *Theoretical Computer Science*, 237(1–2):149–158, 2000.
115. G. Vaszil. On the descriptional complexity of some rewriting mechanisms regulated by context conditions. *Theoretical Computer Science*, 330(2):361–373, 2005.
116. J. von zur Gathen and J. Gerhard. *Modern Computer Algebra.* Cambridge University Press, New York, 2nd edition, 2003.
117. D. Wood. *Grammars and L Forms: An Introduction.* Springer, New York, 1980.
118. D. Wood. *Theory of Computation: A Primer.* Addison-Wesley, Boston, 1987.
119. P. Zemek. Normal forms of one-sided random context grammars. Submitted to *EEICT 2012.*
120. P. Zemek. On the nonterminal complexity of left random context E0L grammars. In *Proceedings of the 17th Conference STUDENT EEICT 2011 Volume 3*, pages 510–514, Brno, CZ, 2011. Faculty of Information Technology BUT.
121. G. Zetzsche. Erasing in Petri net languages and matrix grammars. In *DLT '09: Proceedings of the 13th International Conference on Developments in Language Theory*, pages 490–501. Springer, 2009.
122. G. Zetzsche. On erasing productions in random context grammars. In *ICALP'10: Proceedings of the 37th International Colloquium on Automata, Languages and Programming*, pages 175–186. Springer, 2010.
123. G. Zetzsche. Toward understanding the generative capacity of erasing rules in matrix grammars. *International Journal of Computer Mathematics*, 22(2):411–426, 2011.