A String–Rewriting Characterization of Muller and Schupp's Context–Free Graphs

Hugues Calbrix[1] and Teodor Knapik[2]

[1] **College Jean Lecanuet**, BP 1024
76171 Rouen Cedex, France
HugCalbrix@aol.com

[2] IREMIA, Université de la Réunion, BP 7151,
97715 Saint Denis Messageries Cedex 9, Réunion
knapik@univ--reunion.fr

Abstract. This paper introduces Thue specifications, an approach for string–rewriting description of infinite graphs. It is shown that *strongly reduction–bounded* and *unitary reduction–bounded* rational Thue specifications have the same expressive power and both characterize the context–free graphs of Muller and Schupp. The problem of strong reduction–boundedness for rational Thue specifications is shown to be undecidable but the class of unitary reduction–bounded rational Thue specifications, that is a proper subclass of strongly reduction–bounded rational Thue specifications, is shown to be recursive.

1 Introduction

Countable graphs or, more precisely, transition systems can model any software or digital hardware system. For this reason at least, the study of infinite graphs is, in authors opinion, an important task. Obviously, dealing with infinite graphs requires a finite description. When viewed as systems of equations, some graph grammars [9] may have infinite graphs as solutions. In this paper another approach is introduced. The idea comes from the categorist's way of expressing equations between words (of the monoid generated by the arrows of a category) by means of commutative diagrams.[1]

By orienting them, equations between words are turned into string–rewrite rules. String–rewriting systems were introduced early in this century by Axel Thue [17] in his investigations about the word problem and are also known as semi–Thue systems. Later, semi–Thue systems became useful in formal languages theory (see [15] for an overview) and, as pointed out in this paper, are also of interest as finite descriptions of infinite graphs. Other approaches relating semi–Thue systems to infinite graphs may be found in [16], [5] and [4]. In the latter paper, the class of Muller and Schupp's context–free graphs [12] is characterized by means of prefix rewriting using labeled rewrite rules.

[1] The idea underlying the definition of the Cayley graph associated to a group presentation leads to a similar result.

V. Arvind and R. Ramanujam (Eds.): Foundation of Software Technology, LNCS 1530, pp. 331–342, 1998.

The link between infinite graphs and semi–Thue systems introduced in this paper raises the following question. Which classes of graphs may be described by semi–Thue systems ? As a first element of the answer, a string–rewriting characterization of context–free graphs of Muller and Schupp is provided as follows. Sect. 2 is devoted to basic definitions. Thue specifications and their graphs are defined in Sect. 3. Two classes of Thue specifications are described in Sect. 4 and the main result is established in Sect. 5. In Sect. 6 the authors investigate whether these classes are recursive. Short conclusion closes the paper.

2 Preliminaries

Assuming a smattering of string–rewriting and formal languages, several basic definitions and facts are reviewed in this section. An introductory material on above topics may be found in e.g. [2], [11] and [14].

Words. Given a finite set A called *alphabet*, the elements of which are called *letters*, A^* stands for the *free monoid* over A. The elements of A^* are all *words* over A, including the *empty word*, written ε. A subset of A^* is a *language* over A. Each word u is mapped to its *length*, written $|u|$, via the unique monoid homomorphism from A^* onto $(\mathbb{N}, 0, +)$ that maps each letter of A to 1. When $u = xy$ for some words x and y, then x (resp. y) is called a *prefix* (resp. *suffix*) *of* u. If in addition $y \neq \varepsilon$ (resp. $x \neq \varepsilon$), then x (resp. y) is a *proper prefix* (resp. *suffix*) *of* u. The set of suffixes of u is written $\mathrm{suff}(u)$. This notation is extended to sets in the usual way: $\mathrm{suff}(L) = \bigcup_{u \in L} \mathrm{suff}(u)$ for any language L. When $u = xyz$ for some words x, y, z (resp. such that $xz \neq \varepsilon$), then y is called a *factor* (resp. *proper factor*) *of* u. A word u *properly overlaps on left* a word v if $u = xy$ and $v = yz$ for some words x, y, z such that $x \neq \varepsilon$.

Semi–thue Systems. A semi–Thue system $\mathcal{S}$ (an *sts* for short) over A is a subset of $A^* \times A^*$. A pair (l, r) of $\mathcal{S}$ is called *(rewrite) rule*, the word l (resp. r) is its lefthand (resp. righthand) side. As any binary relation, $\mathcal{S}$ has its domain (resp. range), written $\mathrm{Dom}(\mathcal{S})$ (resp. $\mathrm{Ran}(\mathcal{S})$). Throughout this paper, only finite semi–Thue systems are considered.

The *single–step reduction relation* induced by $\mathcal{S}$ on A^*, is the binary relation $\rightarrow_{\mathcal{S}} = \{(xly, xry) \mid x, y \in A^*,\ (l, r) \in \mathcal{S}\}$. A word u *reduces* into a word v, written $u \rightarrow^*_{\mathcal{S}} v$, if there exist words $u_0, \ldots u_k$ such that $u_0 = u$, $u_k = v$ and $u_i \rightarrow_{\mathcal{S}} u_{i+1}$ for each $i = 0, \ldots, k-1$. The integer k is then the *length of the reduction* under consideration. A word v is *irreducible* with respect to (w.r.t. for short) $\mathcal{S}$ when v does not belong to $\mathrm{Dom}(\rightarrow_{\mathcal{S}})$. Otherwise v is *reducible* w.r.t. $\mathcal{S}$. It is easy to see that the set of all irreducible words w.r.t. $\mathcal{S}$, written $\mathrm{Irr}(\mathcal{S})$, is rational whenever $\mathrm{Dom}(\mathcal{S})$ is so, since $\mathrm{Dom}(\rightarrow_{\mathcal{S}}) = A^*\big(\mathrm{Dom}(\mathcal{S})\big)A^*$. A word v is a *normal form* of a word u, when v is irreducible and $u \rightarrow^*_{\mathcal{S}} v$.

