Regularity and Context-Freeness over Word Rewriting Systems

Didier Caucal and Trong Hieu Dinh

LIGM, UMR CNRS 8049, Université Paris-Est, Marne-la-Valle, France
{caucal,dinh}@univ-mlv.fr

Abstract. We describe a general decomposition mechanism to express the derivation relation of a word rewriting system R as the composition of a (regular) substitution followed by the derivation relation of a system $R' \cup D$, where R' is a strict sub-system of R and D is the Dyck rewriting system. From this decomposition, we deduce that the system R (resp. R^{-1}) preserves regular (resp. context-free) languages whenever $R' \cup D$ (resp. its inverse) does. From this we can deduce regularity and context-freeness preservation properties for a generalization of tagged bifix systems.

1 Introduction

A central problem in the reachability analysis of word rewriting systems is, given a language L and a system R, to determine the set $\longrightarrow_R^*(L)$ of all words which can be derived by R from some word in L. Though this set is not recursive in general for L finite, a lot of attention has been devoted to the characterization of rewriting systems whose derivation relation $\longrightarrow_R^*$ preserves classes of languages with good decidability or closure properties. In particular, a system R is said to preserve regularity (resp. context-freeness) if, for any regular (resp. context-free) language L, $\longrightarrow_R^*(L)$ is also regular (resp. context-free).

Many classes of rewriting systems preserving regularity or context-freeness can be found in the literature. For instance, it is well known that the prefix derivation of any finite rewriting system preserves regularity [8,9], and that the so-called context-free systems (systems whose left-hand sides are of length at most 1) preserve context-free languages and their inverse derivations preserve regularity (see for instance [6]). In [13], Hofbauer and Waldmann proved that the derivation of any finite deleting system preserves regularity and that its inverse derivation preserves context-freeness, thus completing a result by Hibbard [12]. They provided a clever decomposition of the derivation relation into a finite substitution followed by the derivation of an inverse context-free system and a restriction to the original alphabet. From this, they were able to deduce many previously known preservation results. In [10], Endrullis, Hofbauer and Waldmann gave a general decomposition of the derivation of any system into a context-free system followed by an inverse context-free system with empty right hand sides. The main contribution of our paper is to use this derivation decomposition idea to extend the decomposition of [13] to infinite rewriting systems with prefix and suffix rules.

M. Hofmann (Ed.): FOSSACS 2011, LNCS 6604, pp. 214–228, 2011.

Our construction is based on the following observation. Given a word $u = a_1 \dots a_n$, let us write $\overleftarrow{u} = \overleftarrow{a_n} \dots \overleftarrow{a_1}$ and $\overrightarrow{u} = \overrightarrow{a_n} \dots \overrightarrow{a_1}$ where $\overleftarrow{a}$ and $\overrightarrow{a}$ are fresh letters for all a. Let R be a rewriting system and $u \to v \in R$ be one of its rules, and consider factors of the form $\overleftarrow{u_1} v \overrightarrow{u_2}$ with $u_1 u_2 = u$. The intended meaning is that as an effect of this rewrite rule, right-hand side v can be inserted at a certain position i in a word provided that u_1 can be erased to the left of position i and u_2 to the right. In other words, applying rule $u \to v$ to a word can be simulated by first inserting some such factor $\overleftarrow{u_1} v \overrightarrow{u_2}$ at an appropriate position, and then erasing factors of the form $u\overleftarrow{u}$ or $\overrightarrow{u}u$. This double-phased procedure described in [10] can be performed as a substitution followed by a normalization using inverse context-free rules of the form $\overrightarrow{a}a \to \varepsilon$ and $a\overleftarrow{a} \to \varepsilon$ (constituting what we call the Dyck rewriting system, see also [15]).

Under certain syntactical criteria, this simulation step can be used to eliminate rewrite rules from the original rewriting system altogether. More precisely, we are able to decompose the derivation of a system R into a (regular) substitution h, whose role is to insert factors (as described above) corresponding to some subset R' of R, followed by the derivation according to a system $S \cup D$, where S is simply $R - R'$ and D denotes the Dyck system; we say that R can be decomposed into S. As a consequence, the derivation of R (resp. R^{-1}) preserves regularity (resp. context-freeness) if the derivation of $S \cup D$ (resp. its inverse) does. This remains true even for infinite systems, as long as the relation R' is recognizable.

This result can be used to characterize several families of systems whose derivations preserve regularity or context-freeness. First, we observe that in the case of deleting systems the decomposition yields an empty S (i.e. all rules can be simulated and eliminated from R). Since the Dyck system is inverse context-free, this indeed extends the result of [13] to infinite (recognizable) systems. Moreover, contrary to [13] our decomposition only uses a single inverse context-free system, namely D. Note however that many other systems can be directly decomposed into the empty system, for instance the well-known prefix rewriting systems (which encode pushdown system transition relations), their bifix variant, and left-to-right systems; in the finite case, most of these systems can also be simulated by deleting systems [13]. As an example, since multi-pushdown systems as defined in [7] can be seen as left-to-right systems, we can recover from our results that their transition relations preserve context-freeness and their inverse preserves regularity.

Our main application concerns tagged systems, which generalize the notions of prefix and suffix rewriting. Given a set of special symbols called tags, which we separate into prefix and suffix tags, we consider rules of the form $\#u \longrightarrow \#'v$, where $\#, \#'$ are prefix tags and u does not contain any tags. We also allow suffix, bifix, and untagged rules which are defined similarly. Since v may contain tags, this strictly extends the earlier notions of tagged systems defined in [1]. If the set of tagged rules is recognizable and the set of untagged rules is context-free, we show that our decomposition result applies, which entails that the derivation of such a system (resp. its inverse) preserves context-freeness (resp. regularity). This

result still holds when we do not partition the set of tags, at the cost of imposing that tags in the left-hand side of a rule remain invariant in the corresponding right-hand side. Both results extend previously known preservation properties of simpler tagged systems [1].

The remainder of the paper is organized as follows. After some elementary notations and definitions, Section 2 presents our derivation decomposition theorem, and relates it to the class of deleting systems. Section 3 details several classes of rewriting systems whose known preservation results can be recovered using our technique (prefix, suffix and bifix systems in Section 3.1) or for which new preservation results can be shown (left-to-right systems in Section 3.2, tagged systems in Section 3.3).

2 Derivation Decomposition

This section focuses on regularity and context-freeness preservation properties for rewriting systems. After reviewing some known preservation results (Section 2.2), we generalize the derivation decomposition of [13] to arbitrary rewriting systems (Section 2.3), which allows us to deduce new preservation properties. We start by recalling some basic definitions and notations.

