# The Degree of Word-Expansion of Lexicalized RRWW-Automata – A New Measure for The Degree of Nondeterminism of (Context-Free) Languages*

**František Mráz**†
Faculty of Mathematics and Physics,
Department of Computer Science
Charles University, 118 00 Praha 1, Czech Republic
E-mail: mraz@ksvi.ms.mff.cuni.cz

**Friedrich Otto**
Fachbereich Elektrotechnik/Informatik
Universität Kassel, 34109 Kassel, Germany
E-mail: otto@theory.informatik.uni-kassel.de

**Martin Plátek**†
Faculty of Mathematics and Physics,
Department of Computer Science,
Charles University, 118 00 Praha 1, Czech Republic
E-mail: Martin.Platek@mff.cuni.cz

November 1, 2007

**Abstract**

Restarting automata can be seen as analytical variants of classical automata as well as of regulated rewriting systems. We study a measure for the degree of nondeterminism of (context-free) languages in terms of deterministic restarting automata that are (strongly) lexicalized. This measure is based on the number of auxiliary symbols (categories)

*Some of the results have been announced at CIAA 2007, Prague, July 2007.

†F. Mráz and M. Plátek were supported by the program 'Information Society' under project 1ET100300517. F. Mráz was also supported by the Grant Agency of Charles University in Prague under Grant-No. 358/2006/A-INF/MFF.

used for recognizing a language as the projection of its characteristic language onto its input alphabet. This type of recognition is typical for analysis by reduction, a method used in linguistics for the creation and verification of formal descriptions of natural languages. Our main results establish a hierarchy of classes of context-free languages and two hierarchies of classes of non-context-free languages that are based on the expansion factor of a language.

# 1 Introduction

Automata with a restart operation were originally introduced in order to describe a method of grammar-checking for the Czech language (see, e.g., [9]). These automata, which work in a fashion similar to the automata used in this paper, started the investigation of restarting automata as a suitable tool for modeling the so-called *analysis by reduction*. Analysis by reduction facilitates the development and testing of categories for syntactic and semantic disambiguation of sentences of natural languages. It is often used (implicitly) for developing grammars for natural languages based on the notion of *dependency* [10]. In particular, the Functional Generative Description (FGD) for the Czech language developed in Prague (see, e.g., [11]) is based on this method.

Analysis by reduction consists in stepwise simplifications (reductions) of a given extended sentence (enriched by syntactical and semantical categories) until a correct simple sentence is obtained. Each simplification replaces a small part of the sentence by an even shorter phrase. Here we formalize analysis by reduction by using deterministic restarting automata for characteristic languages, that is, these automata work on languages that include auxiliary symbols (categories) in addition to the input symbols. By requiring that the automata considered are *lexicalized* we restrict the lengths of the blocks of auxiliary symbols that are allowed on the tape by a constant. This restriction is quite natural from a linguistic point of view, as these blocks of auxiliary symbols model the meta-language categories from all linguistic layers with which an input string is being enriched when its disambiguated form is being produced (see, e.g., [11]). We use deterministic restarting automata in order to ensure the *correctness preserving property* for the analysis.

While it is well-known that monotone deterministic restarting automata without auxiliary symbols recognize exactly the deterministic context-free languages [7], we will see that exactly the context-free languages are recognized as proper languages of lexicalized (deterministic) restarting automata that are monotone. Then we define the *word-expansion factor* of a restarting automaton $M$. This is the maximal number of auxiliary symbols that $M$ uses simultaneously on its tape when processing a word from its characteristic language $L_{\mathrm{C}}(M)$. If $L$ is a (context-free) language, then the minimal

word-expansion factor for any lexicalized (deterministic) restarting automaton $M$ with proper language $L$ can be seen as a measure for the degree of nondeterminism of $L$. This is quite natural from a language-theoretic point of view, as the auxiliary symbols inserted in an input sentence can be interpreted as information that is used to single out a particular computation of an otherwise nondeterministic restarting automaton. Corresponding notions have been investigated before for finite-state automata, and for some other devices [1, 4, 5]. An overview about degrees of nondeterminism for pushdown automata can be found in [14].

For the monotone case we will see that strongly lexicalized RRWW-automata and lexicalized RRWW-automata have exactly the same expressive power. Accordingly, we establish three hierarchies of language classes that are based on the word-expansion factor: one for monotone deterministic RRWW-automata that are (strongly) lexicalized, and two for the nonmonotone case. Observe that due to our result above the hierarchy for the monotone case is a hierarchy of context-free languages above the level of deterministic context-free languages.

The paper is structured as follows. In Section 2 we define the deterministic RRWW-automaton, which is the model of restarting automata we will use, and restate some basic results on these automata. In particular, we prove that the class of proper languages of deterministic RRWW-automata is almost universal. Then in Section 3 we introduce our measure of nondeterminism and derive the announced results. In the concluding section (Section 4) we summarize our results, describe some related decidability and undecidability results, and present some open problems for future work.

## 2 Proper Languages of Restarting Automata

Here we describe in short the type of restarting automaton we will be dealing with. More details on restarting automata in general can be found in [12]. In what follows, $\lambda$ denotes the empty word, and $\mathbb{N}_+$ and $\mathbb{N}$ denote the set of positive and the set of nonnegative integers, respectively.

A *one-way deterministic restarting automaton* (det-RRWW-automaton for short) is a deterministic machine $M = (Q, \Sigma, \Gamma, ¢, \$, q_0, k, \delta)$. It consists of a finite-state control, a flexible tape, and a read/write window of a fixed size $k \geq 1$. Here $Q$ denotes a finite set of (internal) states that contains the initial state $q_0$, $\Sigma$ is a finite input alphabet, and $\Gamma$ is a finite tape alphabet that contains $\Sigma$. The elements of $\Gamma \setminus \Sigma$ are called *auxiliary symbols*. The additional symbols $¢, \$ \notin \Gamma$ are used as markers for the left and the right end of the workspace, respectively. They cannot be removed from the tape. The behaviour of $M$ is described by a transition function $\delta$ that associates transition steps to pairs $(q, u)$ consisting of a state $q$ and a possible content $u$ of the read/write window. There are four types of transition steps: *move-*

*right steps*, *rewrite steps*, *restart steps*, and *accept steps*. A *move-right step* simply shifts the read/write window one position to the right and changes the internal state. A *rewrite step* replaces the content of the read/write window by a shorter word, in this way shortening the tape, shifts the read/write window across the newly written factor, and changes the internal state. A *restart step* causes $M$ to place its read/write window over the left end of the tape, so that the first symbol it sees is the left sentinel ¢, and to reenter the initial state $q_0$. Finally, an *accept step* simply causes $M$ to halt and accept. It is required that, when ignoring move-right operations, then in any computation of $M$, rewrite steps and restart steps alternate, with a rewrite step coming first. However, it is more convenient to describe $M$ by a finite set of so-called *meta-instructions* (see below).

A *configuration* of $M$ is described by a string $\alpha q \beta$, where $q \in Q$, and either $\alpha = \lambda$ and $\beta \in \{¢\} \cdot \Gamma^* \cdot \{\$\}$ or $\alpha \in \{¢\} \cdot \Gamma^*$ and $\beta \in \Gamma^* \cdot \{\$\}$; here $q$ represents the current state, $\alpha\beta$ is the current content of the tape, and it is understood that the head scans the first $k$ symbols of $\beta$ or all of $\beta$ when $|\beta| \leq k$. A *restarting configuration* is of the form $q_0 ¢ w \$$, where $w \in \Gamma^*$.