An sts $\mathcal{S}$ is *terminating*, if there is no infinite chain $u_0 \rightarrow_{\mathcal{S}} u_1 \rightarrow_{\mathcal{S}} u_2 \rightarrow_{\mathcal{S}} \cdots$. An sts is *left–basic*, if no righthand side of a rule is a proper factor of a lefthand side nor is properly overlapped on left by a lefthand side. An sts $\mathcal{S}$ over A is *special* (resp. *monadic*), if $\mathrm{Ran}(\mathcal{S}) = \{\varepsilon\}$ (resp. $\mathrm{Ran}(\mathcal{S}) \subseteq A \cup \{\varepsilon\}$ and $|l| > |r|$ for each $(l, r) \in \mathcal{S}$).

Graphs. Given an alphabet A, a *simple directed edge–labeled graph G over A* is a set of *edges*, viz a subset of $D \times A \times D$ where D is an arbitrary set. Given $d, d' \in D$, an edge from d to d' labeled by $a \in A$ is written $d \xrightarrow{a} d'$. A (finite) *path* in G from some $d \in D$ to some $d' \in D$ is a sequence of edges of the following form: $d_0 \xrightarrow{a_1} d_1, \ldots, d_{n-1} \xrightarrow{a_n} d_n$, such that $d_0 = d$ and $d_n = d'$.

For the purpose of this paper, the interests lies merely in graphs, the vertices of which are all accessible from some distinguished vertex. Thus, a graph $G \subseteq D \times A \times D$ is said to be *rooted on a vertex* $e \in D$, if there exists a path from e to each vertex of G. The following assumption is made for the sequel. Whenever in a definition of a graph a vertex e is distinguished as root, then the maximal subgraph rooted on e is understood.

Pushdown Machines and Context–Free Graphs. An important class of graphs with decidable monadic second–order theory is characterized in [12]. The graphs of this class are called *context–free* by Muller and Schupp and may be defined by means of pushdown machines.

A *pushdown machine*[2] over A (a *pdm* for short) is a triple $\mathrm{P} = (Q, Z, T)$ where Q is the set of *states*, Z is the *stack alphabet* and T is a finite subset of $A \cup \{\varepsilon\} \times Q \times Z \times Z^* \times Q$, called the set of *transition rules*. A is the *input alphabet*. A pdm P is *realtime* when T is a finite subset of $A \times Q \times Z \times Z^* \times Q$.

An *internal configuration* of a pdm P is a pair $(q, h) \in Q \times Z^*$. To any pdm P together with an internal configuration ι, one may associate an edge–labeled oriented graph $\mathrm{G}(\mathrm{P}, \iota)$ defined as follows. The vertices of the graph are all internal configurations accessible from the configuration ι. The latter one is the root of the graph. There is an edge labeled by $c \in A \cup \{\varepsilon\}$ from (q_1, h_1) to (q_2, h_2) whenever there exists a letter $z \in Z$ and two words $g_1, g_2 \in Z^*$ such that $h_1 = g_1 z$, $h_2 = g_1 g_2$ and $(c, q_1, z, g_2, q_2) \in T$.

It may be useful to note that the context–free graphs of Muller and Schupp are exactly (up to isomorphism) all HR–equational (in the sense of [6], also called regular in [4]) as well as VR–equational (in the sense of [1]) graphs of finite degree. Finally, since any pdm over A is a realtime pdm over $A \cup \{\varepsilon\}$, realtime pdm's are as powerful as pdm's for describing graphs. In other words, the graphs of realtime pdm's form a complete set of representatives of Muller and Schupp's context–free graphs.

3 Thue Specifications and Their Graphs

The key ideas of this paper are introduced in the present section.

[2] It is a pushdown automaton with no specified accepting configurations.

Definition 1. *An (oriented) Thue specification (an ots for short) over an alphabet A is a triple $\langle \mathcal{S}, L, u\rangle$ where $\mathcal{S}$ is a semi–Thue system over A, L is a subset of* $\mathrm{Irr}(\mathcal{S})$ *and u is a word of L. An ots is rational if L is so.*

The semantics of a Thue specification may be defined as a category of models similarly to [10] but this topic is not developed in the sequel. In fact, for the purpose of this paper only the initial model, referred to in the following as *the model*, is relevant. Without summoning up the initial semantics, it may be simply defined as follows.

Definition 2. *The model of an ots $\langle \mathcal{S}, L, u\rangle$ is the graph, written* $\mathrm{G}(\mathcal{S}, L, u)$, *defined as follows. The vertices of the graph are all words of L that are accessible via edges from the root u of the graph. The edges of* $\mathrm{G}(\mathcal{S}, L, u)$ *are labeled by the letters of A. There is an edge $v \xrightarrow{a} w$ whenever w is a normal form of va.*

It should be noted that termination of $\mathcal{S}$ is not required in this definition. Thus a vertex v of $\mathrm{G}(\mathcal{S}, L, u)$ has no outgoing edge labeled by a if and only if no normal form of va belongs to L.