2.1 Notations

For ease of notation, a singleton $\{x\}$ will often be identified with x. The image by a binary relation $R \subseteq E \times F$ of a subset $P \subseteq E$ by a binary relation R is $R(P) = \{y \mid \exists\, x \in P,\ x\, R\, y\}$. Let N be a finite set of symbols (called an *alphabet*), we write $\mathrm{Alph}(u) = \{u(i) \mid 1 \leq i \leq |u|\}$ the set of letters occurring in a word $u \in N^*$ (whose $u(i)$ the letter of u at position i). This is extended by union to any language P over N: $\mathrm{Alph}(P) = \{a \mid \exists\, u \in P,\ a \in \mathrm{Alph}(u)\ \}$. The *concatenation* of binary relations R, S on N^* is $R.S = \{(ux, vy) \mid u\, R\, v \wedge x\, S\, y\}$, and the left and right concatenation of R by a language $P \subseteq N^*$ is $R.P = R.\mathrm{Id}_P = \{(uw, vw) \mid u\, R\, v \wedge w \in P\}$ and $P.R = \mathrm{Id}_P.R$, where $\mathrm{Id}_P = \{(w, w) \mid w \in P\}$ denotes the *identity relation* on P.

A *regular language* over N is the language recognized by a finite automaton labelled in N (or N^*). A *substitution* h over N is a binary relation on N^* whose image $h(a)$ is defined for every letter $a \in N$ and extended by morphism to words: $h(a_1 \ldots a_n) = h(a_1) \ldots h(a_n)$ for all $n \geq 0$ and $a_1, \ldots, a_n \in N$. It is said to be finite (resp. regular) if $h(a)$ is a finite (resp. regular) language for all $a \in N$. A *recognizable relation* R on N^* is a finite union of binary products of regular languages: $R = U_1 \times V_1 \cup \ldots \cup U_p \times V_p$ for some $p \geq 0$ and regular languages $U_1, V_1, \ldots, U_p, V_p$. A *transducer* A over N is an automaton labelled in $N^* \times N^*$ whose language is interpreted as a binary relation, called a *rational relation* [4].

A *word rewriting system* (or just *system*) R over an alphabet N is a binary relation on N^* seen as a set of rules (u, v); we do not assume R to be finite. Let $\mathrm{Alph}(R)$ be the set of letters of R. The *rewriting relation* (or single step reduction) of R is the binary relation $\longrightarrow_R = N^*.R.N^*$, *i.e.* $xuy \longrightarrow_R xvy$ for

all $u\,R\,v$ and $x, y \in N^*$; we also sometimes write $xuy \longrightarrow_{R,|x|} xvy$ to denote the position $|x|$ where the rule is applied. Note that $(\longrightarrow_R)^{-1} = \longrightarrow_{R^{-1}}$. The *derivation relation* (or reduction relation) $\longrightarrow_R^*$ of R is the reflexive and transitive closure (under composition) of $\longrightarrow_R$, *i.e.* $u \longrightarrow_R^* v$ if there exist $n \geq 0$ and $u_0, \ldots, u_n \in N^*$ such that $u = u_0 \longrightarrow_R u_1 \ldots \longrightarrow_R u_n = v$. Note that $(\longrightarrow_R^*)^{-1} = \longrightarrow_{R^{-1}}^*$.

A *context-free grammar* over N is a finite relation $R \subseteq M \times (M \cup N)^*$ for some alphabet M disjoint of N; it generates from $u \in (M \cup N)^*$ the *context-free language* $L(R, u) = \{v \in N^* \mid u \longrightarrow_R^* v\}$.

2.2 Preservation Properties

A first and very well-known preservation result is that any rational relation R (and its inverse) preserves regularity: the image $R(L)$ of any regular language L remains regular. It is also well-known in the field of language theory that the family of context-free languages is also closed under rational relations [4].

Lemma 2.1. *Any rational relation preserves regularity and context-freeness.*

Since both recognizable relations and regular substitutions are special cases of rational relations, it follows that regular and context-free languages are also closed under direct and inverse recognizable relations and regular substitutions.

In this paper, we are concerned with the characterization of classes of rewriting systems whose derivation relations preserve regularity or context-freeness, using a general decomposition mechanism detailed in the next subsection. A simple way to simulate the application of a rewriting rule (uv, w) to a word x is to insert, at the appropriate position in x, a factor $\overleftarrow{u}\,w\,\overrightarrow{v}$ whose intended meaning is that at this position, right-hand side w can appear after applying the rule if a factor u can be deleted on the left and v on the right. After this word is inserted, appropriate deletions are performed using a single rewriting system called the Dyck system. Therefore, the language preservation properties of that rewriting system play a central role in our study.

Formally, let $R \subseteq N^* \times N^*$ be a rewriting system, and consider a new alphabet $\overleftrightarrow{N} = \overrightarrow{N} \cup \overleftarrow{N}$ consisting of two disjoint copies of N, with $\overrightarrow{N} = \{\overrightarrow{a} \mid a \in N\}$ and $\overleftarrow{N} = \{\overleftarrow{a} \mid a \in N\}$. This notation is extended to words over N as follows: $\overrightarrow{a_1 \ldots a_n} = \overrightarrow{a_n} \ldots \overrightarrow{a_1}$ and $\overleftarrow{a_1 \ldots a_n} = \overleftarrow{a_n} \ldots \overleftarrow{a_1}$ for any $n \geq 0$ and $a_1, \ldots, a_n \in N$. The *Dyck system* $D = N{\downarrow} \cup {\downarrow}N$ defined over $\overline{N} = N \cup \overleftrightarrow{N}$ is the union of the *right* and *left Dyck systems* $N{\downarrow} = \{(\overrightarrow{a}\,a, \varepsilon) \mid a \in N\}$ and ${\downarrow}N = \{\,(a\overleftarrow{a}\,, \varepsilon) \mid a \in N\}$.

For any rule (uv, w) in R, the word xwy obtained by rewriting $xuvy$ can also be derived from the word $xu\overleftarrow{u}\,w\,\overrightarrow{v}vy$ using D. Note that when $u = \varepsilon$ (resp. $v = \varepsilon$), it suffices to use $N{\downarrow}$ (resp. ${\downarrow}N$). It is a classical and widely-used result that the derivation relation of $N{\downarrow}$ preserves regularity [3] but not context-freeness. An example [14] is to take the context-free languages L and M solutions of the equations $L = \overrightarrow{a}La \cup M$ and $M = b \cup aMM\overrightarrow{a}$. So $\xrightarrow[N{\downarrow}]{*}(L)$ is not context-free:

$$\xrightarrow[N{\downarrow}]{*}(L) \cap b^* = \{b^{2^n} \mid n \geq 0\}$$

Furthermore the derivation of $N{\downarrow}^{-1}$ preserves context-freeness but not regularity:

$$\xrightarrow[N{\downarrow}^{-1}]{*}(\varepsilon) \cap \overrightarrow{a}^* a^* \;=\; \{\overrightarrow{a}^n a^n \mid n \geq 0\}$$

which is not regular (but is context-free). We thus call the system $N{\downarrow}$ reg/cf-preserving, as defined below.

