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

Wojciech Moczydłowski, Jr* Alfons Geser†

December, 2002

**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. We also show that Condition (1) is decidable, thus complementing McNaughton's decidability proofs of (2) and (3). We introduce the notion of a single-threaded derivation. A derivation is single-threaded if every pair of successive steps has an overlap.

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

*Partly supported by KBN Grant 7 T11C 028 20. Address: Faculty of Mathematics, Computer Science and Mechanics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland. Email: `Wojciech.Moczydlowski@mimuw.edu.pl`

†Partly supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the second author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681-2199, USA. Address: ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681. Email: `geser@icase.edu`

# Contents

# 1 Introduction

The uniform termination problem of one-rule Semi-Thue Systems (1STS) is open for 12 years. A systematic exploration of the problem was started by Kurth [Kur90]. Research since has accomplished partial results [McN94, McN95, MS96, Sén96, McN97, KKST01, STKK97, ZG00, Ges02a, Ges02b, Ges02c].

The most promising approach in our view is the investigation of consumption and introduction patterns in derivations. This investigation, started by Kurth [Kur90], has been fully developed by McNaughton [McN95, McN97]. 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 showed that it is decidable whether a rule is well-behaved, i.e., admits only well-behaved derivations. Moreover he showed that uniform termination is decidable for well-behaved rules.

For the self-overlap free (SOF) case, Kobayashi et al. [KKST01] have introduced derivation patterns that are less restrictive ("simple", "gentle", and "tame" derivations). Kobayashi et al. have shown that a gentle 1STS can be transformed to another Semi-Thue System with more rules and an explicit structure. The two systems have equivalent uniform termination problems. As a special case, 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. However Kobayashi et al. have no decidability result to offer beyond the simple systems.

The work we report here is a different extension of McNaughton's well-behaved derivations. We operate with three propositions about derivations in a 1STS:

- Condition **2A** means that all positions introduced at some step are consumed later, some of them from the left and some of them from the right;
- Condition **2B** means that all positions introduced at some step are consumed later, but none of them from the right;
- Condition **2C** means that all positions introduced at some step are consumed later, but none of them from the left.

A derivation is well-behaved if and only if it satisfies $\neg 2A \wedge \neg 2B \wedge \neg 2C$. A derivation is simple if and only if it is tame and it satisfies $\neg 2A$. Our approach aims at adding further combinations of truth values for the three conditions.

While McNaughton uses a finite automaton to decide termination and uniform termination, we want to describe the behaviour of a derivation by a pushdown automaton. Uniform termination of such 1STSs is so reducible to the uniform halting problem of pushdown automata which is decidable [Tuc80]. It turns out that a SOF 1STS has this behaviour if it satisfies $\neg 2A \wedge \neg 2C$, or symmetrically, $\neg 2A \wedge \neg 2B$. McNaughton [McN95] showed that $2B$ and $2C$ are decidable. We show that $2A$ is decidable for SOF 1STS; thus their membership in the class $\neg 2A \wedge \neg 2C$ is decidable.

The proof that the derivation behaviour can be simulated by a pushdown automaton uses Kurth's chain graph concept. The chain graph of a derivation shows the relation between the redex positions. Whereas every infinite derivation has a path in the chain graph that contains infinitely many active edges, our 1STSs admit infinite paths that have active edges *exclusively*. We call such a derivation *single-threaded*. Intuitively, in a single-threaded derivation every step overlaps with the previous one.

This work is a thoroughly revised and extended version of the first author's master's thesis [Moc02].

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. Decidability of $2A$ is proven in Section 4.

# 2 Preliminaries

A string $u$ is called a *factor* of $v$, in symbols $s \sqsubseteq t$, 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 all proper prefixes of the word $u$ is denoted by $\mathrm{Pref}(u)$. 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 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 $D : w_0 \to w_1 \to \ldots$ to denote a derivation named $D$ 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 $s \to t$. 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 [Kur90, Criterion D]. We henceforth assume that both sets are non-empty. 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 $|s| < |t|$.

**Definition 2.1 ([KKST01]).** If $\alpha \in \mathrm{OVL}(t, s)$ then let $s_\alpha$ and $t_\alpha$ be defined by $s = \alpha s_\alpha$ and $t = t_\alpha \alpha$. If $\beta \in \mathrm{OVL}(s, t)$ then let $s_\beta$ and $t_\beta$ be defined by $s = s_\beta \beta$ and $t = \beta t_\beta$.

