Termination of Single-Threaded One-rule Semi-Thue Systems

Wojciech Moczydłowski*[1] and Alfons Geser**[2]

[1] Cornell University, Dept. of Computer Science
5162 Upson Hall, Ithaca, NY 14853, USA
wojtek@cs.cornell.edu

[2] National Institute for Aerospace (NIA)
144 Research Drive, Hampton, VA 23666, USA
geser@nianet.org

Abstract. This paper is a contribution to the long standing open problem of uniform termination of Semi-Thue Systems that consist of one rule $s \to t$. McNaughton previously showed that rules incapable of (1) deleting t completely from both sides, (2) deleting t completely from the left, and (3) deleting t completely from the right, have a decidable uniform termination problem. We use a novel approach to show that Premise (2) or, symmetrically, Premise (3), is inessential. Our approach is based on derivations in which every pair of successive steps has an overlap. We call such derivations single-threaded.

Key Words and Phrases: string rewriting, semi-Thue system, uniform termination, termination, one-rule, single-rule, single-threaded, well-behaved

1 Introduction

The decidability of the uniform termination problem of one-rule Semi-Thue Systems (1STS) has been open for 14 years. A systematic exploration of the problem was started by Kurth [5].

This problem is both a test case for the strength of termination proof methods and a trigger for their development. Remarkable progress has been made by investigating the consumption and introduction patterns in derivations [7, 8, 4].

McNaughton's notion of a well-behaved derivation is based on the idea that some rules act as if there was an invisible barrier ("inhibitor") somewhere at their right hand side. This inhibitor cannot be removed, so derivations cannot exhibit global communication through the string. McNaughton shows that it is decidable whether a rule is well-behaved, i.e. admits only well-behaved derivations. Moreover he shows that uniform termination is decidable for well-behaved rules.

* Partly supported by KBN Grant 7 T11C 028 20.

** Partly supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046.

In a well-behaved derivation the contractum introduced by any step during a derivation cannot be consumed completely. The contractum can be consumed partially from the left or from the right. We want to study non-well-behaved derivations and hence call a derivation:

– *both-sides-digestible* (BD) if the remainder of some step after partial consumption from the left and partial consumption from the right is consumed later completely;
– *left-digestible* (LD) if the remainder of some step after partial consumption from the left (without any partial consumption from the right) is consumed later completely;
– *right-digestible* (RD) if the remainder of some step after partial consumption from the right (without any partial consumption from the left) is consumed later completely.

We study the following question:

– A 1STS is obviously well-behaved iff it satisfies none of these properties. Can we decide uniform termination also if some of them are true?

An interesting special case is given when the left hand side of the rule has no self-overlap. For this self-overlap free (SOF) case, Kobayashi et al. [4] introduce derivation patterns that are less restrictive than well-behavedness and they call derivations which satisfy them *tame*, *gentle* and *simple*. They show that a gentle 1STS can be transformed to another Semi-Thue System which may have more rules. The two systems have equivalent uniform termination problems. Typically, the transformed system is more amenable to the classic termination criteria. Kobayashi et al. call the properties $\neg LD$, $\neg RD$, and their conjuntion "left very gentle", "right very gentle", and "very gentle", respectively. They show that very gentle 1STSs are gentle and that the image of a simple 1STS is a context-free grammar whence its uniform termination problem is decidable. Other examples can often be solved by a transformation and a subsequent ad hoc argument. Beyond the SOF, simple systems no decidability result is available yet.

In a straightforward way the notions of tame, gentle, and simple 1STSs are generalized to non-SOF 1STSs [2]. These properties form a hierarchy:

$$\begin{array}{ccccc} \text{very gentle} & \Rightarrow & \text{gentle} & \Rightarrow & \text{tame} \\ \Uparrow & & \Uparrow & & \\ \text{well-behaved} & \Rightarrow & \text{simple} & \Rightarrow & \neg BD \end{array}$$

It is easily verified that a 1STS is simple iff it is tame and $\neg BD$. We establish the following result:

– Uniform termination is decidable for 1STSs that satisfy $\neg BD \wedge (\neg LD \vee \neg RD)$.

We reduce the uniform termination problem of 1STSs that satisfy $\neg BD \wedge \neg RD$ to the uniform halting problem of pushdown automata which is decidable [12]. For this purpose we show that each non-terminating such 1STS has an infinite derivation where each step overlaps with the previous one. We call such derivations *single-threaded*. In this case the left and right contexts of the redex can be represented as the contents of two stacks. By $\neg RD$, the left stack is size bounded.

This class of 1STSs includes the following examples which are not covered by Kobayashi et al.: examples that are simple and non-SOF; examples that are non-simple (thus non-tame), non-SOF. On the other hand, Kobayashi et al. also cover the SOF, simple, left-digestible, right-digestible 1STSs, a class which however may be void.

Our examples are not covered by any existing automated termination criteria, except *inverse match-boundedness* [3]. Inverse match-boundedness covers all well-behaved 1STSs, but it is unknown what other classes of 1STSs it also covers.

This work is a thoroughly revised and extended version of the first author's master's thesis [9] and a Technical Report [10].

The paper is organized as follows: In Section 2, we introduce concepts important in our framework, such as chain graph and mother-in-law. In Section 3, we introduce the notion of single-threaded derivation and we derive the decidability result of uniform termination. In Section 4 we give examples of the systems to which our results apply.

2 Preliminaries

We assume familiarity of the reader with semi-Thue systems (string rewriting) [1].