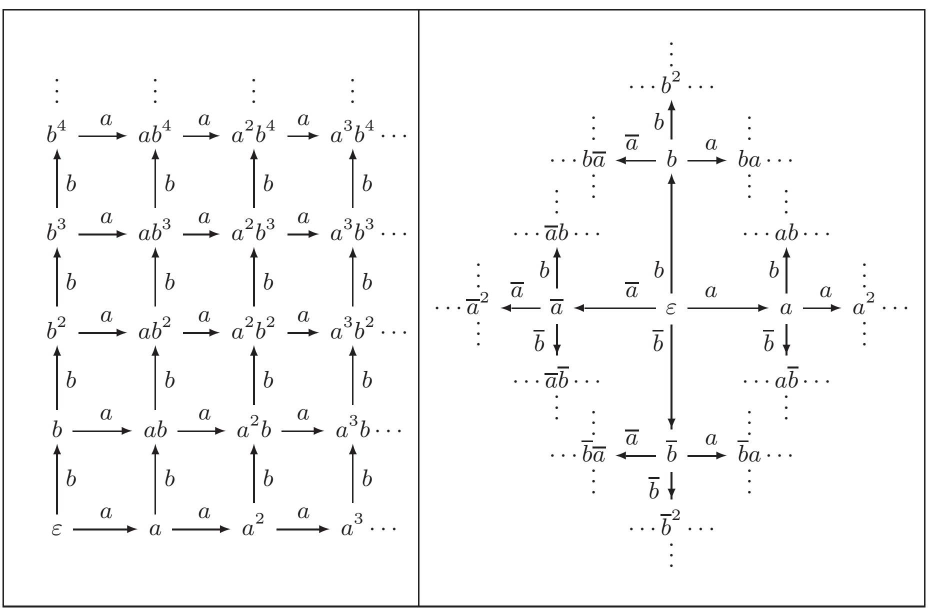

Fig. 1. Graphs $\mathrm{G}(\mathcal{S}_1, \mathrm{Irr}(\mathcal{S}_1), \varepsilon)$ and $\mathrm{G}(\mathcal{S}_2, \mathrm{Irr}(\mathcal{S}_2), \varepsilon)$

Example 1. Over the alphabet $A_1 = \{a, b\}$, consider a single–rule sts $\mathcal{S}_1 = \{(ba, ab)\}$. The set of irreducible words is a^*b^*. The graph $\mathrm{G}(\mathcal{S}_1, \mathrm{Irr}(\mathcal{S}_1), \varepsilon)$ (see Fig. 1) is isomorphic to $\omega \times \omega$.

Example 2. Over the alphabet $A_2 = \{a, b, \overline{a}, \overline{b}\}$ consider the following sts: $\mathcal{S}_2 = \{(a\overline{a}, \varepsilon), (\overline{a}a, \varepsilon), (b\overline{b}, \varepsilon), (\overline{b}b, \varepsilon)\}$. The set of irreducible words w.r.t. $\mathcal{S}_2$ is the set of all reduced words representing the elements of the free group generated by $\{a, b\}$. The graph[3] $\mathrm{G}(\mathcal{S}_2, \mathrm{Irr}(\mathcal{S}_2), \varepsilon)$ (see Fig. 1) is isomorphic to the Cayley graph of the two–generator free group.

4 Two Equivalent Conditions

This section introduces two notions that help characterizing Thue specifications that generate the context–free graphs of Muller and Schupp.

An sts $\mathcal{S}$ over A is *strongly reduction–bounded* on a subset L of $\mathrm{Irr}(\mathcal{S})$ when there exists a positive integer k such that for each word u of L and each a in A, the length of any reduction of ua is less than k. The integer k is then called a *reduction bound* of $\mathcal{S}$ (on L). An sts $\mathcal{S}$ is *unitary reduction–bounded* on a subset L of $\mathrm{Irr}(\mathcal{S})$ when it is strongly reduction–bounded on L and 1 is a reduction bound of $\mathcal{S}$.

An ots $\langle \mathcal{S}, L, u \rangle$ is *strongly reduction–bounded* (resp. *unitary reduction–bounded*) if $\mathcal{S}$ is strongly reduction–bounded (resp. unitary reduction–bounded) on L.

The system $\mathcal{S}_1$ from Example 1 is not strongly reduction–bounded on $\mathrm{Irr}(\mathcal{S}_1)$. Indeed, the word b^n is irreducible and n is the length of the reduction of the word $b^n a$ into its normal form ab^n. Since n is an arbitrary positive integer, $\mathcal{S}_1$ has no reduction bound.

On the contrary, the sts $\mathcal{S}_2$ from Example 2 is unitary reduction–bounded on $\mathrm{Irr}(\mathcal{S}_2)$. As a matter of fact, given a nonempty word $w = uc$ of $\mathrm{Irr}(\mathcal{S}_2)$ with $c \in A_2$, for any $d \in A_2$ the normal form of wd is u if $c = \overline{d}$ or $d = \overline{c}$, and wd otherwise. In both cases the length of the corresponding reduction is not greater than 1. It may be observed that $\mathrm{G}(\mathcal{S}_2, \mathrm{Irr}(\mathcal{S}_2), \varepsilon)$ is a tree.