Definition 2.2. *A system R is reg/cf-preserving if its derivation relation preserves regularity and its inverse derivation preserves context-freeness.*
A system R is cf/reg-preserving if R^{-1} is reg/cf-preserving.

One can extend the reg/cf-preservation of the (right) Dyck system to wider classes of rewriting systems. We say that a binary relation R on N^* is a *context-free system* if $R \subseteq (N \cup \{\varepsilon\}) \times N^*$ with $R(a)$ a context-free language for all $a \in N \cup \{\varepsilon\}$. The system D^{-1} is a context-free system.

Proposition 2.3 ([5]). *Context-free systems are cf/reg-preserving.*

Another class of cf/reg-preserving systems is defined in [12]. A system R is called *context-limited* if there exists a partial ordering $<$ on N such that for any rule $(u, v) \in R$, any letter of u is less than some letter of v: $\forall\ a \in \mathrm{Alph}(u)\ \exists\ b \in \mathrm{Alph}(v)\ a < b$. It is shown that the derivation relations of finite context-limited systems preserve context-free languages [12]. Additionally, the inverse R^{-1} of a context-limited system R is called a *deleting system*, and the derivation relation of any finite deleting system preserves regularity [13].

Proposition 2.4 ([12,13]). *Finite context-limited systems are cf/reg-preserving.*

This proposition follows from the decomposition [13] of the derivation relation of any finite deleting system R into a finite substitution h over an extended alphabet composed with the derivation of the inverse of a finite context-free system S, and followed by a restriction to the original alphabet:

$$\xrightarrow[R]{*} \;=\; \big(h \circ \xrightarrow[S^{-1}]{*}\big) \cap N^* \times N^*.$$

In the following section, we extend this reasoning to arbitrary rewriting system. We will see in particular that in the case of deleting systems, S^{-1} can always be chosen to be the Dyck system.

2.3 Decomposition

In this subsection we build up on the technical ideas behind Proposition 2.4 and propose a more general notion of derivation decomposition for arbitrary rewriting systems. As already sketched in the previous section, the application of a single rewriting rule (uv, w) to a word x can be simulated by inserting the factor $\overleftarrow{u}\, w\, \overrightarrow{v}$ inside x, and then deleting the extra letters using the Dyck system. We make use of this idea by identifying sets of rules whose role in the derivation can be accurately simulated by this process.

More precisely, for a given rewriting system R over some alphabet N, we identify a subset of rules $R' \subseteq R$ such that

$$\xrightarrow[R]{*} \;=\; \big(h \circ \xrightarrow[(R-R')\cup D]{*}\big) \cap N^*\times N^*$$

where h is a substitution inserting factors of the form $\overleftarrow{u}\,w\,\overrightarrow{v}$. This decomposition is performed by eliminating left or right recursion from the system. Formally, for any $R \subseteq N^*\times N^*$ and $M \subseteq N$, we define the sub-system

$$R_M \;=\; \{(u,v) \in R \mid \mathrm{Alph}(uv) \cap M \neq \emptyset\}$$

consisting of all the rules of R with a letter in M; hence $R - R_M$ is the maximal sub-system of R over $N - M$. We want to decompose the derivation of R into the composition of some substitution h together with the derivation of the system $(R - R_M) \cup D$ for suitable subsets M of N.

Definition 2.5. *A set $M \subseteq \mathrm{Alph}(R)$ is called a prefix sub-alphabet of R if*

$$R \;\subseteq\; MN^* \times M(N-M)^* \;\cup\; N^* \times (N-M)^*.$$

This definition means that for each rule $(u, av) \in R$ with $a \in N$, v has no letter in M, and if $a \in M$ then u must begin by a letter in M (see Example 2.8). Note that the set of prefix sub-alphabets of R is closed under union and we can compute its maximal element (with respect to inclusion). For any prefix sub-alphabet M of R, we define over $\overline{N}$ the language

$$P \;=\; \{\overleftarrow{u}\,w\,\overrightarrow{v} \mid uv\,R\,w \wedge u \in (N-M)^* \wedge v \in MN^*\}$$

and the substitution $h_M : \overline{N} \longrightarrow 2^{\overline{N}^*}$ with $h_M(x) = P^*x$ if $x \in M$ and $h_M(x) = x$ otherwise. Both the language P and the substitution h_M are regular whenever R_M is recognizable. When M is a prefix sub-alphabet of R, we can decompose $\longrightarrow_R^*$ by removing R_M from R.

Lemma 2.6. *For any prefix sub-alphabet M of R and for any $u \in \overline{N}^*$,*

$$\xrightarrow[R\cup D]{*}(u) \cap N^* \;=\; \xrightarrow[(R-R_M)\cup D]{*}\big(h_M(u)\big) \cap N^*.$$

Proof. Let us write $S \;=\; (R - R_M) \cup D$ and $h = h_M$.
$\supseteq$**:** We first establish two preliminary observations.

First, whenever a factor $\overrightarrow{v}$ (resp $\overleftarrow{v}$) can be removed during a derivation by $R \cup D$, one can always rearrange the derivation steps so that at some point factor v appears immediately to the right (resp. left) of $\overrightarrow{v}$ (resp. $\overleftarrow{v}$), and the resulting factor $\overrightarrow{v}v$ (resp. $v\overleftarrow{v}$) is deleted using D. Formally, for any $u, w \in \overline{N}^*$ and any $v, z \in N^*$,

$$w\overrightarrow{v}u \xrightarrow[R\cup D]{*} z \;\Longrightarrow\; \exists\, \overline{w},\; u \xrightarrow[R\cup D]{*} v\overline{w} \;\wedge\; w\overline{w} \xrightarrow[R\cup D]{*} z,$$

$$u\overleftarrow{v}w \xrightarrow[R\cup D]{*} z \;\Longrightarrow\; \exists\, \overline{w},\; u \xrightarrow[R\cup D]{*} \overline{w}v \;\wedge\; \overline{w}w \xrightarrow[R\cup D]{*} z.$$

This can be proven by induction on derivation length. For any $R \subseteq N^* \times N^*$, let

$$\overleftrightarrow{R} \;=\; \{\overleftarrow{u}\, w\, \overrightarrow{v} \mid (uv, w) \in R\}$$

We have

$$\xrightarrow[R\cup D]{*}\big(u[\overleftrightarrow{R}]\big) \cap N^* \;\subseteq\; \xrightarrow[R\cup D]{*}(u) \quad \text{for any } u \in \overline{N}^*,$$

meaning that even randomly inserting factors from $\overleftrightarrow{R}$ in u does not increase the set of words in N^* obtained by derivation using $R \cup D$. In other words, the specific positions at which h inserts factors is only relevant for the converse inclusion (which is proven below). The proof is done using the previous observation and for some word $x \in u[\overleftrightarrow{R}]$, by induction on the minimal number of insertions of words of $\overleftrightarrow{R}$ which must be performed in order to obtain x from u.