A string u is called a *factor* of v, in symbols $u \sqsubseteq v$, if $v = xuy$ for some $x, y \in \Sigma^*$; a *prefix* if $v = uy$ for some $y \in \Sigma^*$; a *suffix* if $v = xu$ for some $x \in \Sigma^*$. The prefix or suffix u of v is called *proper* if $u \neq v$. The set of all proper suffixes of the word u is denoted by $\mathrm{Suf}(u)$.

The set of *overlaps* of a string u with a string v is defined by

$$\mathrm{OVL}(u, v) = \{w \in \Sigma^+ \mid u = u'w, v = wv', u'v' \neq \varepsilon, u', v' \in \Sigma^*\}$$

The *length* of a string u is denoted by $|u|$.

A *Semi-Thue System* R is a finite set of *rules* $(s, t) \in \Sigma^* \times \Sigma^*$, also written $s \to t$. The one-step *rewrite relation* $\to \subseteq \Sigma^* \times \Sigma^*$ is defined by $usv \to utv$ if $u, v \in \Sigma^*$ and $(s, t) \in R$. The factors s and t are also called the *redex* and the *contractum*, respectively. Occasionally we underline the redex and overline the contractum, as in the following two rewrite steps for the example system $ab \to ba$: $a\underline{ab} \to \underline{a\overline{b}}\overline{a} \to \overline{ba}a$. A sequence of rewrite steps is called a *derivation*. We write $\mathcal{D} : w_0 \to w_1 \to \ldots$ to denote a derivation named $\mathcal{D}$ with rewrite steps $w_0 \to w_1 \to \ldots$. A system R is called *terminating* if there is no infinite derivation $w_0 \to w_1 \to \ldots$.

We focus on one-rule Semi-Thue Systems (1STS) $\{s \to t\}$, also written $s \to t$. As $s \to t$ is non-terminating if $s \sqsubseteq t$, and terminating if $|s| \geq |t|$ and $s \neq t$, we assume throughout the paper that $s \not\sqsubseteq t$ and $|s| < |t|$. A 1STS $s \to t$ is called self-overlap free (SOF), if $\mathrm{OVL}(s,s) = \emptyset$. If $\mathrm{OVL}(t,s) = \emptyset$ or $\mathrm{OVL}(s,t) = \emptyset$, then $s \to t$ terminates [5, Criterion D]. If $\mathrm{OVL}(t,s) \cap \mathrm{OVL}(s,t) \neq \emptyset$ ("bordered rule") then the uniform termination problem of $s \to t$ is reducible to that of a non-bordered rule [2, Theorem 6.21]. We henceforth assume that $\mathrm{OVL}(t,s)$ and $\mathrm{OVL}(s,t)$ are disjoint and non-empty.

Definition 1 ([4]). *If $\alpha \in \mathrm{OVL}(t,s)$ then let s_α and t_α be defined by $s = \alpha s_\alpha$ and $t = t_\alpha \alpha$. If $\beta \in \mathrm{OVL}(s,t)$ then let s_β and t_β be defined by $s = s_\beta \beta$ and $t = \beta t_\beta$.*

By $\mathrm{OVL}(t,s) \cap \mathrm{OVL}(s,t) = \emptyset$, there can be no confusion between s_α and s_β or between t_α and t_β.

2.1 Positions

By $[m,n]$ we mean the set of integer numbers between, and including, m and n. We flip the square bracket next to m or n to indicate that m or n, respectively, shall be excluded. Positions in a string w are integer numbers in $[0,|w|]$. We call 0 and $|w|$ the (left and right, respectively) *boundary positions* of w, and the other positions the *inner positions* of w. The inner positions represent the spaces between letters.

Let a (finite or infinite) derivation $\mathcal{D} : w_0 \to w_1 \to \dots$ be presupposed. We denote positions in $\mathcal{D}$ by pairs (i,p) where p is a position in w_i. The position $(i-1,p)$ *corresponds* to the position (i,q), in symbols $(i-1,p) \hookrightarrow_{res} (i,q)$, if there are $x, y \in \Sigma^*$ such that $w_{i-1} = xsy$, $w_i = xty$, and either $0 \leq q = p \leq |x|$ or $|xs| \leq p \leq |xsy|$ and $q = p - |s| + |t|$.

If to a given $(i-1,p)$ a q exists such that $(i-1,p) \hookrightarrow_{res} (i,q)$, then q is unique. If no such q exists, i.e., if $|x| < p < |xs|$, then p is said to be *consumed* at step i. Likewise if to a given (i,q) a p exists such that $(i-1,p) \hookrightarrow_{res} (i,q)$, then p is unique. If no such p exists, i.e., $|x| < q < |xt|$, then q is said to be *introduced* at step i.

The *redex position*, $R(i)$, of the i-th rewrite step in $\mathcal{D}$ is defined by $R(i) = |x|$ if $w_{i-1} = xsy$ and $w_i = xty$ for some $x, y \in \Sigma^*$.

The set of positions consumed in step i is $]R(i), R(i)+|s|[$. The set of positions introduced in step i is $]R(i), R(i)+|t|[$.

The equivalence closure of $\hookrightarrow_{res}$, denoted by $\sim_{res}$, allows us to identify a position in w_i with its corresponding position in w_j. If $(i,p) \sim_{res} (j,q)$ then the position p in w_i and the position q in w_j are called *residuals* (of each other). We will conveniently speak about a position p in string w_i when we mean the residual of p.