The next proposition and the subsequent comment establish inclusions between some familiar classes of semi–Thue systems (see e.g. [2] and [15]) on one hand, and strongly (resp. unitary) reduction–bounded semi–Thue systems on the other hand.

Proposition 1. *For all finite semi–Thue systems $\mathcal{S}$ the following holds.*

1. *If $\mathcal{S}$ is special then $\mathcal{S}$ is unitary reduction–bounded.*
2. *If $\mathrm{Ran}(\mathcal{S}) \subseteq \mathrm{Irr}(\mathcal{S})$ and no word of $\mathrm{Ran}(\mathcal{S})$ is properly overlapped on left by a word of $\mathrm{Dom}(\mathcal{S})$ then $\mathcal{S}$ is unitary reduction–bounded.*
3. *If $\mathcal{S}$ is terminating left–basic then $\mathcal{S}$ is strongly reduction–bounded.*

Proof. Both (1) and (2) are obvious. Let then $\mathcal{S}$ be a finite terminating left–basic semi–Thue system over A. Let $w \in \mathrm{Irr}(\mathcal{S})$ and $a \in A$. Consider the longest suffix v of w such that $va \in \mathrm{Dom}(\mathcal{S})$. Observe that, since $\mathcal{S}$ is left–basic, each reduction of wa concerns only va. Let therefore k be the maximum of the lengths of all reductions of the words of $\mathrm{Dom}(\mathcal{S})$. Obviously, k is a reduction bound of $\mathcal{S}$. □

[3] For each edge, there is the opposite edge (not depicted) corresponding to the formal inverse of the label.

It may be noted that no converse of statements (1), (2) or (3) of the above proposition holds. Indeed, the semi–Thue system $\{aa \to a\}$ over $\{a\}$ is unitary reduction–bounded but is not special; a is properly overlapped on left by aa hence the system is not left–basic. On the other hand, Proposition 1 cannot be strengthened to the case of monadic semi–Thue systems. The semi–Thue system $\{ab \to b\}$ over $\{a, b\}$ is monadic without being strongly reduction–bounded.

The following result demonstrates that the strongly reduction–bounded rational ots and unitary reduction–bounded rational ots have the same expressive power for describing graphs.

Proposition 2. *Given any strongly reduction–bounded rational ots $\langle \mathcal{S}, R, u\rangle$, one may construct a unitary reduction–bounded rational ots $\langle \mathcal{S}', R', u'\rangle$ such that the graphs $\mathrm{G}(\mathcal{S}, R, u)$ and $\mathrm{G}(\mathcal{S}', R', u')$ are isomorphic.*

Proof. Let k be a reduction bound of $\mathcal{S}$ and $m = \max_{l\in \mathrm{Dom}(\mathcal{S})} |l|$ be the maximum length of the lefthand sides of $\mathcal{S}$. Consider any reduction of length n of a word wa such that $w \in R$ and $a \in A$. If $|wa| > mn$, then only a proper suffix of wa is reduced. The length of the reduced suffix cannot exceed mn. Since $n \le k$, for any reduction, the length of the reduced suffix cannot exceed mk.

Let $\mathcal{S}' = \mathcal{S}'_\# \cup \mathcal{S}'_A$ be an sts over $A \cup \{\#\}$, where $\# \notin A$, defined as follows:

$$\mathcal{S}'_\# = \{(\#wa, \#v) \mid a \in A,\ w \in R\,, v \in \mathrm{Irr}(\mathcal{S}),\ wa \xrightarrow[\mathcal{S}]{*} v \text{ and } |wa| < mk\},$$
$$\mathcal{S}'_A = \{(wa, v) \mid a \in A,\ w \in \mathrm{suff}(R)\,, v \in \mathrm{Irr}(\mathcal{S}),\ wa \xrightarrow[\mathcal{S}]{*} v \text{ and } |wa| = mk\}$$

and let $R' = \#R$.

Unit reduction–boundedness of $\mathcal{S}'$ on R' is established as follows. Let $w \in R'$ and $a \in A'$. If $a = \#$ then $wa \notin R'$. Assume then that $a \in A$. Observe first that, if $wa \to_{\mathcal{S}'_\#} x$ for some $x \in (A')^*$ then $x \in \mathrm{Irr}(\mathcal{S}')$. Assume therefore by contradiction that there is a reduction $wa \to_{\mathcal{S}'_A} x \to_{\mathcal{S}'} y$ for some $x, y \in (A')^*$. Let w_1 be the longest prefix of w this reduction is not concerned with and let w_2 be the remaining suffix, viz $w = w_1w_2$. Now, either $w_2 \in \mathrm{suff}(R)$ (when $x \to_{\mathcal{S}'_A} y$) and $|w_2| > mk$ or $w_2 = \#w'_2$ (when $x \to_{\mathcal{S}'_\#} y$) for some $w'_2 \in R$ such that $|w'_2| > mk$. Hence, there exists a reduction of w_2a (resp. w'_2a) w.r.t. $\mathcal{S}$ the length of which exceeds k. This contradicts the assumption that k is a reduction bound of $\mathcal{S}$ on R.

Observe that for all $w \in R$, $v \in \mathrm{Irr}(\mathcal{S})$ and $a \in A$, v is a normal form of wa w.r.t. $\mathcal{S}$ if and only if $\#v$ is a normal form of $\#wa$ w.r.t. $\mathcal{S}'$. Hence, the mapping $w \mapsto \#w$ restricted to the vertices of $\mathrm{G}(\mathcal{S}, R, u)$ extends to a graph isomorphism between $\mathrm{G}(\mathcal{S}, R, u)$ and $\mathrm{G}(\mathcal{S}', R', u')$ where $u' = \#u$. □

5 Main Result

Proposition 2 together with the statements of this section lead to the main result of this paper.