Now let $u \in \overline{N}^*$, since $h(u) \subseteq u[\overleftrightarrow{R}]$ and $S \subseteq R \cup D$ and by the above inclusion we obtain

$$\xrightarrow[S]{*}(h(u)) \cap N^* \;\subseteq\; \xrightarrow[R\cup D]{*}\big(u[\overleftrightarrow{R}]\big) \cap N^* \;\subseteq\; \xrightarrow[R\cup D]{*}(u).$$

$\subseteq$**:** Let $u \longrightarrow^*_{R\cup D} v$ with $u \in \overline{N}^*$. Let us show that $h(v) \subseteq \longrightarrow^*_S \big(h(u)\big)$. To prove this inclusion, we need to make sure that the insertion process performed by h does not prevent any of the words originally derivable from u using $R \cup D$ to be also derivable from $h(u)$ using S. Intuitively, this is guaranteed by the definition of the set P and the substitution h_M which only inserts factors at specific positions.

By induction on the length of the derivation of v from u, it remains to check the inclusion for $u \longrightarrow_{R\cup D} v$. Let $u = xu_0y$ and $v = xv_0y$ for some $(u_0, v_0) \in R \cup D$. We distinguish the three complementary cases below.

Case 1: $(u_0, v_0) \notin R_M \cup D$. By definition $u_0, v_0 \in (N - M)^*$. This means that neither u_0 nor v_0 is affected by h: we have $h(u) = h(x)u_0h(y)$ and $h(v) = h(x)v_0h(y)$. Hence

$$h(v) \;=\; h(x)v_0h(y) \;\subseteq\; \underset{\{(u_0,v_0)\}}{\longrightarrow}\big(h(x)u_0h(y)\big) \;\subseteq\; \xrightarrow[S]{*}\big(h(u)\big).$$

Case 2: $(u_0, v_0) \in D$. By definition, $v_0 = \varepsilon$. Thus

$$h(v) \;=\; h(x)h(y) \;\subseteq\; \underset{D}{\longrightarrow}\big(h(x)u_0h(y)\big) \;\subseteq\; \xrightarrow[S]{*}\big(h(xu_0y)\big) \;=\; \xrightarrow[S]{*}\big(h(u)\big).$$

Case 3: $(u_0, v_0) \in R_M$. This rule can be of two types, corresponding to the two subcases below.

Case 3.1: $u_0 \in MN^*$ and $v_0 \in M(N-M)^*$. We have $h(x)P^*u_0h(y) \subseteq h(u)$ and $h(v) = h(x)P^*v_0h(y)$. As $v_0\overrightarrow{u_0} \in P$, $h(x)P^*v_0\overrightarrow{u_0}u_0h(y) \subseteq h(u)$. Hence

$$h(v) \;=\; h(x)P^*v_0h(y) \;\subseteq\; \xrightarrow[D]{*}(h(x)P^*v_0\overrightarrow{u_0}u_0h(y)) \;\subseteq\; \xrightarrow[S]{*}(h(u)).$$

Case 3.2: $u_0 \in N^*MN^*$ and $v_0 \in (N-M)^*$. We have $u_0 = u_0' \# u_0''$ with $u_0' \in (N-M)^*$, $\# \in M$, $u_0'' \in N^*$, and $\overleftarrow{u_0'} v_0 \overrightarrow{\#u_0''} \in P$. Hence $h(x)u_0' P^* \# u_0'' h(y) \subseteq h(u)$, which implies in particular that

$$h(x)u_0'\overleftarrow{u_0'}v_0\overrightarrow{u_0''}\overrightarrow{\#}\#u_0''h(y) \;\subseteq\; h(u).$$

Finally we obtain that

$$h(v) = h(x)v_0h(y) \;\subseteq\; \xrightarrow[D]{*}\big(h(x)u_0'\overleftarrow{u_0'}v_0\overrightarrow{u_0''}\overrightarrow{\#}\#u_0''h(y)\big) \;\subseteq\; \xrightarrow[S]{*}(h(u)).$$

This concludes the proof that $h(v) \subseteq \longrightarrow_S^*(h(u))$ for $u \longrightarrow_{R\cup D}^* v$. As $v \in h(v)$, we finally get $\longrightarrow_{R\cup D}^*(u) \subseteq \longrightarrow_S^*(h(u))$. □

A similar decomposition can also be achieved using suffix sub-alphabets instead of prefix ones. A subset M of N is a *suffix sub-alphabet* of R if

$$R \;\subseteq\; N^*M\times(N-M)^*M \;\cup\; N^*\times(N-M)^*.$$

It can also be seen as a prefix sub-alphabet of $\widetilde{R} = \{(\widetilde{u},\widetilde{v}) \mid u\,R\,v\}$, where $\widetilde{u} = u(|u|)\ldots u(1)$ is the *mirror* of word u. Lemma 2.6 remains true for any suffix sub-alphabet M, with the difference that $h_M(x) = xQ^*$ for all $x \in M$ with $Q = \{\overleftarrow{u}\,w\,\overrightarrow{v} \mid uv\,R\,w \wedge u \in N^*M \wedge v \in (N-M)^*\}$.

Using prefix and suffix sub-alphabets, we can now iterate this decomposition process as long as at least one such sub-alphabet remains. We say that $R \subseteq N^*\times N^*$ is *u-decomposable* for $u \in (2^N)^*$ if $u = \varepsilon$, or $u = Mv$ with M a prefix or suffix sub-alphabet of R and $R - R_M$ is v-decomposable. For any $u \in (2^N)^*$, we define the sub-system R_u of R as $R_u = R_{u(1)} \cup \ldots \cup R_{u(|u|)} = R_{u(1)\cup\ldots\cup u(|u|)}$ consisting of the subset of rules of R with at least one letter in $u(1)\cup\ldots\cup u(|u|)$.