Whenever we use the symbol $s_\alpha$ it will be clear from which set $\alpha$ originates.

## 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 $w$. Let a (finite or infinite) derivation $D : w_0 \to w_1 \to \ldots$ be presupposed.

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

We say that a position $p$ in $w_{i-1}$ corresponds to a position $q$ in $w_i$, in symbols $(p, i-1) \hookrightarrow_{res} (q, i)$, if there are $x, y \in \Sigma^*$ such that $w_{i-1} = xsy$, $w_i = xty$, and either

$$0 \leq p' = p \leq |x|$$

or

$$|xs| \le p \le |xsy| \quad \text{and} \quad p' = p - |s| + |t|.$$

If to a given $(p, i-1)$ a $q$ exists such that $(p, i-1) \hookrightarrow_{res} (q, i)$, 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 $(q, i)$ a $p$ exists such that $(p, i-1) \hookrightarrow_{res} (q, i)$, 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 set of positions consumed in step $i$ is $]R(i), R(i) + |s|[$. The set of positions introduced in step $i$ is $]R(i), R(i) + |t|[$.

*Example* 2.2. As a running example we use the system $aabbab \to abbaabba$. Consider the following derivation $D$:

$$\begin{array}{c}
aabb\underline{aabbab}bb \\
\downarrow \\
\underline{aabb\overline{ab}}\overline{b \ast aabba}bb \\
\downarrow \\
\overline{abbaabba}b \ast \underline{aabbab}b \\
\downarrow \\
abb\underline{aabbab} \ast \overline{abbaabba}b \\
\downarrow \\
abb\overline{abbaabba} \ast abb\underline{aabbab} \\
\downarrow \\
abbabbaabb\underline{a \ast abb\overline{ab}}\overline{baabba} \\
\downarrow \\
abbabbaabb\overline{abbaabba}baabba
\end{array}$$

The set of positions consumed in the first step of $D$ is $[5, 9]$. The set of positions introduced in the first step of $D$ is $[5, 11]$.

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 $(p, i) \sim_{res} (q, j)$ then the position $p$ in $w_i$ and the position $q$ in $w_j$ are called *residuals* (of each other). The *trace* of a position $p$ in $w_i$ is defined as the set of all residuals $q$ at any $w_j$. We will conveniently speak about a position $p$ in string $w_i$ when we mean the residual, or the trace, of $p$.

*Example* 2.3. The position marked by $\ast$ in Example 2.2 in any word in $D$ is a residual of the position marked by $\ast$ in any other word. According to our convention, we may say that the position $\ast$ introduced in the first step of $D$ is consumed by the last step of $D$.

**Definition 2.4 ([McN97]).** A step $i$ is called *ill*, in symbols $EI(i)$, if all positions introduced in $i$ are later consumed. The derivation is called *well-behaved* if no step in it is ill. The 1STS $s \to t$ is called *well-behaved* if all its derivations are well-behaved.

*Example* 2.5. The first step in the derivation $D$ from Example 2.2 is ill.

**Theorem 2.6 ([McN97]).** *It is decidable whether an arbitrary 1STS is well-behaved. Uniform termination is decidable for the class of well-behaved 1STS.*

**Definition 2.7 ([Moc02, Definition 5.2]).** Let $\mathrm{Rem}(i,j)$ (for "remainder") denote the set of all positions in $w_j$ that have been introduced at step $i < j$. Step $j > i$ is said to *consume from the left* the remainder of step $i$ if

$$\min \mathrm{Rem}(i,j) \in [R(j), R(j) + |s|] \quad \text{and} \quad \mathrm{Rem}(i, j+1) \neq \emptyset.$$

Step $j > i$ is said to *consume from the right* the remainder of step $i$ if

$$\max \mathrm{Rem}(i,j) \in [R(j), R(j) + |s|] \quad \text{and} \quad \mathrm{Rem}(i, j+1) \neq \emptyset.$$

*Example* 2.8. The second step in the derivation $D$ from Example 2.2 consumes from the left the remainder of step 1, whereas the third step consumes it from the right.