A *rewriting meta-instruction* for $M$ has the form $(E_1, u \to v, E_2)$, where $E_1$ and $E_2$ are regular languages (often given in terms of regular expressions), and $u, v \in \Gamma^*$ are words satisfying the restrictions $k \geq |u| > |v|$. Starting from the restarting configuration $q_0 ¢ w \$$, $M$ can execute this meta-instruction only if $w$ admits a factorization of the form $w = w_1 u w_2$ such that $¢ w_1 \in E_1$ and $w_2 \$ \in E_2$. In this case the leftmost of these factorizations is chosen, and $q_0 ¢ w \$$ is transformed into $q_0 ¢ w_1 v w_2 \$$. This computation is called a *cycle* of $M$. It is expressed as $w \vdash_M^c w_1 v w_2$. In order to describe the tail of an accepting computation, that is, that part that follows after the last execution of a restart step, we use *accepting meta-instructions* of the form $(E_1, \mathrm{Accept})$, where the strings from the regular language $E_1$ are accepted by $M$ after scanning them from left to right.

The meta-instructions used to describe restarting automata can be interpreted in a nondeterministic way. For example, the word $w$ on the tape may simultaneously admit factorizations that correspond to different meta-instructions. Therefore we will always suppose that our restarting automata are defined by explicit transition functions, which can be obtained from the given descriptions by meta-instructions.

A word $w \in \Gamma^*$ is *accepted* by $M$, if there is a computation which, starting from the restarting configuration $q_0 ¢ w \$$, consists of a finite sequence of cycles that is followed by an application of an accepting meta-instruction. By $L_{\mathrm{C}}(M)$ we denote the language consisting of all words accepted by $M$. It is the *characteristic language* of $M$.

By $\mathsf{Pr}^\Sigma$ we denote the projection from $\Gamma^*$ onto $\Sigma^*$, that is, $\mathsf{Pr}^\Sigma$ is the morphism defined by $a \mapsto a$ $(a \in \Sigma)$ and $A \mapsto \lambda$ $(A \in \Gamma \setminus \Sigma)$. If $v := \mathsf{Pr}^\Sigma(w)$, then $v$ is the $\Sigma$-*projection* of $w$, and $w$ is an *expanded version* of $v$. For a language $L \subseteq \Gamma^*$, $\mathsf{Pr}^\Sigma(L) := \{\, \mathsf{Pr}^\Sigma(w) \mid w \in L \,\}$.

In recent papers (see, e.g., [12]) restarting automata were mainly used as acceptors. The (*input*) *language* accepted by a restarting automaton $M$ is the set $L(M) := L_{\mathrm{C}}(M) \cap \Sigma^*$, that is, it is the set of input words $w \in \Sigma^*$ for which there exists an accepting computation starting from the configuration $q_0 ¢ w \$$. Here, motivated by linguistic considerations to model the processing of sentences that are enriched by syntactic and semantic categories, we are rather interested in the so-called *proper language of* $M$, which is the set of words $L_{\mathrm{P}}(M) := \mathsf{Pr}^{\Sigma}(L_{\mathrm{C}}(M))$. Hence, a word $v \in \Sigma^*$ belongs to $L_{\mathrm{P}}(M)$ if and only if there exists an expanded version $u$ of $v$ such that $u \in L_{\mathrm{C}}(M)$.

We are also interested in some restrictions on rewrite-instructions (expressed by the second part of the class name): -WW denotes no restriction, -W means that no auxiliary symbols are available (that is, $\Gamma = \Sigma$), and -$\lambda$ means that no auxiliary symbols are available and that each rewrite step is simply a deletion (that is, if $u \to v$ is a rewrite instruction of $M$, then $v$ is obtained from $u$ by deleting some symbols).

For each type X of restarting automata, we use $\mathcal{L}_{\mathrm{C}}(\mathsf{X})$ to denote the class of all characteristic languages of automata of this type. Analogously, $\mathcal{L}(\mathsf{X})$ and $\mathcal{L}_{\mathrm{P}}(\mathsf{X})$ denote the class of all input languages and the class of all proper languages of these automata.

The following property is of central importance (see, e.g., [7]).

**Definition 2.1 (Correctness Preserving Property.)** *An* RRWW*-automaton $M$ is* correctness preserving *if $u \in L_{\mathrm{C}}(M)$ and $u \vdash_M^{c^*} v$ imply that $v \in L_{\mathrm{C}}(M)$.*

It is easily seen that each deterministic RRWW-automaton is correctness preserving. In proofs we will repeatedly use the following simple generalization of a fact given in [12].

**Proposition 2.2** *For any* RRWW*-automaton $M$, there exists a constant $p$ such that the following property holds. Assume that $uvw \vdash_M^c uv'w$ is a cycle of $M$, where $u = u_1 u_2 \cdots u_n$ for some non-empty words $u_1, \ldots, u_n$ and a constant $n > p$. Then there exist $r, s \in \mathbb{N}_+$, $1 \leq r < s \leq n$, such that*

$$u_1 \cdots u_{r-1}(u_r \cdots u_{s-1})^i u_s \cdots u_n vw \vdash_M^c u_1 \cdots u_{r-1}(u_r \cdots u_{s-1})^i u_s \cdots u_n v'w$$

*holds for all $i \geq 0$, that is, $u_r \cdots u_{s-1}$ is a 'pumping factor' in the above cycle. Similarly, such a pumping factor can be found in any factorization of length greater than $p$ of $w$. Such a pumping factor can also be found in any factorization of length greater than $p$ of a word accepted in a tail computation.*

As deterministic restarting automata only accept Church-Rosser languages (see, e.g., [12]), we have the following complexity result.

**Proposition 2.3** *If $M$ is a deterministic* RRWW*-automaton, then the membership problem for the language $L_{\mathrm{C}}(M)$ is solvable in linear time.*

In contrast to this result we will now show that the class $\mathcal{L}_{\mathrm{P}}(\mathsf{det\text{-}RRWW})$ of proper languages of deterministic RRWW-automata is 'almost' universal.

In [13] it is shown that the class CRL of Church-Rosser languages is a *basis* for the class RE of recursively enumerable languages, that is, for each recursively enumerable language $L \subseteq \Sigma^*$, there exists a Church-Rosser language $B$ on some alphabet $\Delta$ strictly containing $\Sigma$ such that $\mathsf{Pr}^{\Sigma}(B) = L$. As CRL coincides with the class of input languages of deterministic RRWW-automata (see, e.g., [12]), there exists a deterministic RRWW-automaton $M'$ with input alphabet $\Delta$ and tape alphabet $\Gamma$ such that $L(M') = B$. Hence, $L = \mathsf{Pr}^{\Sigma}(B) \subseteq \mathsf{Pr}^{\Sigma}(L_{\mathrm{C}}(M'))$. However, the language $L_{\mathrm{C}}(M')$ will in general also contain words for which the projection onto $\Sigma$ does not belong to the language $L$, that is, the above inclusion is in general a strict one. Accordingly, in order to derive the intended universality result we need a somewhat more sophisticated construction.

In the following considerations we will restrict our attention to recursively enumerable languages over the fixed two-letter alphabet $\Sigma_0 := \{a, b\}$. Let $\Sigma_1 := \Sigma_0 \cup \{c\}$, let $\varphi_0 : \Sigma_0^* \to \Sigma_0^*$ be the injective morphism that is defined by $a \mapsto aa$ and $b \mapsto bb$, and let $\varphi : \Sigma_0^* \to \Sigma_1^*$ denote the mapping that is defined by $\varphi(w) := \varphi_0(w) \cdot c$. Then $\varphi$ is an encoding that can be computed by a rational transducer. The following result expresses the universality of $\mathcal{L}_{\mathrm{P}}(\mathsf{det\text{-}RRWW})$ announced above.