Proposition 3. *Given any realtime pdm* P *over* A *and an internal configuration* ι *of* P*, one may construct a unitary reduction–bounded rational ots* $\langle \mathcal{S}, R, u\rangle$ *such that the graphs* $\mathrm{G}(\mathrm{P}, \iota)$ *and* $\mathrm{G}(\mathcal{S}, R, u)$ *are isomorphic.*

Proof. Let $\mathrm{P} = (Q, Z, T)$ be a realtime pdm over A and let $\iota = (q_0, h_0)$ be an internal configuration of P. Without loss of generality A, Q and Z may be assumed pairwise disjoint. Set $A' = A \cup Q \cup Z$. Define an sts $\mathcal{S}$ over A' as follows

$$\mathcal{S} = \{(zqa, hq') \mid (a, q, z, h, q') \in T\}$$

and let $u = h_0 q_0$. It is well known that the pushdown store language of a pdm is rational. The following language is therefore rational:

$$R = \{hq \mid (q, h) \text{ is an internal configuration of P accessible from } \iota\} \ .$$

Moreover $R \subseteq \mathrm{Irr}(\mathcal{S})$.

Observe that $\mathcal{S}$ is unitary reduction–bounded on R. Indeed, let $v \in R$ and $a \in A'$. For va to be reducible, there must exist $w \in \mathrm{Irr}(\mathcal{S})$ and a rewrite rule (zqa, hq') such that $v = wzqa$. Consequently, $va \to_{\mathcal{S}} whq'$. But no word of $\mathrm{Dom}(\mathcal{S})$ may overlap hq' on left. Since w is irreducible, so is whq'.

The fact that is $\mathrm{G}(\mathrm{P}, \iota)$ and $\mathrm{G}(\mathcal{S}, R, u)$ are isomorphic is readily established by induction on the distance of vertex from the root, using the following one–to–one mapping of the vertices of $\mathrm{G}(\mathrm{P}, \iota)$ onto the vertices of $\mathrm{G}(\mathcal{S}, R, u)$: $(q, h) \mapsto hq$. □

In order to establish the converse, the following lemma is useful.

Lemma 1. *Given a pdm* $\mathrm{P} = (Q, Z, T)$ *over* A*, an internal configuration* ι *of* P *and a rational subset* R *of* Z^*Q*, one may construct a pdm* P' *together with* ι' *such that the graph* $\mathrm{G}(\mathrm{P}', \iota')$ *is isomorphic to the restriction of* $\mathrm{G}(\mathrm{P}, \iota)$*, rooted on* ι*, to the following set of vertices:* $\{(q, h) \in Q \times Z^* \mid hq \in R\}$*. Moreover* P' *is realtime if* P *is so.*

The following proof uses a device similar to a predicting automaton of [11]. More precisely, P' is constructed so that the contents of its pushdown store encodes a successful run of a finite automaton accepting R.

Proof. Let $\mathrm{A} = (D, d_0, \delta, F)$ be a finite deterministic and complete automaton over A that accepts R. Here D is the set of states of A, $d_0 \in D$ is the initial state, $\delta\colon D \times A \to D$ is the transition function and $F \subseteq D$ is the set of final states of A.

A is the input alphabet of P' and the stack alphabet is $Z' = D \times Z$. Consider a map $\kappa\colon D \times Z^* \to (D \times Z)^*$ defined as follows

$$\begin{aligned} \kappa(d, \varepsilon) &= \varepsilon, && \text{for all } d \in D,\\ \kappa(d, zg) &= \langle d, z\rangle \kappa(\delta(d, z), g), && \text{for all } \langle d, z\rangle \in D \times Z \text{ and } g \in Z^*. \end{aligned}$$

Let $\iota' = (d_0, \kappa(d_0, h_0))$. On the whole $\mathrm{P}' = (Q, Z', T')$ where

$$T' = \{(a, q, \langle d, z\rangle, \kappa(d, h), q') \mid (a, q, z, h, q') \in T,\ d \in D,\ \delta(d, zq) \in F,\\ \delta(d, hq') \in F\} \ .$$

Observe that an edge $(q, gz) \xrightarrow{a} (q', gh)$ is in the restriction of $\mathrm{G}(\mathrm{P}, \iota)$ to $\{(q,h) \mid hq \in R\}$ rooted on ι if and only if the vertex (q, gz) is in this restriction and

$$\begin{aligned}&\kappa(d_0, gz) = H\langle d, z\rangle \quad \text{for some } H \in (Z')^* \text{ and some } d \in D,\\ &\delta(d, zq) \in F,\ \delta(d, hq') \in F \text{ and}\\ &(a, q, z, h, q') \in T\ .\end{aligned}$$

Hence equivalently, there is an edge $(q, \kappa(d_0, g)\langle d, z\rangle) \xrightarrow{a} (q', \kappa(d_0, g)\kappa(d, h))$ in $\mathrm{G}(\mathrm{P}', \iota')$. Since $\iota' = (d_0, \kappa(d_0, h_0))$, the result follows by induction from the above. □

It may be noted that the size of the resulting pdm P′ is in $O(nk)$, where n is the number of states of the finite automaton A accepting R used in the construction and k is the size of P. Indeed $|Z'| = n|Z|$ and $|T'| \leq n|T|$.