**Definition 2.9 ([Moc02, Definition 5.5]).** We say that

- Condition $2A(i)$ holds if $EI(i)$ 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 $i$;
- Condition $2B(i)$ holds if $EI(i)$ holds and no steps $j > i$ consume from the right the remainder of step $i$;
- Condition $2C(i)$ holds if $EI(i)$ holds and 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 $2A$, $2B$, or $2C$, if some of its steps $i$ satisfies $2A(i)$, $2B(i)$, or $2C(i)$, respectively. A 1STS $s \to t$ satisfies $2A$, $2B$, or $2C$, if there is a derivation satisfying $2A$, $2B$, or $2C$, respectively. Note that by definition a 1STS is well-behaved if and only if $\neg 2A \wedge \neg 2B \wedge \neg 2C$ holds.

*Example* 2.10. The condition $2A(1)$ holds for the derivation from Example 2.2.

**Theorem 2.11 ([McN94]).** *The Conditions $2B$ and $2C$ are decidable for 1STSs.*

*Proof.* Conditions $2B$ and $2C$ are equivalent to McNaughton's conditions $C2$ and $C3$, respectively [McN94, Theorem 6.1]. This shows up in cases I and II in his proof. □

However, Condition $2A$ is not equivalent to McNaughton's $C1$.

*Example* 2.12. We claim that the system from Example 2.2 satisfies $2A$ and $\neg 2C$ and $\neg C1$. It does not satisfy $C1$ by definition of $C1$. It does not satisfy $2C$ by equivalence of $2C$ and $C3$. The derivation $D$ exhibits $2A$.

## 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 2.13.** Let $D : w_0 \to w_1 \to \ldots$. Let $w_i = xsy$ for some $i, x, y$. The redex $s$ in $w_i$ is called *active* if:

- All of its inner positions are consumed during $D$.

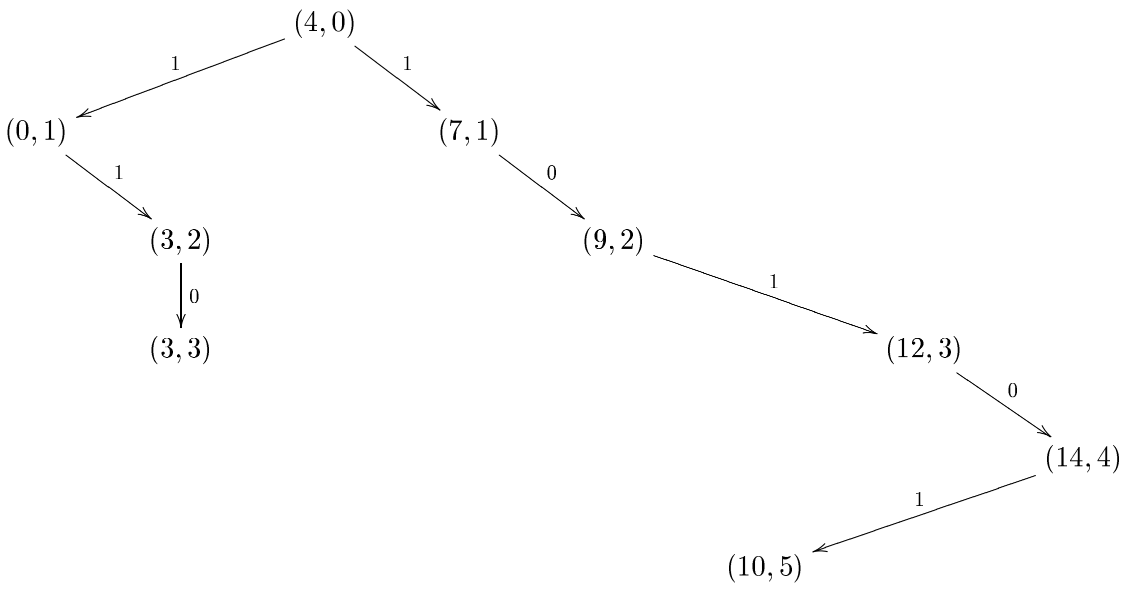


Figure 1: The graph of the derivation from Example 2.2

- The fact that one of its positions has been consumed in the step $w_{j-1} \to w_j$ implies that all of its inner positions have been consumed in the step $w_{j-1} \to w_j$.

Informally speaking, $s$ has to be reduced and it cannot be touched before the reduction.