Example 1. As a running example we use the system $aabbab \to abbaabba$. Consider the following derivation $\mathcal{D}$:

$$\begin{aligned} w_0 &= aabb\underline{aabbab}bb \to \underline{aabb\overline{ab}}\overline{b * aabba}bb \to \overline{abbaabba}b * \underline{aabbab}b \\ &\to abb\underline{aabbab} * \overline{abbaabba}b \to abb\overline{abbaabba} * abb\underline{aabbab} \\ &\to abbabbaabb\underline{a * abb\overline{ab}}\overline{baabba} \to abbabbaabb\overline{abbaabba}baabba = w_6 \end{aligned}$$

The set of positions consumed in the first step of $\mathcal{D}$ is $[5,9]$. The set of positions introduced in the first step of $\mathcal{D}$ is $[5,11]$. The position marked by $*$ in any word in $\mathcal{D}$ is a residual of the position marked by $*$ in any other word. According to our convention, we may say that the position $*$ introduced in the first step of $\mathcal{D}$ is consumed by the last step of $\mathcal{D}$.

Definition 2 ([8]). *A step i is called* digestible*, in symbols $D(i)$, if all contractum positions in w_i are later consumed. The derivation is called* well-behaved *if no step in it is digestible. The 1STS $s \to t$ is called* well-behaved *if all its derivations are well-behaved.*

Note that according to our definitions, the inner positions of the contractum are exactly the introduced positions.

Example 2. The first step in the derivation $\mathcal{D}$ from Example 1 is digestible.

Theorem 1 ([8]). *It is decidable whether an arbitrary 1STS is well-behaved. Uniform termination is decidable for the class of well-behaved 1STS.*

Definition 3 ([9, Definition 5.2]). *For each $j \geq i$ let* $\mathrm{Rem}(i,j)$ *(for "remainder") denote the set of all residuals in w_j of the set of contractum positions from step i. Step $j \geq i$ is said to* consume from the left *the remainder of step i if* $\mathrm{Rem}(i,j) \neq \emptyset$ *and*

$$\min \mathrm{Rem}(i, j-1) \in]R(j), R(j) + |s|[.$$

Step $j > i$ is said to consume from the right *the remainder of step i if* $\mathrm{Rem}(i,j) \neq \emptyset$ *and*

$$\max \mathrm{Rem}(i, j-1) \in]R(j), R(j) + |s|[.$$

Intuitively, step j consumes from the left (right) the remainder of step i if it consumes the leftmost (rightmost) position, but not every position, from the remainder at step $j-1$.

Example 3. The second step in the derivation $\mathcal{D}$ from Example 1 consumes from the left the remainder of step 1, whereas the third step consumes it from the right.

Definition 4 ([9, Definition 5.5]). *We say that step i is*

- both-sides-digestible*, in symbols $BD(i)$, if $D(i)$ holds and some steps $j > i$ consume from the left the remainder of step i, and some steps $j > i$ consume from the right the remainder of step i;*
- left-digestible*, in symbols $LD(i)$, if $D(i)$ holds and all steps $j > i$ that partially consume the remainder of step i do so from the left (i.e., no steps $j > i$ consume from the right the remainder of step i);*
- right-digestible*, in symbols $RD(i)$, if $D(i)$ holds and all steps $j > i$ that partially consume the remainder of step i do so from the right (i.e., no steps $j > i$ consume from the left the remainder of step i).*

The conditions are mutually exclusive for given i. A derivation is said to satisfy BD, LD, or RD, if some of its steps i satisfy $BD(i)$, $LD(i)$, or $RD(i)$, respectively. A 1STS $s \to t$ satisfies BD, LD, or RD, if some of its derivations satisfy BD, LD, or RD, respectively. We define (both-sides, left, right)-indigestibility for steps, derivations and systems, denoting them by $\neg BD, \neg LD, \neg RD$, by negating the respective conditions. Note that by definition a 1STS is well-behaved if and only if it satisfies $\neg BD \wedge \neg LD \wedge \neg RD$.

Example 4. The condition $BD(1)$ holds for the derivation from Example 1.

Theorem 2 ([6]). *The Conditions LD and RD are decidable for 1STSs.*

Proof. Conditions LD and RD are equivalent to McNaughton's conditions $C2$ and $C3$, respectively [6, Theorem 6.1]. This shows up in cases I and II in his proof. □

If $\neg LD \wedge \neg RD$ holds then BD is equivalent to McNaugton's Condition $C1$. However, Condition BD is not equivalent to $C1$ in the general case.

Example 5. The system from Example 1 satisfies $\neg RD$ and $\neg C1$. However, it satisfies BD as the derivation $\mathcal{D}$ exhibits.

2.2 Chain Graphs

The notion of chain graph gives one the means to reason in detail about the relation between steps in a derivation.

Definition 5. *Let $\mathcal{D} : w_0 \to w_1 \to \dots$. Let $w_i = xsy$ for some i, x, y. The factor s in w_i is called* live *if:*

- *there is a step $j \geq i$ such that $(i, |x|) \sim_{res} (i-1, R(j))$, i.e., at step j the redex $|x|$ from w_i is reduced;*
- *$(i, p) \sim_{res} (j-1, p')$ for all $|x| \leq p \leq |xs|$; i.e., no position of s is consumed until s is rewritten.*

Informally speaking, a live factor is finally reduced and it is not touched before then. Note that the live factor in w_i need not be reduced in the very next step $w_i \to w_{i+1}$. Since the residuals of overlapping redexes overlap, live factors do not overlap.