When the letters of u are prefix and suffix sub-alphabets, we define the substitution $h_u : \overline{N} \longrightarrow 2^{\overline{N}^*}$ by $h_u = h_{u(1)} \circ \ldots \circ h_{u(|u|)}$ where for every $1 \le i \le |u|$, $h_{u(i)}$ is the substitution associated to the prefix, or suffix but not prefix, sub-alphabet $u(i)$ of R. Note that if R_u is recognizable, h_u is a regular substitution. Let us now iterate the decomposition of Lemma 2.6.

Proposition 2.7. *If R is u-decomposable then*

$$\xrightarrow[R]{*} \;=\; \big(h_u \circ \xrightarrow[(R-R_u)\cup D]{*}\big) \cap N^*\times N^*.$$

Proof. We have

$$\begin{aligned}
\xrightarrow[R]{*} \;&=\; \xrightarrow[R\cup D]{*} \cap\, N^*\times N^* \\
&=\; \big(h_{u(1)} \circ \xrightarrow[(R-R_{u(1)})\cup D]{*}\big) \;\cap\; N^*\times N^* \text{ by Lemma 2.6} \\
&=\; \big(h_{u(1)} \circ \ldots \circ h_{u(|u|)} \circ \xrightarrow[(R-R_{u(1)}\ldots-R_{u(|u|)})\cup D]{*}\big) \;\cap\; N^*\times N^* \\
&=\; \big(h_u \circ \xrightarrow[(R-R_u)\cup D]{*}\big) \;\cap\; N^*\times N^*.
\end{aligned}$$

□

We say that R is *decomposable* into S if R is u-decomposable for some u and $R - R_u = S$. This decomposition relation is reflexive and transitive. Let us illustrate this mechanism on an example.

Example 2.8. Consider the rewriting system $R = \{(abb, ab), (a, \varepsilon), (cb, cc)\}$ and the derivation $caabb \longrightarrow_R caab \longrightarrow_R cab \longrightarrow_R cb \longrightarrow_R cc$.

As $\{a\}$ is a prefix sub-alphabet of R, and by Lemma 2.6, we have

$$\xrightarrow[R]{*} \;=\; \big(h_a \circ \xrightarrow[R' \cup D]{*}\big) \cap N^* \times N^*$$

with $R' = \{(cb, cc)\}$ and $h_a(a) = \{\overrightarrow{a}, ab\,\overrightarrow{b}\,\overrightarrow{b}\,\overrightarrow{a}\}^* a$.
As $\{b\}$ is a prefix sub-alphabet of R' (but not of R), we have

$$\xrightarrow[R' \cup D]{*} \cap\, \overline{N}^* \times N^* \;=\; \big(h_b \circ \xrightarrow[D]{*}\big) \cap\, \overline{N}^* \times N^*$$

with $h_b(b) = \{\overleftarrow{c}\,cc\,\overrightarrow{b}\}^* b$. Thus R is $\{a\}\{b\}$-decomposable into $\emptyset$ and

$$\xrightarrow[R]{*} \;=\; \big(h_{ab} \circ \xrightarrow[D]{*}\big) \cap N^* \times N^*$$

with $h_{ab}(a) = h_b(h_a(a)) = \{\overrightarrow{a}, a\{\overleftarrow{c}\,cc\,\overrightarrow{b}\}^* b\,\overrightarrow{b}\,\overrightarrow{b}\,\overrightarrow{a}\}^* a$ and $h_{ab}(b) = h_b(h_a(b)) = h_b(b) = \{\overleftarrow{c}\,cc\,\overrightarrow{b}\}^* b$. For instance

$$u \;=\; c.\overrightarrow{a}\,a.\overrightarrow{a}\,a\overleftarrow{c}\,cc\,\overrightarrow{b}\,b\,\overrightarrow{b}\,\overrightarrow{b}\,\overrightarrow{a}\,a.b.b \in h_{ab}(caabb) \quad \text{and} \quad u \xrightarrow[D]{*} cc.$$

Finally R is terminating (no infinite derivation) although R is not match-bounded [11]. □

For any letter $a \in N - \mathrm{Im}(R)$ which does not appear in the right hand sides of rules of R, $\{a\}$ is a prefix (or suffix) sub-alphabet of R and we call a a *reducible letter* of R. We say that R is *reducible* into S if R is *u-decomposable* into S for some word u composed only of reducible letters (of the successive remaining sub-relations). Note that this is *not* the case of the rewriting system in the above example, even though it is decomposable into the empty system. The systems which can be reduced into $\emptyset$ or $\{(\varepsilon, \varepsilon)\}$ are exactly the deleting systems.

Proposition 2.9. *R is deleting if and only if R is reducible into $\emptyset$ or $\{(\varepsilon, \varepsilon)\}$.*

When R is decomposable into S and $R - S$ is recognizable, we say there is a *recognizable decomposition* of R into S. Let us apply Proposition 2.7 to that setting.

Theorem 2.10. *For R recognizable decomposable into S,*

$$\xrightarrow[S \cup D]{*} \text{ preserves regularity} \implies \xrightarrow[R]{*} \text{ preserves regularity,}$$
$$\xrightarrow[S^{-1} \cup D^{-1}]{*} \text{ preserves context-freeness} \implies \xrightarrow[R^{-1}]{*} \text{ preserves context-freeness.}$$

Proof. By Proposition 2.7, there is a regular substitution h such that for all $L \subseteq N^*$, $\longrightarrow_R^*(L) = \longrightarrow_{S \cup D}^*(h(L)) \cap N^*$ and $\longrightarrow_{R^{-1}}^*(L) = h^{-1}\big(\longrightarrow_{S^{-1} \cup D^{-1}}^*(L)\big) \cap N^*$. Since h is a regular substitution, this proves the theorem. □

Note that we cannot suppress D or D^{-1} in Theorem 2.10, and that the reverse implications are false. Indeed the system $S = \{(\#a, bb\#), (b\&, \&a)\}$ has a rational derivation, hence its derivation preserves regularity and context-freeness, but $\longrightarrow_{S \cup D}^*$ does not preserve regularity:

$$\xrightarrow[S \cup D]{*}\big((\#\overrightarrow{\&})^*\#a(\overleftarrow{\#}\&)^*\big) \cap \#a^* \;=\; \{\#a^{2^n} \mid n \geq 0\}$$

and $\longrightarrow_{S \cup D^{-1}}^*$ does not preserve context-freeness:

$$\xrightarrow[S \cup D^{-1}]{*}(a) \cap (\overrightarrow{\#}\&)^*a^*(\overleftarrow{\&}\#)^* \;=\; \{(\overrightarrow{\#}\&)^n a^{2^n} (\overleftarrow{\&}\#)^n \mid n \geq 0\}.$$

To conclude this section, let us apply Theorem 2.10 together with Proposition 2.3 to transfer regularity and context-freeness preservation properties from context-free systems to a larger class of rewriting systems.