The converse of Proposition 3 is stated in the following.

Proposition 4. *Given any unitary reduction–bounded rational ots* $\langle \mathcal{S}, R, u\rangle$ *over* A, *one may construct a realtime pdm* P *and an internal configuration* ι *of* P *such that the graphs* $\mathrm{G}(\mathcal{S}, R, u)$ *and* $\mathrm{G}(\mathrm{P}, \iota)$ *are isomorphic.*

In the following proof a word $w \in \mathrm{Irr}(\mathcal{S})$ is cut into factors $w = x_1 \ldots x_n y$, so that all words x_i are of equal length m and $|y| < m$, where m is the maximum length of the lefthand sides of $\mathcal{S}$. Then w is encoded by an internal configuration $(q_y, z_1 \ldots z_n)$ where $z_1, \ldots, z_n$ are letters of the stack alphabet that is in one–to–one correspondence with the words over A of length m.

Proof. A pushdown machine $\mathrm{P}' = (Q', Z', T')$ is defined first together with an internal configuration ι' so that $\mathrm{G}(\mathrm{P}', \iota')$ is isomorphic to $\mathrm{G}(\mathcal{S}, \mathrm{Irr}(\mathcal{S}), u)$. Set $m = \max_{l \in \mathrm{Dom}(\mathcal{S})} |l|$. Define a pdm P′ as follows. The set Q of the states is indexed by irreducible words, the length of which is strictly less than m viz $Q = \{q_w \mid w \in \mathrm{Irr}(\mathcal{S}) \text{ and } |w| < m\}$. The stack alphabet $Z' = Z'' \cup \{z_0\}$ has the bottom symbol $z_0 \notin Z''$ and Z'' is an arbitrary set that is in one–to–one correspondence f with the set of irreducible words of length m,

$$f \colon \{w \in \mathrm{Irr}(\mathcal{S}) \mid |w| = m\} \to Z''\ .$$

The set T' of transition rules of P′ is constructed as follows.

- For any $a \in A$, any $q_w \in Q$ and any $z \in Z''$, one has
  1. $(a, q_w, z, z, q_{wa}) \in T$ when $f^{-1}(z)wa \in \mathrm{Irr}(\mathcal{S})$ and $|wa| < m$,
  2. $(a, q_w, z, zf(wa), q_\varepsilon) \in T$ when $f^{-1}(z)wa \in \mathrm{Irr}(\mathcal{S})$ and $|wa| = m$,
  3. $(a, q_w, z, \varepsilon, q_{vr}) \in T$ when $f^{-1}(z)wa = vl$ for some $v \in \mathrm{Irr}(\mathcal{S})$ and $(l, r) \in \mathcal{S}$ such that $|vr| < m$,
  4. $(a, q_w, z, f(x_1) \ldots f(x_n), q_y) \in T$ when $f^{-1}(z)wa = vl$ for some $v \in \mathrm{Irr}(\mathcal{S})$ and $(l, r) \in \mathcal{S}$ such that $vr = x_1 \ldots x_n y$, where $x_1, \ldots, x_n, y \in \mathrm{Irr}(\mathcal{S})$ are such that $|x_1| = \cdots = |x_n| = m$ and $|y| < m$.
- For any $a \in A$ and any $q_w \in Q$, one has
  1. $(a, q_w, z_0, z_0, q_{wa}) \in T$ when $wa \in \mathrm{Irr}(\mathcal{S})$ and $|wa| < m$,
  2. $(a, q_w, z_0, z_0 f(wa), q_\varepsilon) \in T$ when $wa \in \mathrm{Irr}(\mathcal{S})$ and $|wa| = m$,

3. $(a, q_w, z_0, z_0, q_{vr}) \in T$ when $wa = vl$ for some $v \in \mathrm{Irr}(\mathcal{S})$ and $(l, r) \in \mathcal{S}$ such that $|vr| < m$,
4. $(a, q_w, z_0, z_0 f(x_1) \dots f(x_n), q_y) \in T$ when $wa = vl$ for some $v \in \mathrm{Irr}(\mathcal{S})$ and $(l, r) \in \mathcal{S}$ such that $vr = x_1 \dots x_n y$, where $x_1, \dots, x_n, y \in \mathrm{Irr}(\mathcal{S})$ are such that $|x_1| = \dots = |x_n| = m$ and $|y| < m$.

Define now the internal configuration ι' of P' corresponding to the root of the graph $\mathrm{G}(\mathcal{S}, \mathrm{Irr}(\mathcal{S}), u)$ as follows. If $|u| < m$, set $\iota' = (q_u, z_0)$. Otherwise one has $u = x_1 \dots x_n y$ for some $x_1, \dots, x_n, y \in \mathrm{Irr}(\mathcal{S})$ such that $|x_1| = \dots = |x_n| = m$ and $|y| < m$. Set then $\iota' = (q_y, z_0 f(x_1) \dots f(x_n))$.

It is easy to check that the one–to–one mapping $(q_w, z_0 h) \mapsto f^{-1}(h)w$ of the vertices of $\mathrm{G}(\mathrm{P}', \iota')$ onto the vertices of $\mathrm{G}(\mathcal{S}, \mathrm{Irr}(\mathcal{S}), u)$ extends to a graph isomorphism. Moreover P' is realtime.