Definition 6 ([5, Definition 4.25]). *The chain graph of a (finite or infinite) derivation $\mathcal{D} : w_0 \to w_1 \to \dots$ is a directed graph (V, E). The vertices in V are the positions of live factors. The edges in $E = E_0 \cup E_1$ are defined as follows:*

- *if $(i-1, p) \hookrightarrow_{res} (i, q)$ and $(i-1, p), (i, q) \in V$ then $((i-1, p), (i, q)) \in E_0$;*
- *if $(i-1, R(i)), (i, q) \in V$, and some of the positions $(i, q), \dots, (i, q+|s|)$ are introduced by step i, then $((i-1, R(i)), (i, q)) \in E_1$.*

We define selector functions $\mathsf{src}, \mathsf{tgt}, \mathsf{level} : E \to \mathbb{N}$ *for the* source*, the* target*, and the* level *of an edge $k \in E$ by $\mathsf{src}(k) = p$, $\mathsf{tgt}(k) = q$, $\mathsf{level}(k) = i$ if $k = ((i-1, p), (i, q))$.*

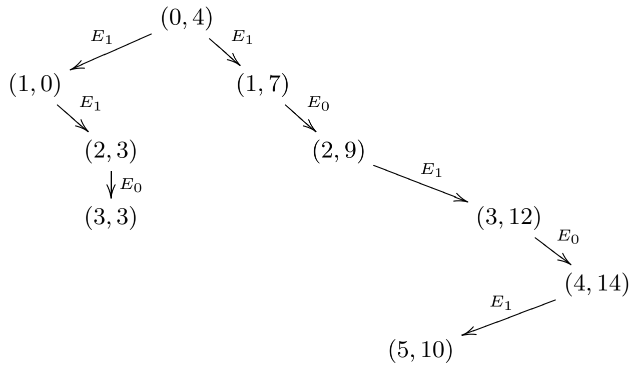

Fig. 1. The chain graph of the derivation from Example 1

The chain graph is a forest of finitely many trees, $T_1, \ldots, T_K$, rooted at the positions $p_1 < \cdots < p_K$ of the live redexes in w_0.

Example 6. Figure 1 shows the chain graph of the system in Example 1. The lowest edge has the source vertex $(4, 14)$, the target vertex $(5, 10)$, it is in E_1, and its level is 5.

Definition 7. *Edges from the set E_1 will also be called* active. *An active edge k is called a* left edge *if* $\mathsf{src}(k) > \mathsf{tgt}(k)$*; and a* right edge *if* $\mathsf{src}(k)+|s| < \mathsf{tgt}(k)+|t|$. *We will call the active edges on the same level* rivals.

By $s \not\sqsubseteq t$ and $|s| < |t|$, every active edge is a left or a right edge.

Lemma 1. *If k is a left edge at level i then $w_{i-1} = zs_\beta sy$ and $w_i = zst_\beta y$ for some $\beta \in \mathrm{OVL}(s, t)$ and $z, y \in \Sigma^*$. Moreover $\mathsf{src}(k) = |zs_\beta|$ and $\mathsf{tgt}(k) = |z|$. If k is a right edge at level i then $w_{i-1} = xss_\alpha v$ and $w_i = xt_\alpha sv$ for some $\alpha \in \mathrm{OVL}(t, s)$ and $x, v \in \Sigma^*$. Moreover $\mathsf{src}(k) = |x|$ and $\mathsf{tgt}(k) = |xt_\alpha|$.*

Proof. Straightforward from the definitions. □

Lemma 2. *If $(i, p) \sim_{res} (j, q)$ and $(i, p') \sim_{res} (j, q')$ and $p < p'$ then $q < q'$.*

Proof. By induction on $|j - i|$, with the inductive step done by case analysis on $R(i) \leq p$, $p < R(i) < p'$, and $p' \leq R(i)$. □

2.3 Family Members

New tools developed in this section will enable us to speak in more detail about infinite derivations.

Definition 8 ([9, Definition 7.1]). *Let k be a right edge at level i. Then s_α and $(i-1, |xs|)$ in Lemma 1 are called the* husband *and its position, respectively. Likewise for a left edge, s_β and $(i-1, |z|)$ are called the* husband *and its position, respectively.*

Intuitively, a husband is a non-empty factor that is supplemented to a live redex by the next rewrite step. The husband positions of k are the residuals of the positions of the live redex created by k.

Example 7. In the chain graph of the derivation from Example 1, the husbands of the edges at level 1 are *aabb* at position $(0, 0)$ and b at position $(0, 10)$.

Definition 9. *[[9, Definition 7.3]] Let p be a position in the husband h of an active edge k. Then we call the vertex $(i-1, R(i))$ the* mother-in-law *of p if p is introduced in step i. A* mother-in-law *of the active edge k is the mother-in-law of one of the positions in the husband of k. The step that rewrites the target redex of k is called the* marriage consumption *step of k.*

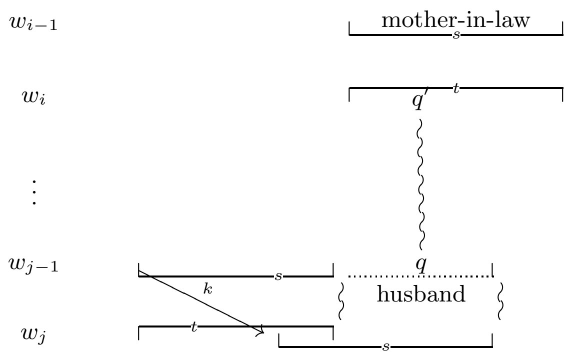

Fig. 2. Husband and mother-in-law

Example 8. The vertex $(0, 4)$ in the chain graph of the derivation $\mathcal{D}$ from Example 1 is the mother-in-law of the position $*$ in w_4. The string *aabb* at position $(4, 10)$ is the husband of the edge going from $(4, 14)$ to $(5, 10)$, and the vertex $(3, 3)$ is its mother-in-law.