**Definition 2.14 ([Kur90, Definition 4.25]).** The chain graph of a (finite or infinite) derivation $D : w_0 \to w_1 \to \dots$ is a directed graph $(V, E, \mathrm{lab})$. The vertices in $V$ are pairs $(p, i)$ such that $p$ is the position of an active factor $s$ in $w_i$. The edges in $E$ and their labels $\mathrm{lab} : E \to \{0, 1\}$ are defined as follows:

- if $(p, i-1) \hookrightarrow_{res} (q, i)$ and $(p, i-1), (q, i) \in V$ then $k = ((p, i-1), (q, i)) \in E$, and $\mathrm{lab}(k) = 0$;
- if $p = R(i)$, and the factor $s$ at position $q$ in $w_i$ is introduced[1] by step $i$ then $k = ((p, i-1), (q, i)) \in E$, and $\mathrm{lab}(k) = 1$.

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

*Example* 2.15. Figure 1 shows the chain graph of the system in Example 2.2. The lowest edge has the source vertex $(14, 4)$, the target vertex $(10, 5)$, is labelled by 1, and its level is 5.

**Lemma 2.16.** *The chain graph is a forest of finitely many trees, $T_1, \dots, T_K$, rooted at the positions $p_1 < \dots < p_K$ of the active redexes in $w_0$.*

*Proof.* There are finitely many trees because the word $w_0$ is finite. □

**Definition 2.17.** Edges labelled by 1 will also be called *active*. An active edge $k$ is called a *left edge* if $\alpha(k) > \omega(k)$; and a *right edge* if $\alpha(k) < \omega(k)$. We will call the active edges on the same level *rivals*.

[1]A factor is called introduced if some of its positions is introduced.

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

**Lemma 2.18.** *Rivals $k, k'$ have the same source vertices:* $\text{level}(k) = \text{level}(k') \Rightarrow \alpha(k) = \alpha(k')$. *If SOF holds then at every level there is at most one left edge and at most one right edge.*

*Proof.* The first claim follows immediately from the definition of the chain graph. For the second claim suppose that there is more than one left edge. We then have: $xsy \to xty = x_1 st_\alpha y = x_2 st_{\alpha'} y$ for some $\alpha, \alpha' \in \text{OVL}(s,t)$, $x_1, x_2, y, x \in \Sigma^*$. Since one of $t_\alpha, t'_\alpha$ is a suffix of the other, the two redexes overlap. So only one of them can be active, a contradiction. □

*Example* 2.19. The two edges on the first level in Figure 1 are rivals.

**Lemma 2.20 ([Kur90, Lemma 4.26]).** *The chain graph of an infinite derivation contains a path with infinitely many active edges.*

**Lemma 2.21.** *Let $S$ be a path in the chain graph of an infinite derivation, which contains exclusively active edges. 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 for all $n > N$. Let $a_n = |w_n| - R(n)$. Obviously $a_n \geq 0$ for all $n > N$. On the other hand, the sequence of all $a_n$, $n > N$ strictly decreases. This gives a contradiction. □

**Lemma 2.22.** *If $(p, i-1) {\sim_{res}} (q, i)$ and $i = \text{level}(k)$ then*

1. *$\alpha(k) \leq p$ implies $\omega(k) \leq q$, and*
2. *$\alpha(k) \geq p$ implies $\omega(k) \geq q$.*

*Proof.* Straightforward from the definitions. □

$$
\begin{tikzcd}
i-1 & p \arrow[d, squiggly] & \alpha(k) \arrow[d] & p' \arrow[d, squiggly] \\
i & q & \omega(k) & q'
\end{tikzcd}
$$

Figure 2: Illustration to Lemma 2.22

By induction on $\text{level}(k) - \text{level}(k')$ we immediately get

**Corollary 2.23 ([Moc02, Proposition 20]).** *If $k'$ is a descendant of the edge $k$ and $(p, \text{level}(k) - 1) {\sim_{res}} (q, \text{level}(k') - 1)$ then*

1. *$\alpha(k) \leq p$ implies $\alpha(k') \leq q$, and*
2. *$\alpha(k) \geq p$ implies $\alpha(k') \geq q$.*

By case analysis whether $k$ and $\ell$ are active/inactive, one concludes from Lemma 2.22:

**Corollary 2.24 ([Moc02, Proposition 5]).** *If* $\text{level}(k) = \text{level}(\ell)$ *and* $\alpha(k) < \alpha(\ell)$ *then* $\omega(k) < \omega(\ell)$.

The paths in the chain graph hence do not cross.

**Corollary 2.25.** *If $n < n'$ then $p < q$ holds for all vertices $(p, i) \in T_n$ and $(q, i) \in T_{n'}$ at the same level $i$, where $T_n, T_{n'}$ come from Lemma 2.16.*

*Proof.* By induction on $i$. □

### 2.3 Family Members

In this section we will introduce more notions to speak about derivations.