**Proposition 2.4** *For each recursively enumerable language $L \subseteq \Sigma_0^+$, there exists a* det-RRWW*-automaton $M$ such that $L_{\mathrm{P}}(M) \cap \Sigma_0^* \cdot c = \varphi(L)$.*

**Proof.** Let $L \subseteq \Sigma_0^+$ be a recursively enumerable language. In the proof of Theorem 7.1 of [13], a Church-Rosser language of the form $B := \{\, wde^{m(w)} \mid w \in L \,\}$ is constructed from a Turing machine accepting the language $L$, where $m(w)$ is a unique integer associated with the word $w$.

Let $\Sigma_0' := \{a', b'\}$, $\Delta' := \Sigma_0' \cup \{d, e\}$, and let $\varphi' : \Sigma_0^* \to \{a', b'\}^*$ be the morphism induced by $a \mapsto a'$ and $b \mapsto b'$. By $B'$ we denote the Church-Rosser language $B' := \{\, \varphi'(w)de^{m(w)} \mid w \in L \,\}$. As noted above there exists a deterministic RRWW-automaton $M' = (Q', \Delta', \Gamma', ¢, \$, q_0', k', \delta')$ satisfying $L(M') = B'$.

From $M'$ we construct a deterministic RRWW-automaton $M = (Q, \Sigma_1, \Gamma, ¢, \$, q_0, k, \delta)$ satisfying $L_{\mathrm{P}}(M) \cap \Sigma_0^* \cdot c = \varphi(L)$ as follows. As tape alphabet $\Gamma$ we take $\Gamma := \Gamma' \cup \Sigma_1$, where we assume that $\Gamma' \cap \Sigma_1 = \emptyset$. Given an input of the form $xcde^m$, $x \in \Sigma_0^*$ and $m \geq 0$, the automaton $M$ works in two phases:

- In the first phase $M$ checks that the prefix $xc$ is a word of the form $\varphi(w)$ for some $w \in \Sigma_0^*$. In the negative it halts and rejects, while in the affirmative it replaces $\varphi(w)$ by the word $\varphi'(w)$ by deleting the suffix

$c$ in the first cycle, and by rewriting the rightmost factor $aa$ or $bb$ by $a'$ or $b'$, respectively, in the following cycles. In this way the input $xcde^m = \varphi(w)de^m$ is transformed into the word $\varphi'(w)de^m$. $M$ detects that this phase is complete when the tape content starts with a prefix of the form $\mathfrak{c}a'$ or $\mathfrak{c}b'$.

- Now $M$ simulates the automaton $M'$ step by step.

It follows that $M$ is a deterministic RRWW-automaton. From the above description it is easily seen that a word $z \in \Gamma^*$ belongs to the characteristic language $L_{\mathrm{C}}(M)$ if and only if it belongs to the characteristic language $L_{\mathrm{C}}(M')$, or $z = \varphi(w)de^m$ for some word $w \in L$ and the corresponding integer $m$, or $z = \varphi_0(w_1)\varphi'(w_2)de^m$, where $w = w_1w_2 \in L$ and $m$ is the corresponding integer. Thus, as the words in $\Gamma'^*$ do not contain any occurrences of the input letters $\Sigma_1$, we see that the proper language $L_{\mathrm{P}}(M)$ is the set

$$\mathsf{Pr}^{\Sigma_1}(L_{\mathrm{C}}(M)) = \{\, \varphi(w) \mid w \in L \,\} \cup \{\, \varphi_0(w_1) \mid \exists w_2 \in \Sigma_0^* : w_1w_2 \in L \,\}.$$

It follows that $L_{\mathrm{P}}(M) \cap \Sigma_0^* \cdot c = \varphi(L)$, which proves our claim. □

Thus, a word $w \in \Sigma_0^*$ belongs to the recursively enumerable language $L$ if and only if its image $\varphi(w)$ belongs to the proper language $L_{\mathrm{P}}(M)$. This yields the following consequence.

**Corollary 2.5** *There exists a* det-RRWW*-automaton $M$ such that the language $L_{\mathrm{P}}(M)$ is non-recursive.*

# 3 Lexicalized RRWW-Automata

From Proposition 2.4 and its corollary we see that proper languages of deterministic RRWW-automata are in general far more complex than the corresponding input and characteristic languages. Therefore we restrict our attention in the following to deterministic RRWW-automata for which the use of auxiliary symbols is somehow restricted.

**Definition 3.1** *Let $M = (Q, \Sigma, \Gamma, \mathfrak{c}, \$, q_0, k, \delta)$ be a deterministic* RRWW*-automaton.*

(a) *A word $w \in \Gamma^*$ is* not immediately rejected by *$M$ if there exists a meta-instruction of $M$ that is applicable to the restarting configuration $q_0\mathfrak{c}w\$$, that is, $M$ can either perform a cycle of the form $w \vdash_M^c z$ for some word $z \in \Gamma^*$, or $M$ accepts $w$ in a tail computation. By $\mathsf{NIR}(M)$ we denote the set of all words that are not immediately rejected by $M$.*

(b) *The deterministic* RRWW*-automaton $M$ is called* lexicalized *if there exists a constant $j \in \mathbb{N}_+$ such that, whenever $v \in (\Gamma \setminus \Sigma)^*$ is a factor of a word $w \in \mathsf{NIR}(M)$, then $|v| \leq j$.*

(c) *$M$ is called* strongly lexicalized *if it is lexicalized, and if each of its rewrite steps only deletes symbols.*

Strong lexicalization is a technique that is used in dependency (or categorially) based formal descriptions of natural languages [11].

If $M$ is a lexicalized RRWW-automaton, and if $w \in \Gamma^*$ is an extended version of an input word $v = \mathsf{Pr}^{\Sigma}(w)$ such that $w$ is not immediately rejected by $M$, then $|w| \leq (j+1) \cdot |v| + j$ for some constant $j > 0$. Thus, $L_{\mathrm{P}}(M)$ is context-sensitive, contrasting Proposition 2.4. Actually we have the following stronger result.

**Proposition 3.2** *If $M$ is a lexicalized* RRWW*-automaton, then the proper language $L_{\mathrm{P}}(M)$ is growing context-sensitive.*

**Proof.** Let $M$ be a deterministic RRWW-automaton with input alphabet $\Sigma$ and tape alphabet $\Gamma$, and assume that $M$ is lexicalized with constant $j \in \mathbb{N}$. Then no word $w \in L_{\mathrm{C}}(M)$ contains any factor from $(\Gamma \smallsetminus \Sigma)^*$ of length exceeding $j$. Thus, the morphism $\mathsf{Pr}^{\Sigma} : \Gamma^* \to \Sigma^*$ has *$j$-limited erasing* (see, e.g, [6]) on $L_{\mathrm{C}}(M)$. As $M$ is a deterministic RRWW-automaton, $L_{\mathrm{C}}(M)$ is a Church-Rosser language, which implies that it belongs to the class GCSL of *growing context-sensitive languages* [3]. This in turn implies that $L_{\mathrm{P}}(M) = \mathsf{Pr}^{\Sigma}(L_{\mathrm{C}}(M))$ is also growing context-sensitive, as this class is closed under limited erasing [2]. □

Observe, however, that not every growing context-sensitive language is the proper language of a lexicalized RRWW-automaton.