Note that a mother-in-law need not be the source vertex of an edge. In other words, the rewrite step $w_{i-1} \to w_i$ need not create a live redex.

3 Uniform Termination of One-rule Single-Threaded Systems

In this section we define single-threaded derivations, show how single-threadedness can be derived, and use single-threadedness for decidability of uniform termination in a special case.

3.1 Single-Threadedness and Independence

Definition 10. *A path in the chain graph of a derivation is called* single-threaded *if every edge on it is active. A derivation is called* single-threaded *if its chain graph is a single-threaded path. A 1STS is called* single-threaded *if it admits an infinite single-threaded derivation.*

Theorem 3 ([7, Theorem 7.4],[9, Theorem 3]). *Every non-terminating, well-behaved 1STS is single-threaded.*

McNaughton's xy-sequence corresponds to a single-threaded path.

Definition 11 ([9, Definitions 8.2 and 8.3]). *An active edge k is called* independent *if all its mothers-in-law are ancestors of k. A mother-in-law that is not an ancestor of k is called* alien to k. *A path is called* independent *if every active edge on it is independent.*

In other words, an edge is dependent iff it has an alien mother-in-law. In contrast, an independent path does not need any other paths to proceed with its reductions.

Example 9. The left path in Figure 1 is independent. However, the right path is not — the mother-in-law $(3,3)$ of the boundary position $(4,10)$ of the husband is a vertex in the left path and is hence alien to the edge $((4,14),(5,10))$.

Lemma 3. *If there is an infinite derivation whose chain graph contains an infinite independent path whose first i edges are active, then there is also an infinite derivation whose chain graph contains an infinite independent path whose first $i+1$ edges are active. Moreover, the two derivations coincide up to, and including, step i.*

Proof. Let k denote the active edge at level i in the independent path S. Let $j > i$ denote the next level at which S has an active edge, k'. By symmetry we may assume that k' is a right edge. Let h be the husband of k', i.e., there are $x, y, g \in \Sigma^*$ such that $w_{j-1} = xshy \to xthy = xgsy = w_j$. Since $k' \in S$ and all edges between k and k' are inactive, the occurrence of s is preserved, i.e., none of its positions is consumed, during the derivation $w_i \to^* w_{j-1}$. Only the parts left or right to it in w_i may be touched during this derivation. All mothers-in-law of h are above level i since they are both redexes and ancestor nodes of k. Hence h is present in w_i and not touched during the derivation $w_i \to^* w_{j-1}$ either. Let $x' \to^* x$ and $y' \to^* y$ render the changes that happened during the derivation $w_i \to^* w_{j-1}$. Then the derivation can be rearranged thus:

$$\begin{array}{ccc} w_i = x'shy' & \xrightarrow{*} & w_{j-1} = xshy \\ \downarrow & & \downarrow \\ w'_{i+1} = x'gsy' & \xrightarrow{*} & w'_j = w_j = xgsy \end{array}$$

We will show that the chain graph of the new derivation $\mathcal{D}' : w_0 \to w_1 \to \cdots \to w_i \to w'_{i+1} \to \cdots \to w'_{j-1} \to w'_j = w_j \to w_{j+1} \to \ldots$ has an infinite independent path with the first $i+1$ edges active. In steps $w'_{i+1} \to \cdots \to w'_{j-1} \to w'_j$ we execute reductions left and right from gs in the same order as they were executed in $\mathcal{D}$.

First note that the inactive edge at level $i+1$ having source $(i, |x'|)$ in the chain graph of $\mathcal{D}$ is replaced by the active edge $((i, |x'|), (i+1, |x'g|))$ in the chain graph of $\mathcal{D}'$. Let us denote this active edge by K.

Let S consist of vertices

$$v_0, \ldots, v_{i-1}, (i, |x'|), \ldots, (j-1, |x|), (j, |xg|), v_{j+1}, \ldots$$

and respective edges between them. The path S' consisting of the vertices:

$$v_0, \ldots, v_{i-1}, (i, |x'|), (i+1, |x'g|), \ldots, (j, |xg|), v_{j+1}, \ldots$$

in the chain graph of $\mathcal{D}'$ has by its construction first $i+1$ edges active. It suffices to show that it is an infinite independent path.

Suppose that there is a dependent edge $l \in S'$. There are 3 possible cases:

- $\mathsf{level}(l) \leq i$. Since $\mathcal{D}'$ up to step i is the same as $\mathcal{D}$ and hence the respective parts of their chain graphs are the same, l is dependent also in the chain graph of $\mathcal{D}$, a contradiction.
- $\mathsf{level}(l) \in]i, j]$. Then $l = K$, since K is the only active edge in those levels. Hence one of the positions in the husband h of K is introduced before level $i+1$ by an alien mother-in-law m. But before level $i+1$ the derivations and their chain graphs are the same, hence m is also an alien mother-in-law of k', a contradiction.
- $\mathsf{level}(l) > j$. Then l was present in the original chain graph as well since all reduction steps later than j are the same. Let $m = (j'-1, R(j'))$ be an alien mother-in-law of l. We have 3 possible cases:
  - $j' > j$. Since the steps after j and hence their chain graphs are the same, m is alien to l in S as well.
  - $j' \in]i, j]$. Obviously, $j' \neq i+1$, because m is alien. Let p be the position in the husband of l introduced by the reduction corresponding to m. Then p has a residual p' in w'_j. We can either have $p' < |x|$ or $p' > |xgs|$, since other positions stay untouched during the derivation $w'_{i+1} \to^* w'_j$. Therefore p' is either an inner position of x or of y. To fix our attention, suppose that it is an inner position of y. By Lemma 2, we have $w'_{j'-1} = x''gsy_1sy_2$ for some $x'', y_1, y_2 \in \Sigma^*$, where $x' \to^* x'' \to^* x$ and $y' \to^* y_1sy_2 \to^* y$. Let $|x''gsy_1| < p'' < |x''gsy_1t|$ be the residual of p introduced in step j'. Consider the corresponding reduction step in $\mathcal{D}$: $w_{j'} = x''shy_1sy_2 \to x''shy_1ty_2 = w_{j'+1}$. The position $p''' = |h| - |g| + p''$ in $w_{j'+1}$ is introduced in this reduction. One shows that (j', p''') is a residual of (j, p') in $\mathcal{D}$. Since $(j', |x''|) \in S$, the mother-in-law $(j', |x''shy_1|)$ is alien. Hence S is not independent, a contradiction.

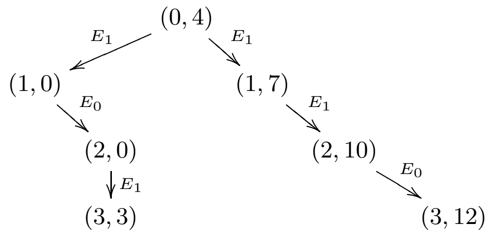

Fig. 3. The chain graph from Example 10

- $j' \leq i$. By definition of mother-in-law, the step $w'_{j'-1} \to w'_{j'}$ in $\mathcal{D}'$ introduces a residual of $(\mathsf{level}(\ell) - 1, p)$. The same step in $\mathcal{D}$ introduces a residual of $(\mathsf{level}(\ell) - 1, p)$ in $\mathcal{D}$, because the derivation $w_i \to^* w_{j-1} \to w_j$ touches exactly the same positions as the derivation $w_i \to w'_{i+1} \to^* w_j$. So m is an alien mother-in-law of ℓ also in the chain graph of $\mathcal{D}$, a contradiction.

So S' contains no dependent edge, which finishes the proof. □

Example 10. Consider the first three steps of $\mathcal{D}$ from Example 1. Let B denote the right branch in its chain graph. The edges on B come from sets E_1, E_0, E_1. Pushing up the second active edge from B, results in the following derivation:

$$aabb\underline{aabbab}bb \to aabb\overline{abb * \underline{aabbab}}b \to$$
$$\to \underline{aabbab}b * \overline{abbaabbab} \to \overline{abbaabbab} * abbaabbab$$

Its chain graph is shown in Figure 3. Note that the right path starts with two active edges.

Lemma 4 ([9, Lemma 15]). *If the chain graph of an infinite derivation contains an infinite independent path then there is a derivation that contains an infinite, single-threaded path starting from level 0.*

Proof. First we drop enough initial steps from the derivation, so that the independent path starts from level 0. Then we construct the i-the step of the target derivation and the i-th level of its chain graph by applying Lemma 3 i times. □

Lemma 5. *A derivation whose chain graph contains an infinite single-threaded path starting from level* 0 *is a single-threaded derivation.*

Proof. Let an infinite derivation be given that contains an infinite, single-threaded path, S. As every edge on the path is active, there cannot be, besides the path, another redex that is rewritten during the derivation. By definition of chain graph there is, therefore, no inactive edge in the chain graph. By the same token, the active edges have no rivals. So there are no edges outside S. □

The concepts and lemmas introduced so far can be used to prove:

Theorem 4. *Every both-sides-indigestible, non-terminating 1STS is single-threaded.*

3.2 Simulation by a Pushdown Automaton

We show in this section that the single-threaded derivations of a right-indigestible 1STS can be rendered by a pushdown automaton, whence the uniform termination problem for single-threaded, right-indigestible 1STSs is decidable.

Lemma 6 ([9, Proposition 34]). *Let $w_0 \to w_1 \to \dots$ be an infinite single-threaded derivation and let $D(i)$ hold for some $i > 0$. If k_i is a left edge then $LD(i)$ holds. If k_i is a right edge then $RD(i)$ holds.*

Proof. To fix our attention, suppose that k_i is a left edge. Hence reduction of the target redex consumes positions introduced in the i-th step from the left. By induction we can show that no step consumes positions from the right. □

During the remainder of this section we assume that the given 1STS $s \to t$ is single-threaded and satisfies $\neg RD$.