Obviously, $\mathrm{G}(\mathcal{S}, R, u)$ is a restriction (on vertices) of $\mathrm{G}(\mathcal{S}, \mathrm{Irr}(\mathcal{S}), u)$ to R rooted on u. Define $\mathcal{C}_{R'} = \{(q_w, z_0 h) \mid h \in (Z'')^*,\ q_w \in Q',\ f(h)w \in R\}$ and $R' = \{z_0 h q_w \mid (q_w, z_0 h) \in \mathcal{C}_{R'}\}$. Observe that R' is rational. Moreover, the restriction of $\mathrm{G}(\mathrm{P}', \iota')$ to $\mathcal{C}_{R'}$ rooted on ι' is isomorphic to $\mathrm{G}(\mathcal{S}, R, u)$. Now, according to Lemma 1, one may construct a pdm P and an internal configuration ι of P such that the graph $\mathrm{G}(\mathrm{P}, \iota)$ is isomorphic to $\mathrm{G}(\mathcal{S}, R, u)$. $\square$

It is not too difficult to figure out that size resulting from the construction described above is doubly exponential w.r.t. the maximum length of the lefthand sides of $\mathcal{S}$.

In view of the results established so far, it is straightforward to conclude this section as follows. Both strongly reduction–bounded and unitary reduction–bounded rational ots characterize the class of Muller and Schupp's context–free graphs. Moreover, since the transition rules of a pdm may be seen as labeled prefix rewrite rules, the graphs of suffix–bounded rational ots can also be generated by prefix rewriting. This remark is rather obvious insofar as prefix rewriting generate exactly context–free graphs of Muller and Schupp [4].

6 Decision Problems

The criterion of strong reduction–boundedness defines a class of Thue specifications, the graphs of which have decidable monadic second–order theory due to the result of Muller and Schupp [12]. It may be asked whether the class of strongly reduction–bounded rational ots is recursive. The answer is positive for the subclass of unitary reduction–bounded rational ots.

Proposition 5. *There is an algorithm to solve the following problem.*
Instance: *A finite semi–Thue system $\mathcal{S}$ and a rational subset R of* $\mathrm{Irr}(\mathcal{S})$.
Question: *Is $\mathcal{S}$ unitary reduction–bounded on R ?*

Proof. Let $\mathcal{S}$ be a finite sts over A. For each rule (r, l) and each $a \in A$ set $R_{(l,r),a} = \big((Ra)l^{-1}\big)r$. Observe that

$$\bigcup_{\substack{(l,r)\in\mathcal{S}\\ a\in A}} R_{(l,r),a} = \{v \mid \exists\, u \in R, \exists\, a \in A \text{ s.t. } ua \underset{\mathcal{S}}{\rightarrow} v\} \ .$$

Thus, $\mathcal{S}$ is unitary reduction–bounded if and only if $R_{(l,r),a} \subseteq \mathrm{Irr}(\mathcal{S})$ for each $(l, r) \in \mathcal{S}$ and $a \in A$. Since both $\mathcal{S}$ and A are finite, there is a finite number of inclusions to test, all between rational languages. □

It is not surprising that the above result may be extended as follows.

Proposition 6. *There is an algorithm to solve the following problem.*
Instance: *A finite semi–Thue system $\mathcal{S}$, a rational subset R of* $\mathrm{Irr}(\mathcal{S})$ *and a positive integer k.*
Question: *Is k a reduction bound of $\mathcal{S}$ on R ?*

Proof. The proof is similar to the one of Proposition 5. One has to test the inclusion in $\mathrm{Irr}(\mathcal{S})$ of the languages of the form $((\cdots(((Ra)l_1^{-1})r_1)\dots l_k^{-1})r_k)$ for each sequence $(l_1, r_1)\dots(l_k, r_k)$ over $\mathcal{S}$ of length k and each $a \in A$. □

As established above, one may decide whether an integer is a reduction bound of a semi–Thue system. However the decision procedure sketched in the proof does not allow, in general, to establish the existence of a reduction bound. The problem, whether a reduction bound exists, may be addressed in the context of the strong boundedness problem for Turing machines.

As defined in [13], a Turing machine T is *strongly bounded* if there exists an integer k such that, for each finite configuration uqv, T halts after at most k steps when starting in configuration uqv. The *strong boundedness problem* for Turing machines is the following decision problem.

Instance: A single–tape Turing machine T.
Question: Is T strongly bounded ?

Now, one may effectively encode an arbitrary deterministic single–tape Turing machine T into a semi–Thue system $\mathcal{S}$ over an appropriate alphabet A and define an effective encoding χ of the configurations of T into words of $\mathrm{Irr}(\mathcal{S})A$ that satisfy the following property.

> Starting from uqv, T halts after k steps if and only if any reduction of $\chi(uqv)$ into an irreducible word is of length k.