**Proposition 3.3** *The Church-Rosser language $L_{\mathrm{e}} := \{\, a^{2^n} \mid n \in \mathbb{N} \,\}$ is not contained in $\mathcal{L}_{\mathrm{P}}$*(lex-RRWW)*.*

**Proof.** Assume that $L_{\mathrm{e}} = L_{\mathrm{P}}(M)$ for a lexicalized RRWW-automaton $M = (Q, \{a\}, \Gamma, ¢, \$, q_0, k, \delta)$, and let $z := a^{2^n} \in L_{\mathrm{e}}$, where $n$ is a large integer. Then there exists an extended version $w \in \Gamma^*$ of $z$ such that $w \in L_{\mathrm{C}}(M)$. Thus, the computation of $M$ with input $w$ is accepting. Based on the pumping lemma (Prop. 2.2) it is easily seen that this computation cannot just consist of an accepting tail computation, that is, it begins with a cycle of the form $w \vdash_M^c w'$. From the correctness preserving property it follows that $w' \in L_{\mathrm{C}}(M)$, which in turn implies that $\mathsf{Pr}^{\{a\}}(w') \in L_{\mathrm{e}}$. Thus, $\mathsf{Pr}^{\{a\}}(w') = a^m$ for some integer $m$ satisfying $2^n - k \leq m < 2^n + k$. From the choice of $z$ it follows that $m = 2^n$, that is, $w'$ is obtained from $w$ by rewriting some auxiliary symbols only. We can repeat this argument until eventually $M$ either rewrites some occurrences of the symbol $a$, which will then yield a word $\hat{w} \in L_{\mathrm{C}}(M)$ for which the projection $\mathsf{Pr}^{\{a\}}(\hat{w})$ does not belong to the language $L_{\mathrm{e}}$ anymore, or until $M$ accepts a word $\tilde{w}$ in a tail computation for which $\mathsf{Pr}^{\{a\}}(\tilde{w}) = a^{2^n}$ holds. In the latter case the pumping lemma can

be applied to show that $L_{\mathrm{P}}(M)$ contains words that do not belong to the language $L_{\mathrm{e}}$. In either case it follows that $L_{\mathrm{e}}$ is not the proper language of any lexicalized RRWW-automaton. □

In what follows we are only interested in lexicalized RRWW-automata and their proper languages. By lex-RRWW we denote the class of these automata, and by str-RRWW we denote the class of strongly lexicalized RRWW-automata. Recall from the definition that lexicalized RRWW-automata are deterministic. Further, we are interested in RRWW-automata that are *monotone*.

Each computation of an RRWW-automaton $M$ can be described by a sequence of cycles $C_1, C_2 \ldots, C_n$, where $C_n$ is the last cycle, which is followed by the tail of the computation. Each cycle $C_i$ of this computation contains a unique configuration of the form $\mathrm{¢}xquy\$$ in which a rewrite step is executed. By $D_r(C_i)$ we denote the *right distance* $|y\$|$ of this cycle. The sequence of cycles $C_1, C_2, \ldots, C_n$ is called *monotone* if $D_r(C_1) \geq D_r(C_2) \geq \cdots \geq D_r(C_n)$ holds. A computation of $M$ is called *monotone* if the corresponding sequence of cycles is monotone. Observe that the tail of the computation is not taken into account here. Finally, an RRWW-automaton is called *monotone* if each of its computations is monotone. We use the prefix mon- to denote this property. Concerning the expressive power of lexicalized RRWW-automata that are monotone we have the following result.

**Theorem 3.4** *The class* CFL *of context-free languages coincides with the class of proper languages of monotone* RRWW*-automata that are* (*strongly*) *lexicalized, that is,*

$$\mathsf{CFL} = \mathcal{L}_{\mathrm{P}}(\mathsf{lex\text{-}mon\text{-}RRWW}) = \mathcal{L}_{\mathrm{P}}(\mathsf{str\text{-}mon\text{-}RRWW}).$$

**Proof.** If $M$ is a monotone RRWW-automaton, then its characteristic language $L_{\mathrm{C}}(M)$ is context-free [7]. As $L_{\mathrm{P}}(M) = \mathsf{Pr}^{\Sigma}(L_{\mathrm{C}}(M))$, and as CFL is closed under morphisms, it follows that $L_{\mathrm{P}}(M)$ is context-free.

Conversely, assume that $L \subseteq \Sigma^*$ is a context-free language. Without loss of generality we may assume that $L$ does not contain the empty word. Thus, there exists a context-free grammar $G = (N, \Sigma, S, P)$ for $L$ that is in *Greibach normal form*, that is, each rule of $P$ has the form $A \to \alpha$ for some string $\alpha \in \Sigma \cdot N^*$ (see, e.g., [6]). For the following construction we assume that the rules of $G$ are numbered from 1 to $m$.

From $G$ we construct a new grammar $G' := (N, \Sigma \cup B, S, P')$, where $B := \{ \nabla_i \mid 1 \leq i \leq m \}$ is a set of new terminal symbols that are in one-to-one correspondence to the rules of $G$, and

$$P' := \{ A \to \nabla_i \alpha \mid (A \to \alpha) \text{ is the } i\text{-th rule of } G,\ 1 \leq i \leq m \}.$$

Obviously, a word $\omega \in (\Sigma \cup B)^*$ belongs to $L(G')$ if and only if $\omega$ has the form $\omega = \nabla_{i_1} a_1 \nabla_{i_2} a_2 \cdots \nabla_{i_n} a_n$ for some integer $n > 0$, where $a_1, \ldots, a_n \in \Sigma$,

$i_1, \ldots, i_n \in \{1, \ldots, m\}$, and these indices describe a (left-most) derivation of $w := a_1 a_2 \cdots a_n$ from $S$ in $G$. Thus, $\mathsf{Pr}^{\Sigma}(L(G')) = L(G) = L$. From $\omega$ this derivation can be reconstructed deterministically. In fact, the language $L(G')$ is deterministic context-free. Hence, there exists a monotone deterministic RR-automaton $M$ for this language [7]. By interpreting the symbols of $B$ as auxiliary symbols, we obtain a monotone deterministic RRWW-automaton $M'$ such that $\mathsf{Pr}^{\Sigma}(L_{\mathrm{C}}(M')) = \mathsf{Pr}^{\Sigma}(L(M)) = \mathsf{Pr}^{\Sigma}(L(G')) = L$.

It remains to verify that $M'$ is lexicalized. From the observation above we see that within each word $\omega \in L(G')$, symbols from $B$ and terminal symbols from $\Sigma$ occur alternatingly. As the RR-automaton $M$ is correctness preserving, each restarting configuration of $M$ within an accepting computation is of the form $q_0 \mathbb{c} \gamma \$$ for some $\gamma \in L(G')$. Thus, it only contains factors from $B^+$ of length one. It follows that $M'$ is lexicalized with constant 1. As $M$ is an RR-automaton, all rewrite operations of $M'$ are deletions. Hence, $M'$ is in fact strongly lexicalized. □

Together with Propositions 3.2 and 3.3 this yields the following consequence, as CRL is incomparable to CFL under inclusion [3].

**Corollary 3.5** *$\mathcal{L}_{\mathrm{P}}$(lex-RRWW) is a proper subclass of* GCSL *that is incomparable under inclusion to the class* CRL *of Church-Rosser languages.*

Next we introduce a static complexity measure for lexicalized RRWW-automata.

**Definition 3.6** *Let $M = (Q, \Sigma, \Gamma, \mathbb{c}, \$, q_0, k, \delta)$ be an* RRWW*-automaton, and let $m \in \mathbb{N}$. The automaton $M$ is said to have* word-expansion *$m$, denoted by $\mathsf{W}(M) = m$, if each word from the set $\mathsf{NIR}(M)$ contains at most $m$ occurrences of auxiliary symbols, that is, if $w \in \mathsf{NIR}(M)$, then $|\mathsf{Pr}^{\Gamma \setminus \Sigma}(w)| \leq m$.*

By W($m$)-RRWW we denote the class of lexicalized RRWW-automata with word-expansion of degree $m$, and the strongly lexicalized variant of this class is denoted by the additional prefix str-.