Lemma 7 ([9, Lemma 35]). *Let k be a right edge at level i in the chain graph of a single-threaded derivation. Then no position $p \leq \mathsf{src}(k)$ in w_{i-1} is consumed later.*

Proof. By contradiction. Suppose that there is a right edge k at level i and the position $p \leq \mathsf{src}(k)$ is consumed later. Since the derivation is single-threaded, we can show by induction that all positions between $\mathsf{src}(k)$ and $\mathsf{tgt}(k)$ in w_i are also consumed; hence $D(i)$ holds. By Lemma 6, we get $RD(i)$, a contradiction. □

Definition 12 ([9, Definition 10.5]). *To a 1STS $s \to t$, we assign a generalized pushdown automaton [11] $\mathcal{A}$ whose transitions will correspond to rewrite steps in a given derivation. The input alphabet and the stack alphabet are Σ each. The state of the automaton is encoded as the contents of a stack of size strictly bounded by $|t|$. So a configuration is a pair $(x, y) \in \Sigma^{<|t|} \times \Sigma^*$. The automaton has the transition relation $\vdash \subseteq (\Sigma^{<|t|} \times \Sigma^*) \times (\Sigma^{<|t|} \times \Sigma^*)$ defined by:*

$$\begin{cases} (x, y) \vdash (x', t_\beta y) & \text{if } x = x' s_\beta,\ \beta \in \mathrm{OVL}(s, t),\ x \in \Sigma^{<|t|}, x', y \in \Sigma^* \\ (x, y) \vdash (t_\alpha, y') & \text{if } y = s_\alpha y',\ \alpha \in \mathrm{OVL}(t, s),\ x \in \Sigma^{<|t|},\ y', y \in \Sigma^* \end{cases}$$

The transition relation $\vdash$ is well-defined by $|x'| < |x| < |t|$ and $|t_\alpha| < |t|$. A finite or infinite sequence of transitions is called a *computation*.

Lemma 8 ([9, Lemma 37]). *If $\mathcal{A}$ admits an infinite computation then there is an infinite derivation.*

Proof. One shows that for all $x, x' \in \Sigma^{<|t|}$ and $y, y' \in \Sigma^*$, if $(x, y) \vdash (x', y')$ then $xsy \to x'sy'$ or $xsy \to xx'sy'$. □

Definition 13. *We say that $\mathcal{A}$ is* put on *the derivation $w_0 \to w_1 \to \dots$, if its configuration is set to (x, y), and $|x| < |t|$, where x and y are the left and right contexts of the first rewrite step, $w_0 \to w_1$.*

Lemma 9. *The automaton $\mathcal{A}$, put on an infinite, single-threaded derivation, admits an infinite computation.*

Proof. We prove that $\mathcal{A}$ admits one transition and thereafter it is put on an infinite, single-threaded derivation again. By applying this argument i times, we can construct the i-th transition of the automaton for any $i > 0$.

To prove the claim, let an infinite, single-threaded derivation $\mathcal{D} : w_0 \to w_1 \to \dots$ be given, and let $w_0 = xsy$ and $R(1) = |x|$ for some $x, y \in \Sigma^*$. If $R(1) > R(2)$ (k_1 is a left edge), then $x = x's_\beta$ and $w_1 = x'st_\beta y$ for some $x' \in \Sigma^*$ and $\beta \in \mathrm{OVL}(s,t)$. The automaton can make a transition $(x, y) \vdash (x', t_\beta y)$, and is so put on the remaining derivation $w_1 \to w_2 \to \dots$. If $R(1) < R(2)$ (k_1 is a right edge), then $y = s_\alpha y'$ and $w_1 = xt_\alpha sy'$ for some $y' \in \Sigma^*$ and $\alpha \in \mathrm{OVL}(t,s)$. By Lemma 7, the prefix x remains unaffected by the derivation $w_1 \to w_2 \to \dots$. Now for all $i > 0$ let w'_i be defined by $w_i = xw'_i$. Then $w'_1 \to w'_2 \to \dots$ is again an infinite, single-threaded derivation. ☐

Lemma 10. *Let S be a path in the chain graph of an infinite, single-threaded derivation. If S contains infinitely many active edges then it contains infinitely many left edges and infinitely many right edges.*

Proof. Suppose that there are only finitely many left edges on S. Then there is some N such that k_n is a right edge, or an inactive edge, for all $n > N$. Let $a_n = |w_n| - \mathtt{tgt}(k_n)$. Obviously $a_n \geq 0$ for all $n > N$. On the other hand, the subsequence of all a_n, $n > N$ for which k_n is a right edge strictly decreases. This gives a contradiction. ☐

Lemma 11. *If $s \to t$ is a right-indigestible, single-threaded 1STS then $\mathcal{A}$ admits an infinite computation.*

Proof. Let $s \to t$ admit the infinite, single-threaded derivation $w_0 \to w_1 \to \dots$. In order to work with Lemma 9, we need to ensure $|x| < |t|$ for the left context of the first rewrite step. This is not the case for an arbitrary derivation, but a suitable derivation can be derived as follows.

By Lemma 10, the single-threaded path of $w_0 \to w_1 \to \dots$ contains a right edge, at level i say. Then the derivation $w_{i-1} \to w_i \to \dots$ starts with a right edge: we have $w_{i-1} = xss_\alpha y'$ and $w_i = xt_\alpha sy'$ for some $x, y' \in \Sigma^*$ and $\alpha \in \mathrm{OVL}(t,s)$. By Lemma 7 the prefix x remains unaffected by the derivation $w_{i-1} \to w_i \to \dots$. Now for all $j \geq i-1$ let w'_j be defined by $w_j = xw'_j$. Then $\mathcal{D} : w'_1 \to w'_2 \to \dots$ is again an infinite, single-threaded derivation. Moreover $|t_\alpha| < |t|$ holds for the left context t_α of its first rewrite step. By Lemma 9, the automaton $\mathcal{A}$ put on $\mathcal{D}$ admits an infinite computation. ☐

Example 11. Consider the well-behaved system $abcd \to cdcdbabab$ taken from [7], and the infinite derivation:

$$abcdcd \to cdcdbababcd \to cdcdbabcdcdbabab \to cdcdbcdcdbababcdbabab \to \dots$$

The corresponding computation is:

$$(\varepsilon, cd) \vdash (cdcdbab, \varepsilon) \vdash (cdcdb, cdbabab) \vdash (cdcdbab, babab) \vdash \dots$$

Definition 14. *The uniform halting problem of pushdown automata is the following problem: "Given a pushdown automaton* $(\Sigma, Z, Q, \vdash, q_0, z_0)$ *— is there* $(x, y) \in Q \times Z^*$ *that initiates an infinite computation?"*

Theorem 5. *The uniform termination problem is decidable for the class of 1STS* $s \to t$ *that satisfy* $\neg BD \wedge (\neg RD \vee \neg LD)$.