Proposition 2.11. *If R^{-1} is recognizable decomposable into S^{-1} where S is a context-free system, then R is cf/reg-preserving.*

By Lemma 2.9, this proposition strictly generalizes Proposition 2.4 by allowing a recognizable set of rules: indeed any recognizable deleting system is recognizable-decomposable into the empty rewriting system (or $\{(\varepsilon, \varepsilon)\}$), whose inverse is trivially a context-free system. This entails that recognizable context-limited systems are cf/reg-preserving. In the following section we give several other applications of Theorem 2.10 and Proposition 2.11.

3 Applications

In this section, we provide several consequences and applications of the decomposition technique presented in the previous section. In particular, we show how to derive from Theorem 2.10 preservation properties for the classes of prefix, suffix and bifix systems as well as their tag-adding variants.

3.1 Prefix, Suffix and Bifix Systems

Proposition 2.4 was already applied in [13] to the prefix derivations of finite systems. Using Theorem 2.10, this can be extended to any recognizable system.

The *prefix rewriting* of a system R is the binary relation $\longmapsto_R = R.N^* = \longrightarrow_{R,0}$, *i.e.* $uy \longmapsto_R vy$ for any uRv and $y \in N^*$. As expected, the *prefix derivation* $\longmapsto_R^*$ of R is the reflexive and transitive closure of the prefix rewriting relation. For any finite system, the regularity of the set of words reached by prefix derivation from a given word [8] is a particular case of the rationality of the prefix derivation; this remains true for any recognizable system.

Proposition 3.1 ([9]). *The prefix derivation of any recognizable system is a rational relation.*

Proof. Let $R = \bigcup_{i=1}^{n} U_i \times V_i$ be a recognizable rewriting system, and $\# \notin N$ be a new symbol, and consider the system $\#R = \{(\#u, \#v) \mid u\,R\,v\}$. By definition, $\{\#\}$ is a prefix sub-alphabet of $\#R$, which is thus recognizable and $\#$-decomposable into $\emptyset$. For any $L \subseteq N^*$, $\longmapsto^*_R(L) = {}^{\#^{-1}}\big(\longrightarrow^*_{\#R}(\#L)\big)$ which by Theorem 2.10, is regular whenever L is regular. By Proposition 2.7, the last equality is equivalent to

$$\xmapsto[R]{*}(L) \;=\; \xrightarrow[N\downarrow]{*}\big(\{v\overrightarrow{u} \mid u\,R\,v\}^*L\big) \cap N^*.$$

Since $\longmapsto^*_{R^{-1}}(U)$ and $\longmapsto^*_R(V)$ remain regular for any regular languages U, V, the relation $\overline{R} = \bigcup_{i=1}^{n} \longmapsto^*_{R^{-1}}(U_i) \times \longmapsto^*_R(V_i)$ is recognizable, hence

$$\xmapsto[R]{*} \;=\; \mathrm{Id}_{N^*} \cup \xmapsto[\overline{R}]{} \;=\; \mathrm{Id}_{N^*} \cup \overline{R}.N^*$$

is recognized by a finite transducer. □

The rules of a system R can also be applied only to suffixes. The *suffix rewriting* of R is the binary relation $\longrightarrow\!\!\!\!\!\rightarrow_R \;=\; N^*.R$, *i.e.* $wu \longrightarrow\!\!\!\!\!\rightarrow_R wv$ for any $u\,R\,v$ and $w \in N^*$. Note that the rewriting relation of a suffix system is isomorphic to that of a prefix one: $u \longrightarrow\!\!\!\!\!\rightarrow_R v$ if and only if $\widetilde{u} \longmapsto_{\widetilde{R}} \widetilde{v}$ where $\widetilde{R} = \{(\widetilde{u}, \widetilde{v}) \mid u\,R\,v\}$. Hence Proposition 3.1 holds for suffix systems as well.

Finally we allow the application of rules both to prefixes and suffixes. The *bifix rewriting relation* of R is $\longmapsto\!\!\!\!\!\rightarrow_R \;=\; \longmapsto_R \cup \longrightarrow\!\!\!\!\!\rightarrow_R$. There exists a generalization of Proposition 3.1 to this type of rewriting.

Proposition 3.2 ([15]). *The bifix derivation of any recognizable system is a rational relation.*

Proof. We take two new symbols $\#, \& \notin N$ and we define the recognizable system $S = \#R \cup R\& = \{(\#u, \#v) \mid u\,R\,v\} \cup \{(u\&, v\&) \mid u\,R\,v\}$. The sets $\{\#\}$ and $\{\&\}$ are respectively prefix and suffix sub-aphabets of S. Thus S is $\#\&$-decomposable in $\emptyset$. For any $L \subseteq N^*$, $\longmapsto\!\!\!\!\!\rightarrow^*_R(L) \;=\; {}^{\#^{-1}}\big(\longrightarrow^*_S(\#L\&)\big)^{\&^{-1}}$ which by Theorem 2.10, is regular whenever L is regular. By Proposition 2.7, the last equality is equivalent to

$$\xmapsto[R]{*}\!\!\!\!\!\rightarrow(L) \;=\; \xrightarrow[D]{*}\big(\{v\overrightarrow{u} \mid u\,R\,v\}^*L\{\overleftarrow{u}\,v \mid u\,R\,v\}^*\big) \cap N^*.$$

This is a possible first step of the construction, given in [15], of a finite transducer recognizing $\longmapsto\!\!\!\!\!\rightarrow^*_R$. □

3.2 Left-to-Right Derivation

Let us apply Theorem 2.10 to another restriction of the derivation. The *left-to-right derivation* $\hookrightarrow^*_R$ of a system R is defined by

$$u \overset{*}{\underset{R}{\hookrightarrow}} v \iff u_0 \underset{R,p_1}{\longrightarrow} u_1 \ldots \underset{R,p_n}{\longrightarrow} u_n \;\text{ with }\; p_1 \leq \ldots \leq p_n, u_0 = u \text{ and } u_n = v.$$

The left-to-right derivation and leftmost derivation are incomparable. In particular, applying a rewrite rule at some position i could in a leftmost derivation enable another rule at some position strictly smaller than i. However, in a left-to-right derivation, successive rewriting positions must be just increasing.