**Theorem 3.7** *For all $m \in \mathbb{N}$, if $M$ is a* W($m$)-RRWW*-automaton, then the membership problem for the language $L_{\mathrm{P}}(M)$ is solvable deterministically in time $O(n^{m+1})$.*

**Proof.** Let $m \in \mathbb{N}$, and assume that $M = (Q, \Sigma, \Gamma, \mathbb{c}, \$, q_0, k, \delta)$ is a lexicalized RRWW-automaton with word-expansion of degree $m$. Then a word $w \in \Sigma^*$ belongs to the language $L_{\mathrm{P}}(M)$ if and only if there exists an expansion $u \in \Gamma^*$ of $w$ such that $u \in L_{\mathrm{C}}(M)$. Thus, $u$ is obtained from $w$ by inserting at most $m$ auxiliary letters. There are $j := |\Gamma \setminus \Sigma|$ many such symbols available to $M$, and there are $\binom{|w|+m}{m}$ options to place $m$ symbols

within the expanded version of $w$ of length $|w|+m$. Hence, there are at most $\binom{|w|+m}{m} \cdot (j+1)^m$ many words of the form required for $u$. Accordingly, these words can be enumerated in a systematic way, and for each of them it can be checked in linear time whether or not it belongs to $L_{\mathrm{C}}(M)$ (Proposition 2.3). This yields the time bound $O((n+m)^m \cdot (j+1)^m \cdot (n+m)) = O(n^{m+1})$, as $m$ and $j$ are fixed. □

Here we are interested in the classes $\mathcal{L}_{\mathrm{P}}(\text{(str-)W}(m)\text{(-mon)-RRWW})$. As monotone deterministic RR-automata accept the deterministic context-free languages [7], we have the following result.

**Proposition 3.8**
$\mathsf{DCFL} = \mathcal{L}_{\mathrm{P}}(\text{str-W}(0)\text{-mon-RRWW}) = \mathcal{L}_{\mathrm{P}}(\text{W}(0)\text{-mon-RRWW})$.

Actually, the correspondence between monotone strongly lexicalized RRWW-automata and monotone lexicalized RRWW-automata carries over to all finite degrees of word-expansion.

**Proposition 3.9**
*For all* $m \in \mathbb{N}$, $\mathcal{L}_{\mathrm{P}}(\text{str-W}(m)\text{-mon-RRWW}) = \mathcal{L}_{\mathrm{P}}(\text{W}(m)\text{-mon-RRWW})$.

**Proof.** Let $m \in \mathbb{N}$, and let $L = L_{\mathrm{P}}(M)$ for some monotone lexicalized RRWW-automaton $M = (Q, \Sigma, \Gamma, ¢, \$, q_0, k, \delta)$. Thus, $L = \mathsf{Pr}^{\Sigma}(L_{\mathrm{C}}(M))$. Consider the deterministic RRW-automaton $M' := (Q, \Gamma, \Gamma, ¢, \$, q_0, k, \delta)$ that is obtained from $M$ by simply interpreting all symbols of $\Gamma$ as input symbols. Then $M'$ is also monotone, and $L(M') = L_{\mathrm{C}}(M') = L_{\mathrm{C}}(M)$. It follows that $L(M')$ is a deterministic context-free language, which in turn implies that there exists a monotone deterministic RR-automaton $\hat{M} = (\hat{Q}, \Gamma, \Gamma, ¢, \$, \hat{q}_0, \hat{k}, \hat{\delta})$ satisfying $L(\hat{M}) = L(M') = L_{\mathrm{C}}(M)$. It follows that $\hat{M}_\Sigma := (\hat{Q}, \Sigma, \Gamma, ¢, \$, \hat{q}_0, \hat{k}, \hat{\delta})$ is a monotone deterministic RRWW-automaton satisfying $L_{\mathrm{P}}(\hat{M}_\Sigma) = L$. In fact, $\hat{M}$ can be designed in such a way that $\hat{M}_\Sigma$ is lexicalized with the same constant as the original automaton $M$. Here we simply have to guarantee that no rewriting meta-instruction of $\hat{M}$ can be applied to any word that contains a factor from $(\Gamma \setminus \Sigma)^*$ of length exceeding $j$, where $j$ is the corresponding constant for $M$. Hence, $\hat{M}_\Sigma$ is strongly lexicalized. Actually, if $M$ has word-expansion of degree $m$, then so does $\hat{M}_\Sigma$. This completes the proof. □

The proper languages of monotone (strongly) lexicalized RRWW-automata with word-expansion of degree 0 are exactly the deterministic context-free languages, while the proper languages of monotone (strongly) lexicalized RRWW-automata with unbounded word-expansion cover all context-free languages (Theorem 3.4). Hence, the degree of word-expansion of monotone (strongly) lexicalized RRWW-automata can be interpreted as a measure for the degree of nondeterminism of context-free languages. It remains to

show that the resulting classes of proper languages form an infinite hierarchy. For doing so we consider a number of example languages. For the following considerations we fix the alphabet $\Sigma_0 := \{a, b\}$.

**Proposition 3.10** *The language* $L_{\mathrm{pal}} := \{\, ww^R \mid w \in \Sigma_0^* \,\}$ *of palindromes of even length belongs to the class* $\mathcal{L}_{\mathrm{P}}(\mathsf{str\text{-}W(1)\text{-}mon\text{-}RRWW})$*, but it is not contained in the class* $\mathcal{L}_{\mathrm{P}}(\mathsf{W(0)\text{-}RRWW})$*.*

**Proof.** Let $M_{\mathrm{pal}}$ be the RRWW-automaton that is given through the meta-instructions $(\mathrm{¢} \cdot \Sigma_0^*, xCx \to C, \Sigma_0^* \cdot \$)$ and $(\mathrm{¢} \cdot C \cdot \$, \mathsf{Accept})$, where $x \in \Sigma_0$. $M_{\mathrm{pal}}$ is deterministic, as to each word over the alphabet $\Sigma_0 \cup \{C\}$ at most one of its meta-instructions applies, and the place of rewriting is unambiguous. Further, all rewrite steps are simply deletions, and it is easily seen that $M_{\mathrm{pal}}$ is monotone, and that $\mathsf{W}(M_{\mathrm{pal}}) = 1$. Also it is rather obvious that $L_{\mathrm{P}}(M_{\mathrm{pal}}) = L_{\mathrm{pal}}$ holds.

On the other hand, it is known that $L_{\mathrm{pal}}$ is not a Church-Rosser language [8], and so it is not the input language of any deterministic RRWW-automaton. However, each lexicalized RRWW-automaton with word-expansion of degree 0 is just a deterministic RRW-automaton. For such an automaton the proper language, the input language, and the characteristic language are all identical. Accordingly, $L_{\mathrm{pal}}$ is not the proper language of any deterministic RRW-automaton, which implies that $L_{\mathrm{pal}} \notin \mathcal{L}_{\mathrm{P}}(\mathsf{W(0)\text{-}RRWW})$. □

Now, for all $m \geq 2$, let $L_p(m) := L_{\mathrm{pal}} \cdot (\{c\} \cdot L_{\mathrm{pal}})^{m-1}$.