Such an encoding χ may consist in the following. Consider a configuration uqv where a Turing machine is in a state q, uv is a tape inscription such that the tape head is positioned on the first letter of v or on the blank $\square$, if $v = \varepsilon$. Let $m = \max(|u|, |v|)$ and let u' (resp. v') be the suffix (resp. prefix) of $\blacksquare^m u$ (resp. $v\square^m$) of length m, where $\blacksquare$ is an additional symbol. Consider now a letterwise shuffle of u' and the reversal $\tilde{v}'$ of v'. Enclose it in a pair of $\#$ and append $\overline{q}$ on right. For instance with $u = abc$ and $v = de$ (resp. $u = ab$ and $v = cde$) we get $\#a\square becd\#\overline{q}$ (resp. $\#\blacksquare eadbc\#\overline{q}$). Formally a configuration uqv is encoded as follows:

$$\chi(uqv) = \#\big((\blacksquare^{\max(0,|v|-|u|)}u) \sqcup\!\sqcup (\square^{\max(0,|u|-|v|)}\tilde{v})\big)\#\overline{q}$$

where $\sqcup\!\sqcup$ denotes the letterwise shuffle of two words (of equal length)

$$a_1 \dots a_n \sqcup\!\sqcup b_1 \dots b_n = a_1 b_1 \dots a_n b_n \enspace .$$

It is not too difficult to simulate in such representation each move of the Turing machine by a single rewrite step w.r.t. an appropriate sts. However, one needs to distinguish whether the tape head is in the part of the tape initially occupied by u or by v. In the former situation the symbols of the states of the Turing machine are used directly whereas in the latter situation, their bared versions (as $\overline{q}$) are used. A complete many–one reduction of the strong boundedness problem for Turing machines into the strong reduction–boundedness problem for semi–Thue systems may be found in a preliminary version of this paper [3].

Now, the strong boundedness problem is undecidable for 2–symbol single–tape Turing machines (cf. Proposition 14 of [13]). This gives the following undecidability result.

Proposition 7. *There exists a rational set R for which the following problem is undecidable.*
Instance: *A finite semi–Thue system $\mathcal{S}$.*
Question: *Is $\mathcal{S}$ strongly reduction–bounded on R ?*

7 Conclusion

Thue specifications and their graphs have been introduced and two classes of Thue specifications have been defined: strongly reduction–bounded and unitary reduction–bounded ots. It has been established that both unitary and strongly reduction–bounded rational Thue specifications characterize the context–free graphs of Muller and Schupp. Moreover, the membership problem for the class of strongly reduction–bounded rational ots has been shown to be undecidable whereas, for its proper subclass of unitary reduction–bounded rational ots, this problem has been established as being decidable.

An important property of context–free graphs is the decidability of their monadic second–order theory. However the class of Muller and Schupp's context–free graphs is not the only well–known class of graphs with decidable monadic second–order theory. More general classes of such graphs are described in e.g. [1], [5], [7] or [8]. Beyond, other classes of infinite graphs, the monadic second–order theory of which is not necessarily decidable, are currently under investigation. How Thue specifications are linked via their graphs to all these classes, is considered for further research.

References

1. K. Barthelmann. On equational simple graphs. Technical Report 9/97, Johannes Gutenberg Universität, Mainz, 1997. 333, 341
2. R. V. Book and F. Otto. *String–Rewriting Systems.* Texts and Monographs in Computer Science. Springer–Verlag, 1993. 332, 335
3. H. Calbrix and T. Knapik. On a class of graphs of semi–Thue systems having decidable monadic second–order theory. Technical Report INF/1998/01/01/a, IREMIA, Université de la Réunion, 1998. 341

4. D. Caucal. On the regular structure of prefix rewriting. *Theoretical Comput. Sci.*, 106:61–86, 1992. 331, 333, 339
5. D. Caucal. On infinite transition graphs having a decidable monadic second–order theory. In F. M. auf der Heide and B. Monien, editors, *23th International Colloquium on Automata Languages and Programming*, LNCS 1099, pages 194–205, 1996. 331, 341
6. B. Courcelle. The monadic second–order logic of graphs, II: Infinite graphs of bounded width. *Mathematical System Theory*, 21:187–221, 1989. 333
7. B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Formal Models and Semantics*, volume B of *Handbook of Theoretical Computer Science*, pages 193–242. Elsevier, 1990. 341
8. B. Courcelle. The monadic second–order theory of graphs IX: Machines and their behaviours. *Theoretical Comput. Sci.*, 151:125–162, 1995. 341
9. J. Engelfriet. Contex–free graph grammars. In G. Rozenberg and A. Salomaa, editors, *Beyond Words*, volume 3 of *Handbook of Formal Languages*, pages 125–213. Springer–Verlag, 1997. 331
10. J. A. Goguen, J. W. Thatcher, and E. G. Wagner. An initial approach to the specification, correctness and implementation of abstract data types. In R. T. Yeh, editor, *Data Structuring*, volume 4 of *Current Trends in Programming Methodology*, pages 80–149. Prentice Hall, 1978. 334
11. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison–Wesley, 1979. 332, 337
12. D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata and second–order logic. *Theoretical Comput. Sci.*, 37:51–75, 1985. 331, 333, 339
13. F. Otto. On the property of preserving regularity for string–rewriting systems. In H. Comon, editor, *Rewriting Techniques and Applications*, LNCS 1232, pages 83–97, 1987. 340, 341
14. G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages*, volume 1. Springer–Verlag, 1997. 332
15. G. Sénizergues. Formal languages and word rewriting. In *Term Rewriting: French Spring School of Theoretical Computer Science*, LNCS 909, pages 75–94, Font–Romeu, May 1993. 331, 335
16. C. C. Squier, F. Otto, and Y. Kobayashi. A finiteness condition for rewriting systems. *Theoretical Comput. Sci.*, 131(2):271–294, 1994. 331
17. A. Thue. Probleme über veränderungen von zeichenreihen nach gegebenen regeln. *Skr. Vid. Kristiania, I Mat. Natuv. Klasse*, 10:34 pp, 1914. 331