Proposition 3.3. *The left-to-right derivation of any recognizable system preserves context-freeness, and its inverse preserves regularity.*

Proof. We consider a new symbol $\# \notin N$ and the system $S = \{(u, \#v) \mid u\,R\,v\}$. By choosing $\{\#\}$ as a prefix sub-alphabet, we can recognizably decompose S^{-1} into $\emptyset$. Note that S^{-1} is deleting for R finite. By Proposition 2.11, $\longrightarrow_S^*$ thus preserves context-freeness and $\longrightarrow_{S^{-1}}^*$ preserves regularity. Furthermore $\longrightarrow_S^*$ can be performed from left to right: $\longrightarrow_S^* \;=\; \hookrightarrow_S^*$. Let π be the morphism defined by $\pi(\#) = \varepsilon$ and $\pi(a) = a$ for any $a \in N$. We have

$$\overset{*}{\underset{R}{\hookrightarrow}} \;=\; \{(u, \pi(v)) \mid u \in N^* \wedge u \overset{*}{\underset{S}{\hookrightarrow}} v\}.$$

Thus for every $L \subseteq N^*$,

$$\overset{*}{\underset{R}{\hookrightarrow}}(L) \;=\; \pi\big(\overset{*}{\underset{S}{\hookrightarrow}}(L)\big) \;=\; \pi\big(\overset{*}{\underset{S}{\longrightarrow}}(L)\big)$$

hence $\hookrightarrow_R^*(L)$ is context-free whenever L is. Finally for any $L \subseteq N^*$,

$$\big(\overset{*}{\underset{R}{\hookrightarrow}}\big)^{-1}(L) \;=\; \overset{*}{\underset{S^{-1}}{\longrightarrow}}\big(\pi^{-1}(L)\big) \cap N^*$$

which is regular whenever L is regular. □

Note that Proposition 2.3, when restricted to recognizable rewriting systems, is a corollary of Proposition 3.3. Indeed for any $R \subseteq (N \cup \{\varepsilon\}) \times N^*$, the derivation $\longrightarrow_R^*$ is equal to $\hookrightarrow_R^*$. Also note that inverse preservation properties do not hold in general. For instance when $R = \{(a, bab)\}$, we have $\hookrightarrow_R^*(a) = \{b^n a b^n \mid n \geq 0\}$ hence $\hookrightarrow_R^*$ does not preserve regularity. Conversely $\longrightarrow_{N\downarrow^{-1}}^* \;=\; \hookrightarrow_{N\downarrow^{-1}}^*$ hence $\big(\hookrightarrow_{N\downarrow^{-1}}^*\big)^{-1} = \longrightarrow_{N\downarrow}^*$ which does not preserve context-freeness.

We conclude this section on left-to-right derivation by showing that the left-to-right derivation of any rewriting system can be described using prefix derivation. To any $R \subseteq N^* \times N^*$, we associate the labelled transition system (*i.e.* labelled graph) $\widehat{R}$ over $N \cup \{\varepsilon\}$ defined as

$$\widehat{R} \;=\; \{u \xrightarrow{\varepsilon} v \mid u\,R\,v\} \cup \{a \xrightarrow{a} \varepsilon \mid a \in N\}$$

and its *prefix transition graph*

$$\widehat{R}.N^* \;=\; \{uw \xrightarrow{\varepsilon} vw \mid u\,R\,v \wedge w \in N^*\} \cup \{aw \xrightarrow{a} w \mid a \in N \wedge w \in N^*\}.$$

The words obtained by left-to-right derivation by R from a word u are precisely the words v labelling paths $u \Longrightarrow^{v}_{\widehat{R}.N^*} \varepsilon$ (in other words *recognized by* $\widehat{R}.N^*$) from vertex u to vertex ε.

Lemma 3.4. *For any system R, $u \hookrightarrow_R^* v \iff u \Longrightarrow_{\widehat{R}.N^*}^{v} \varepsilon$.*

By Proposition 3.3 and Lemma 3.4, we get $\hookrightarrow_R^*(L) = L(\widehat{R}.N^*, L, \varepsilon)$, the language of words labelling paths in the graph $\widehat{R}.N^*$ between vertices in L and the vertex ε. This is a context-free language whenever L is context-free and R is recognizable [9]. When R is finite and taking a new symbol p representing a *control state*, the system $p\widehat{R} = \{pu \longrightarrow^{\varepsilon} pv \mid u\,R\,v\} \cup \{pa \longrightarrow^{a} p \mid a \in N\}$ can be seen as a pushdown automaton with stack alphabet N (as customary, pushdown rules can be straightforwardly obtained by adding new states and rules). We have just described the effective construction of a pushdown automaton recognizing the language $\hookrightarrow_R^*(L)$ by empty stack and with possible initial stack content each word in L.

3.3 Tagged and Tag-Adding Systems

We will now apply Theorem 2.10 to the derivation of systems generalizing bifix derivation. We consider a finite set M of special symbols called tags. Bifix systems can be easily simulated and extended by adding tags at the first and last position of the sides of the rules, enforcing that the first and/or last tags of each side of a rule must be the same [1].

Definition 3.5. *Given disjoint sets M and N, a tagged bifix system over $M \cup N$ is a system*

$$R \subseteq (M \cup \{\varepsilon\})N^*(M \cup \{\varepsilon\}) \times (M \cup N)^*$$

such that for any rule $(u, v) \in R$, $(u(1) \in M \vee u(|u|) \in M)$ and

$$u(1) \in M \implies u(1) = v(1) \quad \text{and} \quad u(|u|) \in M \implies u(|u|) = v(|v|).$$

In such a system all tags are preserved by the rewriting process. Without this condition, we could transform any finite system R over N into the tagged system

$$\begin{aligned} R_\bullet = {} & \{(\bullet u, \bullet v) \mid u\,R\,v\} \cup \{(\bullet a, \#_a) \mid a \in N\} \cup \{(\&_a, \bullet a) \mid a \in N\} \\ & \cup \{(\#_a, a\bullet) \mid a \in N\} \cup \{(a\bullet, \&_a) \mid a \in N\} \end{aligned}$$

with $M = \{\#_a \mid a \in N\} \cup \{\&_a \mid a \in N\} \cup \{\bullet\}$. We have $u \longrightarrow_R^* v \iff \bullet u \longrightarrow_{R_\bullet}^* \bullet v$ for any $u, v \in N^*$. Since $\longrightarrow_R^*$ is not recursive in general, neither is $\longrightarrow_{R_\bullet}^*$. The rationality of bifix derivation can be extended to the derivation of tagged bifix systems.

Proposition 3.6 ([1]). *The derivation relation of any recognizable tagged bifix system is rational.*

Before further extending this class of systems by allowing tag-adding and infix rules, we consider systems whose tag set M is partitioned into a subset M_p of *prefix tags* and a subset M_s of *suffix tags*: $M_p \cup M_s = M$ and $M_p \cap M_s = \emptyset$.