**Proposition 3.11**
$L_p(m) \in \mathcal{L}_{\mathrm{P}}(\mathsf{str\text{-}W}(m)\mathsf{\text{-}mon\text{-}RRWW}) \setminus \mathcal{L}_{\mathrm{P}}(\mathsf{W}(m-1)\mathsf{\text{-}RRWW})$ *for all* $m \geq 2$*.*

**Proof.** Let $m \geq 2$, and let $M_m$ be the RRWW-automaton that is given by the following sequence of meta-instructions, where $x \in \Sigma_0$:

$$
\begin{array}{ll}
(0) & (\mathrm{¢} \cdot (Cc)^{m-1}C \cdot \$, \mathsf{Accept}), \\
(1) & (\mathrm{¢} \cdot \Sigma_0^*, xCx \to C, \Sigma_0^* \cdot (c \cdot \Sigma_0^* \cdot C \cdot \Sigma_0^*)^{m-1} \cdot \$), \\
(2) & (\mathrm{¢} \cdot Cc \cdot \Sigma_0^*, xCx \to C, \Sigma_0^* \cdot (c \cdot \Sigma_0^* \cdot C \cdot \Sigma_0^*)^{m-2} \cdot \$), \\
\ldots & \qquad\qquad \ldots \\
(m) & (\mathrm{¢} \cdot (Cc)^{m-1} \cdot \Sigma_0^*, xCx \to C, \Sigma_0^* \cdot \$).
\end{array}
$$

Then $M_m$ is a monotone deterministic RRWW-automaton, and it is easily seen that $L_{\mathrm{C}}(M_m) = \hat{L}_{\mathrm{pal}} \cdot (c \cdot \hat{L}_{\mathrm{pal}})^{m-1}$, where $\hat{L}_{\mathrm{pal}}$ is the language of palindromes of even length with the middle marked by an occurrence of the symbol $C$. Thus, $L_{\mathrm{P}}(M_m) = L_p(m)$. As $M_m$ has word-expansion of degree $m$, and as it is strongly lexicalized, this proves that $L_p(m) \in \mathcal{L}_{\mathrm{P}}(\mathsf{str\text{-}W}(m)\mathsf{\text{-}mon\text{-}RRWW})$.

On the other hand, assume that $M$ is any lexicalized RRWW-automaton with word-expansion $m-1$ such that $L_{\mathrm{P}}(M) = L_p(m)$ holds, let $\Sigma :=$

$\Sigma_0 \cup \{c\}$, and let $\Gamma$ be the tape alphabet of $M$. For a word of the form $z := w_1 w_1^R c w_2 w_2^R c \cdots c w_m w_m^R \in L_p(m)$, where $w_1, \ldots, w_m \in \Sigma_0^*$ are words of sufficient length, there exists a word $\alpha \in L_{\mathrm{C}}(M)$ such that $\mathsf{Pr}^{\Sigma}(\alpha) = z$ and $|\alpha|_{\Gamma \setminus \Sigma} \leq m - 1$. Thus, the middle of at least one of the palindromes $w_i w_i^R$, $1 \leq i \leq m$, is not marked by an occurrence of an auxiliary symbol. This means that $M$ will not be able to correctly process this particular palindrome, as $M$ is deterministic, and as it must satisfy the correctness preserving property. It follows that $L_{\mathrm{P}}(M) \neq L_p(m)$ holds, implying that $L_p(m) \notin \mathcal{L}_{\mathrm{P}}(\mathsf{W}(m-1)\text{-}\mathsf{RRWW})$. □

From these propositions we immediately obtain the following proper hierarchy results.

**Theorem 3.12** *For each $m \in \mathbb{N}$, the following relations hold:*

| | | | |
|---|---|---|---|
| (a) | $\mathcal{L}_{\mathrm{P}}(\mathsf{W}(m)\text{-}\mathsf{mon}\text{-}\mathsf{RRWW})$ | $\subset$ | $\mathcal{L}_{\mathrm{P}}(\mathsf{W}(m+1)\text{-}\mathsf{mon}\text{-}\mathsf{RRWW})$. |
| (b) | $\mathcal{L}_{\mathrm{P}}(\mathsf{str}\text{-}\mathsf{W}(m)\text{-}\mathsf{RRWW})$ | $\subset$ | $\mathcal{L}_{\mathrm{P}}(\mathsf{str}\text{-}\mathsf{W}(m+1)\text{-}\mathsf{RRWW})$. |
| (c) | $\mathcal{L}_{\mathrm{P}}(\mathsf{W}(m)\text{-}\mathsf{RRWW})$ | $\subset$ | $\mathcal{L}_{\mathrm{P}}(\mathsf{W}(m+1)\text{-}\mathsf{RRWW})$. |
| (d) | $\mathcal{L}_{\mathrm{P}}(\mathsf{str}\text{-}\mathsf{W}(m+1)\text{-}\mathsf{mon}\text{-}\mathsf{RRWW})$ | $\not\subseteq$ | $\mathcal{L}_{\mathrm{P}}(\mathsf{W}(m)\text{-}\mathsf{RRWW})$. |

Finally, let $L_{\mathrm{pal}^+} := \bigcup_{i \geq 1} (\{c\} \cdot L_{\mathrm{pal}})^i$.

**Proposition 3.13** $L_{\mathrm{pal}^+} \notin \mathcal{L}_{\mathrm{P}}(\mathsf{W}(m)\text{-}\mathsf{RRWW})$ *for any $m \geq 0$.*

**Proof.** Let $M'$ be a lexicalized RRWW-automaton with word-expansion of degree $m$, and let $w := c w_1 c w_2 c \cdots c w_{m+1}$, where $w_1, \ldots, w_{m+1}$ are palindromes of sufficient even length over $\Sigma_0$. In order to enable $M'$ to accept the word $w$, auxiliary symbols are needed to mark the middle of each of these palindromes just as in the proofs above. However, as $M'$ only has word-expansion of degree $m$, the middle of at most $m$ of these palindromes can be marked by an auxiliary symbol. It follows that $L_{\mathrm{P}}(M') \neq L_{\mathrm{pal}^+}$. □

On the other hand, it is easily seen that $L_{\mathrm{pal}^+}$ is the proper language of the strongly lexicalized RRWW-automaton $M_{\mathrm{pal}^+}$ on $\Gamma := \{a, b, c, C\}$ that is given through the following meta-instructions, where $x \in \Sigma_0$:

(1) $(\mathrm{¢} \cdot (cC)^* \cdot c \cdot \Sigma_0^*, xCx \to C, \Sigma_0^* \cdot (c \cdot \Sigma_0^* \cdot C \cdot \Sigma_0^*)^* \cdot \$)$,
(2) $(\mathrm{¢} \cdot (cC)^+ \cdot \$, \mathsf{Accept})$.

Obviously, $M_{\mathrm{pal}^+}$ is monotone, but it has unbounded word expansion. Thus, we obtain the following proper inclusions.

**Corollary 3.14**

(a) $\bigcup_{m \geq 0} \mathcal{L}_{\mathrm{P}}(\mathsf{W}(m)\text{-}\mathsf{mon}\text{-}\mathsf{RRWW}) \subset \mathcal{L}_{\mathrm{P}}(\mathsf{lex}\text{-}\mathsf{mon}\text{-}\mathsf{RRWW})$.
(b) $\bigcup_{m \geq 0} \mathcal{L}_{\mathrm{P}}(\mathsf{str}\text{-}\mathsf{W}(m)\text{-}\mathsf{RRWW}) \subset \mathcal{L}_{\mathrm{P}}(\mathsf{str}\text{-}\mathsf{RRWW})$.
(c) $\bigcup_{m \geq 0} \mathcal{L}_{\mathrm{P}}(\mathsf{W}(m)\text{-}\mathsf{RRWW}) \subset \mathcal{L}_{\mathrm{P}}(\mathsf{lex}\text{-}\mathsf{RRWW})$.

According to Theorem 3.12 we have three hierarchies of language classes that are based on the degree of word-expansion of lexicalized RRWW-automata. It remains to separate these hierarchies from one another.