Proof. By symmetry we may assume $\neg RD$. By Lemmas 8 and 11, the uniform termination problem is reduced to the uniform halting problem of pushdown automata which is decidable [12]. □

4 Applications

There is a decidable sufficient criterion for both-sides-indigestibility of 1STSs. First BD can be characterized by the existence of two peculiar single-threaded derivations, then one can develop a simple test for non-existence of such derivations, based on the sets of suffixes of s and t that can be consumed and introduced, respectively. The question whether BD is decidable remains open.

We give several examples of systems to which this criterion and our theorems apply.

Example 12. The 1STS $R = \{caabca \to aabccaabc\}$ is both-sides-indigestible, satisfies $\neg LD$ and RD and is not tame.

Example 13 ([2]). The SOF 1STS $aaabbab \to abbaaabba$ satisfies $\neg BD \wedge \neg LD \wedge RD$ and is tame and terminating.

Example 14. The non-SOF 1STS $R = \{babbabb \to abbabbbba\}$ satisfies $\neg BD \wedge LD \wedge \neg RD$. It is non-tame and non-terminating.

Kobayashi et al. [4, page 603] find no instances for the case $SOF \wedge \neg BD \wedge RD \wedge LD$. Non-SOF systems satisfying $\neg BD \wedge RD \wedge LD$ however do exist:

Example 15. For every $n \geq 3$, the 1STS $R = \{ba(ab)^n \to (ab)^{n+2}a$ is not both-sides-digestible. However R is both left-digestible ($(ab)^{n+1}a$ suffix of $(ba)^{n+1}$) and right-digestible (aba prefix of $(ab)^n$).

5 Conclusion

We have shown that one-rule Semi-Thue Systems (1STSs) that satisfy $\neg BD \wedge (\neg RD \vee \neg LD)$ have a decidable uniform termination problem, for their non-terminating members admit infinite single-threaded derivations, which can be simulated by pushdown automata. The uniform termination problem for 1STSs that satisfy $\neg BD \wedge RD \wedge LD$ is open.

Acknowledgements: We are grateful to Robert McNaughton for his encouraging us to do this investigation and to the anonymous referees for their constructive comments. The first of the authors is grateful to Paweł Urzyczyn for his infinite patience.

References

1. Ronald Book and Friedrich Otto. *String-rewriting systems*. Texts and Monographs in Computer Science. Springer, New York, 1993.
2. Alfons Geser. *Is termination decidable for string rewriting with only one rule?* Habilitation thesis, Wilhelm-Schickard-Institut, Universität Tübingen, Germany, January 2002. 201 pages.
3. Alfons Geser, Dieter Hofbauer, and Johannes Waldmann. Termination proofs for string rewriting systems via inverse match-bounds. *J. Automated Reasoning*, 2005. In print.
4. Yuji Kobayashi, Masashi Katsura, and Kayoko Shikishima-Tsuji. Termination and derivational complexity of confluent one-rule string rewriting systems. *Theoretical Computer Science*, 262(1/2):583–632, 2001.
5. Winfried Kurth. *Termination und Konfluenz von Semi-Thue-Systemen mit nur einer Regel*. Dissertation, Technische Universität Clausthal, Germany, 1990.
6. Robert McNaughton. The uniform halting problem for one-rule Semi-Thue Systems. Technical Report 94-18, Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, August 1994. See also "Correction to 'The Uniform Halting Problem for One-rule Semi-Thue Systems'", unpublished paper, August, 1996.
7. Robert McNaughton. Well-behaved derivations in one-rule Semi-Thue Systems. Technical Report 95-15, Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, November 1995. See also "Correction by the author to 'Well-behaved derivations in one-rule Semi-Thue Systems'", unpublished paper, July, 1996.
8. Robert McNaughton. Semi-Thue Systems with an Inhibitor. *J. Automated Reasoning*, 26:409–431, 1997.
9. Wojciech Moczydłowski. Jednoregułowe systemy przepisywania słów. Masters thesis, Warsaw University, Poland, 2002.
10. Wojciech Moczydłowski and Alfons Geser. Termination of single-threaded one-rule Semi-Thue systems. Technical Report TR 02-08 (273), Warsaw University, December 2002. Available electronically at URL `research.nianet.org/∼geser/papers/single.html`.
11. Arto Salomaa. *Theory of Automata*, volume 100 of *Intl. Series of Monographs in Pure and Applied Mathematics*. Pergamon Press, 1969.
12. J. V. Tucker. Computing in algebraic systems. In F. F. Drake and S. S. Wainer, editors, *Recursion Theory, its Generalisations and Applications*, volume 45 of *London Mathematical Society Lecture Note Series*, pages 215–235. Cambridge University Press, 1980.