Definition 3.7. *A tag-adding prefix/suffix system is a system*

$$R \subseteq (M_p \cup \{\varepsilon\})N^*(M_s \cup \{\varepsilon\})\times(M \cup N)^*$$

such that for any rule $(u, v) \in R$,

$$\big(u(1) \in M_p \implies v(1) \in M_p\big) \quad \textit{and} \quad \big(u(|u|) \in M_s \implies v(|v|) \in M_s\big).$$

Considering tags $\#, \& \in M$ and a letter $a \in N$, the two-rule system $R = \{(\#a, \&), (\&, a\#)\}$ cannot be a tag-adding prefix/suffix system since the tag $\#$ would be simultaneaously prefix and suffix. For this system, neither the direct nor the inverse derivation preserve regularity:

$$\xrightarrow[R]{*}\big((\#a)^*\big) \cap a^*\#^* = \{a^n\#^n \mid n \geq 0\} \text{ and } \xrightarrow[R^{-1}]{*}\big((\#a)^*\big) \cap \#^*a^* = \{\#^n a^n \mid n \geq 0\}.$$

We say that a tag-adding prefix/suffix system R is *context-free* if $R \cap N^*\times N^*$ is a context-free system and $R - N^*\times N^*$ is recognizable. Let us apply Theorem 2.10.

Proposition 3.8. *Any context-free tag-adding prefix/suffix system is cf/reg-preserving.*

Proof. Let R be a context-free tag-adding prefix/suffix system. The sets M_p and M_s are respectively prefix and suffix sub-alphabets of R^{-1}. Thus R^{-1} is M_pM_s-decomposable into $(R \cap N^*\times N^*)^{-1}$ and by Proposition 2.11, R is cf/reg-preserving. □

A particular case of a tag-adding prefix system ($M_s = \emptyset$) is given by a recognizable system $R \subseteq MN^*\times(MN^*)^+$ which is cf/reg-preserving by Proposition 3.8, and also by Proposition 3.3: the derivation is equal to its left-to-right derivation. These particular systems generalize the first model of dynamic networks of pushdown systems [7].

We will now use Proposition 3.8 to obtain the same closure properties for the following extension of tagged bifix systems.

Definition 3.9. *A tag-adding bifix system is a system*

$$R \subseteq (M \cup \{\varepsilon\})N^*(M \cup \{\varepsilon\})\times(M \cup N)^*$$

such that for any rule $(u, v) \in R$,

$$\big(u(1) \in M \implies u(1) = v(1)\big) \quad \textit{and} \quad \big(u(|u|) \in M \implies u(|u|) = v(|v|)\big).$$

Note that the previous system $\{(\#a, \&), (\&, a\#)\}$ is not tag-adding bifix. Proposition 3.8 remains valid for such systems when they are *context-free*, *i.e.* when $R \cap N^*\times N^*$ is a context-free system and $R - N^*\times N^*$ is recognizable.

Theorem 3.10. *Any context-free tag-adding bifix system is cf/reg-preserving.*

Note that Proposition 3.8 and Theorem 3.10 can both be generalized to the systems R such that $R - N^*\times N^*$ is recognizable and $(R \cap N^*\times N^*) \cup D^{-1}$ is cf/reg-preserving.

Theorem 3.10 generalizes Theorem 7 of [1]: for any 'tagged infix system with tag removing rules' R, its inverse R^{-1} is a particular tag-adding bifix system hence $\longrightarrow_R^*$ preserves regularity. As a corollary, Theorem 3.10 also positively answers a conjecture stated in [2] (page 99).

4 Conclusion

In this paper we presented a decomposition mechanism for word rewriting systems, allowing us to transfer the simultaneous regularity and inverse context-freeness preservation of the Dyck system to several classes of rewriting systems.

We are currently investigating more general criteria to widen the scope of this result and to extend this decomposition to terms.

Many thanks to Antoine Meyer for helping us make this paper readable, and to anonymous referees for helpful comments.

References

1. Altenbernd, J.: On bifix systems and generalizations. In: Martín-Vide, C., Otto, F., Fernau, H. (eds.) LATA 2008. LNCS, vol. 5196, pp. 40–51. Springer, Heidelberg (2008)
2. Altenbernd, J.: Reachability over word rewriting systems. Ph.D. Thesis, RWTH Aachen, Germany (2009)
3. Benois, M.: Parties rationnelles du groupe libre. C.R. Académie des Sciences, Série A 269, 1188–1190 (1969)
4. Berstel, J.: Transductions and context-free languages. Teubner, Stuttgart (1979)
5. Book, R., Jantzen, M., Wrathall, C.: Monadic thue systems. Theoretical Computer Science 19, 231–251 (1982)
6. Book, R., Otto, F.: String-rewriting systems. Texts and Monographs in Computer Science. Springer, Heidelberg (1993)
7. Bouajjani, A., Müller-Olm, M., Touili, T.: Regular symbolic analysis of dynamic networks of pushdown systems. In: Abadi, M., de Alfaro, L. (eds.) CONCUR 2005. LNCS, vol. 3653, pp. 473–487. Springer, Heidelberg (2005)
8. Büchi, R.: Regular canonical systems. Archiv für Mathematische Logik und Grundlagenforschung 6, 91–111 (1964)
9. Caucal, D.: On the regular structure of prefix rewriting. Theoretical Computer Science 106, 61–86 (1992); originally published In: Arnold, A. (ed.) CAAP 1990. LNCS, vol. 431, pp. 61–86. Springer, Heidelberg (1990)
10. Endrullis, J., Hofbauer, D., Waldmann, J.: Decomposing terminating rewrite relations. In: Geser, A., Sondergaard, H. (eds.) Proc. 8^{th} WST, pp. 39–43 (2006), `http://www.acm.org/corr/`, Computing Research Repository
11. Geser, A., Hofbauer, D., Waldmann, J.: Match-bounded string rewriting systems. Applicable Algebra in Engineering, Communication and Computing 15, 149–171 (2004)
12. Hibbard, T.: Context-limited grammars. JACM 21(3), 446–453 (1974)
13. Hofbauer, D., Waldmann, J.: Deleting string rewriting systems preserve regularity. Theoretical Computer Science 327, 301–317 (2004); originally published In: Ésik, Z., Fülöp, Z. (eds.) DLT 2003. LNCS, vol. 2710, pp. 301–317. Springer, Heidelberg (2003)
14. Jantzen, M., Kudlek, M., Lange, K.J., Petersen, H.: Dyck_1-reductions of context-free languages. In: Budach, L., Bakharajev, R., Lipanov, O. (eds.) FCT 1987. LNCS, vol. 278, pp. 218–227. Springer, Heidelberg (1987)
15. Karhumäki, J., Kunc, M., Okhotin, A.: Computing by commuting. Theoretical Computer Science 356, 200–211 (2006)