First we show that the classes of the monotone hierarchy are strictly contained in the corresponding classes of the non-monotone hierarchies. For establishing this separation it suffices to realize that there exists a deterministic RR-automaton $M$ such that the input language $L := L(M)$ is not context-free (see, e.g., [12]). As a deterministic RR-automaton, $M$ can also be seen as a strongly lexicalized RRWW-automaton with word-expansion of degree 0. Since $L = L_{\mathrm{C}}(M) = L_{\mathrm{P}}(M)$, it follows that $L \in \mathcal{L}_{\mathrm{P}}(\mathsf{str\text{-}W(0)\text{-}RRWW})$. However, $L$ is not the proper language of any monotone lexicalized RRWW-automaton by Theorem 3.4. Thus, we obtain the following separation result.

**Corollary 3.15**
$\mathcal{L}_{\mathrm{P}}(\mathsf{W}(m)\text{-}\mathsf{mon}\text{-}\mathsf{RRWW}) \subsetneq \mathcal{L}_{\mathrm{P}}(\mathsf{str\text{-}W}(m)\text{-}\mathsf{RRWW})$ *for all* $m \geq 0$.

Finally, we want to separate the hierarchy of proper languages of strongly lexicalized RRWW-automata from the corresponding hierarchy for lexicalized RRWW-automata. To this end we consider the example language

$$L_{\mathrm{expo}} := \{\, a^{i_0}ba^{i_1}b\cdots a^{i_{n-1}}ba^{i_n} \mid \begin{array}{l} n \geq 0,\ i_0,\ldots,i_n \geq 0, \text{ and} \\ \exists m \geq 0 : \sum_{j=0}^{n} 2^j \cdot i_j = 2^m \,\} \cup b^*, \end{array}$$

for which we have the following result.

**Proposition 3.16** $L_{\mathrm{expo}} \in \mathcal{L}_{\mathrm{P}}(\mathsf{W}(0)\text{-}\mathsf{RRWW}) \setminus \mathcal{L}_{\mathrm{P}}(\mathsf{str\text{-}RRWW})$.

**Proof.** Let $M$ be the deterministic RRW-automaton that is given through the following meta-instructions:

(1) $(\mathrm{¢} \cdot a^*, aab \to ba, \Sigma_0^* \cdot \$)$,
(2) $(\mathrm{¢}, b \to \lambda, \Sigma_0^* \cdot \$)$,
(3) $(\mathrm{¢} \cdot a^*, a^4 \to baa, \$)$,
(4) $(\mathrm{¢} \cdot \{\lambda, a, aa\} \cdot \$, \mathsf{Accept})$.

If $w = b^m$ for some $m \geq 0$, then obviously $w$ is accepted by $M$. If $w = a^{i_0}ba^{i_1}b\cdots a^{i_{n-1}}ba^{i_n}$ is given as input to $M$, where $n, i_0, \ldots, i_n \geq 0$, and $\sum_{j=0}^{n} 2^j \cdot i_j = 2^m$ for some $m \geq 0$, then the first occurrence of $b$ is first shifted to the left end of the word. As $\sum_{j=0}^{n} 2^j \cdot i_j = 2^m$, we see that $i_0$ is an even number. Thus, this particular occurrence of $b$ is then deleted. This results in the word $w_1 := a^{i_0/2+i_1}b\cdots a^{i_{n-1}}ba^{i_n}$. As $i_0/2+i_1+\sum_{j=2}^{n} 2^{j-1}\cdot i_j = (\sum_{j=0}^{n} 2^j \cdot i_j)/2 = 2^{m-1}$, we see that this word belongs to the language $L_{\mathrm{expo}}$. This continues until all occurrences of the letter $b$ have been deleted. The resulting word is of the form $a^{2^l}$ for some $l \geq 0$. If $l \leq 1$, then the word is accepted, otherwise an occurrence of $b$ is generated at the right end of the word and shifted through the word as described above, which results in the word $a^{2^{l-1}}$. It follows that $L(M) = L_{\mathrm{P}}(M) = L_{\mathrm{expo}}$. As $M$ is an RRW-automaton, we see that $L_{\mathrm{expo}} \in \mathcal{L}_{\mathrm{P}}(\mathsf{W}(0)\text{-}\mathsf{RRWW})$ holds.

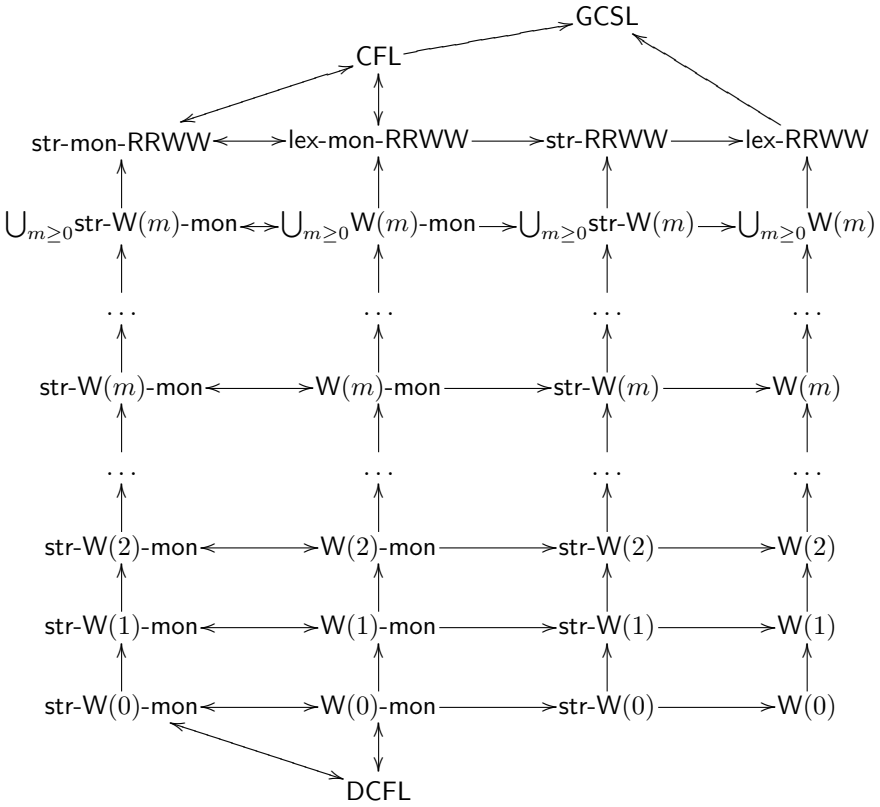


Figure 1: Inclusion relations between language classes defined by various types of lexicalized RRWW-automata. Here str-W($m$)-mon denotes the language class $\mathcal{L}_{\mathrm{P}}$(str-W($m$)-mon-RRWW), and similarly for the other classes. An arrow denotes a proper inclusion, while a double arrow denotes equality. Classes that are not (directly or indirectly) connected are incomparable under inclusion.

On the other hand, assume that $M'$ is a strongly lexicalized RRWW-automaton with input alphabet $\Sigma_0$ and tape alphabet $\Gamma$ such that $L_{\mathrm{expo}} = L_{\mathrm{P}}(M')$ holds. Then a contradiction can be derived in the same way as in the proof of Proposition 3.3. Thus, it follows that $L_{\mathrm{expo}}$ is not the proper language of any strongly lexicalized RRWW-automaton. □

The inclusion results on the classes of proper languages of the various types of lexicalized RRWW-automata are summarized in the diagram in Figure 1.

# 4 Concluding Remarks

We have introduced the degree of word-expansion as a new measure for the degree of nondeterminism for proper languages of restarting automata. Based on this measure we have obtained infinite hierarchies of language classes for monotone and for non-monotone RRWW-automata that are (strongly) lexicalized. In the monotone case these classes form an infinite hierarchy between DCFL and CFL.

It is known that it is decidable whether a given RRWW-automaton is monotone [7]. Here we are concerned with the properties of being lexicalized and of having word-expansion of finite degree.

**Proposition 4.1** *The following problems are decidable:*

(a) *INSTANCE :* *A deterministic* RRWW*-automaton* $M$ *and* $j \in \mathbb{N}$.
*QUESTION :* *Is* $M$ *lexicalized with constant* $j$*?*

(b) *INSTANCE :* *A deterministic* RRWW*-automaton* $M$.
*QUESTION :* *Is* $M$ *lexicalized?*

(c) *INSTANCE :* *A lexicalized* RRWW*-automaton* $M$ *and* $m \in \mathbb{N}$.
*QUESTION :* *Does* $M$ *have word-expansion of degree* $m$*?*

(d) *INSTANCE :* *A lexicalized* RRWW*-automaton* $M$.
*QUESTION :* *Does* $M$ *have word-expansion of finite degree?*

**Proof.** Let $M = (Q, \Sigma, \Gamma, ¢, \$, q_0, k, \delta)$ be a deterministic RRWW-automaton that is given through a sequence of rewriting meta-instructions $((E_{i,1}, u_i \to v_i, E_{i,2}))_{1 \le i \le r}$ and an accepting meta-instruction $(E_0, \mathsf{Accept})$. By LEFT we denote the language that is described by the regular expression $\text{LEFT} := E_0 \cup \bigcup_{i=1}^{r} (E_{i,1} \cdot u_i \cdot E_{i,2})$. It consists of those words to which some meta-instruction of $M$ applies. Thus, $\text{LEFT} = \mathsf{NIR}(M)$. Then $M$ is lexicalized with constant $j$ if and only if $\text{LEFT} \subseteq \Delta^{\le j} \cdot (\Sigma \cdot \Delta^{\le j})^*$, where $\Delta := \Gamma \setminus \Sigma$. Hence, it is decidable whether $M$ is lexicalized with constant $j$.

Further, for the set LEFT we can effectively construct a deterministic finite-state acceptor $A$. Then the number $p$ of states of $A$ can serve as the constant in the pumping lemma for the regular language LEFT. Now $M$ is not lexicalized if and only if there exists a word in the language LEFT that contains a factor from $\Delta^*$ of length $p$. Thus, $M$ is lexicalized if and only if it is lexicalized with constant $p - 1$. This, however, is decidable as seen above.

Next observe that $\mathsf{W}(M) = m$ if and only if the regular language $\text{LEFT}_\Delta := \mathsf{Pr}^\Delta(\text{LEFT})$ satisfies the condition $\text{LEFT}_\Delta \subseteq \Delta^{\le m}$. Finally, $M$ has word-expansion of finite degree if and only if $\mathsf{W}(M) < p'$, where $p'$ is the pumping constant for the language $\text{LEFT}_\Delta$. As above this constant can be determined from $\text{LEFT}_\Delta$. □

Based on the above results the minimal constant $j$ of lexicalization can

be determined for a given lexicalized RRWW-automaton. Also the minimal degree of word-expansion can be computed, in case it is finite. In contrast to the above decidability results, we have the following undecidability result for languages.

**Proposition 4.2** *The following problem of languages is undecidable:*

*INSTANCE :* *A context-free language $L$ and a constant $m \in \mathbb{N}$.*
*QUESTION :* *Does $L \in \mathcal{L}_{\mathrm{P}}(\mathsf{W}(m)\text{-mon-RRWW})$ hold?*

**Proof.** For $m = 0$ this is simply the problem of deciding whether a given context-free language is deterministic context-free, which is known to be undecidable (see, e.g., [6]). For $m \geq 1$, consider the language $L_m := L \cdot c \cdot L_p(m)$, where we assume that $c$ is not contained in the alphabet of $L$. For accepting the suffix $L_p(m)$ we need $m$ occurrences of auxiliary symbols according to Proposition 3.11. Thus, $L_m \in \mathcal{L}_{\mathrm{P}}(\mathrm{W}(m)\text{-mon-RRWW})$ if and only if $L \in \mathsf{DCFL}$. This means that the problem of deciding whether or not $L_m$ is accepted by a monotone lexicalized RRWW-automaton with word-expansion of degree $m$ is undecidable. □

Any lexicalized RRWW-automaton has word-expansion that is bounded from above by a linear function. Thus, it is conceivable that there are languages that cannot occur as proper languages of lexicalized RRWW-automata with a constant degree of word-expansion, but which can be obtained as proper languages of lexicalized RRWW-automata for which the degree of word-expansion is bounded from above by a slowly growing sublinear function.

# References


[1] H. Bordihn and J. Dassow. A note on the degree of nondeterminism. In: G. Rozenberg and A. Salomaa (eds.), *Developments in Language Theory 1993, Proc.*, World Scientific, Singapore, 1994, 70–80.

[2] G. Buntrock. *Wachsende kontextsensitive Sprachen.* Habilitationsschrift, Universität Würzburg, 1996.

[3] G. Buntrock and F. Otto. Growing context-sensitive languages and Church-Rosser languages. *Inform. Comput.* 141 (1998) 1–36.

[4] J. Goldstine, C. Kintala, and D. Wotschke. On measuring nondeterminism in regular languages. *Inform. Comput.* 86 (1990) 179–194.

[5] J. Goldstine, H. Leung, and D. Wotschke. Measuring nondeterminism in pushdown automata. *Journal of Computer and System Sciences* 71

(2005) 440–466. An extended abstract appeared in: R. Reischuk and M. Morvan (eds.), *STACS 97, Proc.*, *Lect. Notes Comput. Sci. 1200*, Springer, Berlin, 1997, 295–306.

[6] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, MA, 1979.

[7] P. Jančar, F. Mráz, M. Plátek, and J. Vogel. On monotonic automata with a restart operation. *J. Autom. Lang. Comb.* 4 (1999) 287–311.

[8] T. Jurdziński and K. Loryś. Church-Rosser languages vs. UCFL. In: P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz, and R. Conejo (eds.), *ICALP 2002, Proc.*, *Lect. Notes Comput. Sci. 2380*, Springer, Berlin, 2002, 147–158.

[9] V. Kuboň and M. Plátek. A grammar based aproach to a grammar checking of free word order languages. In: *COLING 1994, Proc.*, Volume II, Kyoto, Japan, 1994, 906–910.

[10] M. Lopatková, M. Plátek, and V. Kuboň. Modeling syntax of free word-order languages: Dependency analysis by reduction. In: V. Matoušek, P. Mautner, and T. Pavelka (eds.), *TSD 2005, Proc.*, *Lect. Notes Comput. Sci. 3658*, Springer, Berlin, 2005, 140–147.

[11] M. Lopatková, M. Plátek, and P. Sgall. Towards a formal model for functional generative description: Analysis by reduction and restarting automata. *The Prague Bull. of Math. Linguistics* 87 (2007) 7–26.

[12] F. Otto. Restarting automata. In: Z. Ésik, C. Martin-Vide, and V. Mitrana (eds.), *Recent Advances in Formal Languages and Applications*, Studies in Computational Intelligence, Vol. 25, Springer, Berlin, 2006, 269–303.

[13] F. Otto, M. Katsura, and Y. Kobayashi. Infinite convergent string-rewriting systems and cross-sections for finitely presented monoids. *J. Symb. Comput.* 26 (1998) 621–648.

[14] K. Salomaa and S. Yu. Nondeterminism degrees for context-free languages. In: J. Dassow, G. Rosenberg and A. Salomaa (eds.), *Developments in Language Theory II, Proc.*, World Scientific, Singapore, 1996, 154–165.