**Lemma 2.26.** *If $k$ is a left edge at level $i$ then $w_{i-1} = x's_\beta sy$ and $w_i = x'st_\beta y$ for some $\beta \in \mathrm{OVL}(s,t)$ and $x', y \in \Sigma^*$. Moreover $\alpha(k) = |x's_\beta|$ and $\omega(k) = |x'|$. If $k$ is a right edge at level $i$ then $w_{i-1} = xss_\alpha y'$ and $w_i = xt_\alpha sy'$ for some $\alpha \in \mathrm{OVL}(t,s)$ and $x, y' \in \Sigma^*$. Moreover $\alpha(k) = |s_\alpha y'|$ and $\omega(k) = |y'|$.*

*Proof.* Straightforward from the definitions. □

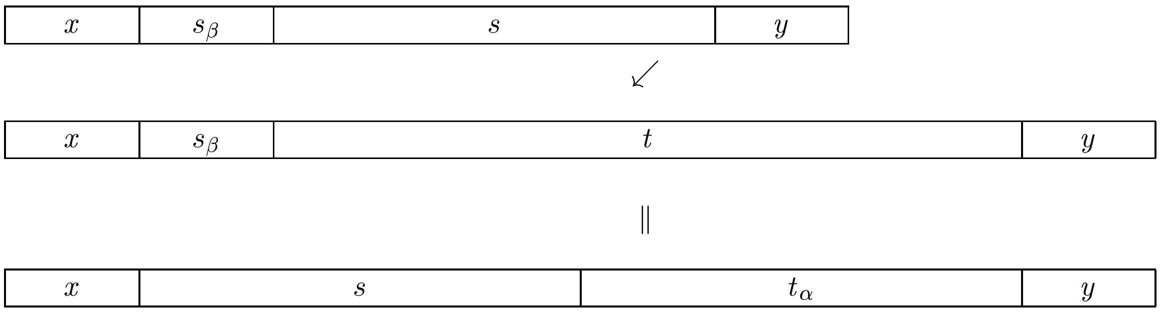


Figure 3: Illustration to Lemma 2.26

**Definition 2.27 ([Moc02, Definition 7.1]).** Let $k$ be a left edge at level $i$. Then $s_\beta$ and $|x|$ in Lemma 2.26 are called the *husband* and its position, respectively. Likewise for a right edge, $s_\alpha$ and $|xs|$ are called the *husband* and its position, respectively.

*Example* 2.28. The husbands of the edges at level 1 in the chain graph of the derivation from Example 2.2 are $aabb$ at position 0 and $b$ at position 12.

**Definition 2.29 ([Moc02, Definition 7.3]).** Let $p$ be a position in the husband of an active edge $k$. Then we call the vertex $(R(i), i-1)$ the *mother-in-law* of $p$ if $p$ is introduced in step $i$. The step that rewrites the redex position $\omega(k)$ is called the *marriage consumption* step of $k$.

*Example* 2.30. The vertex $(4,0)$ in the chain graph of the derivation $D$ from Example 2.2 is the mother-in-law of the position $*$ in the last-but-one word in the derivation. The vertex $(3,3)$ is the mother-in-law of the redex position $(10)$ of the last-but-one word of $D$, and the word $aabb$ is the husband of the edge going from $(14,4)$ to $(10,5)$.

Note that a mother-in-law doesn't have to be the source vertex of an edge.

**Lemma 2.31.** *If $m = (R(i), i-1)$ is some mother-in-law of the active edge $k$ then $i < \mathrm{level}(k)$.*

*Proof.* Let $k$ be an active edge at level $j$, and let $m$ be the mother-in-law of the position $p$ in the husband of $k$. We have $i \leq j$ because the residual of $p$ is introduced at level $i$. As $p$ is not introduced at level $j$, we have $i \neq j$. So $i < j$. □

**Lemma 2.32.** *If a mother-in-law of a right edge $k$ has an outgoing left edge $\ell$ and there is an infinite tree starting from $\ell$, then $k$ is a descendant of $\ell$.*

*Proof.* Let $k$ not be a descendant of $\ell$. Then $k$ is a descendant of an edge at the left of $\ell$. Let $q$ be the ancestor of $k$ at the level $i$, where $m = (R(i), i-1)$ is the mother-in-law of $k$. Let $T$ be the infinite tree starting from $\ell$. The path from $q$ to $k$ would have to cross $T$ in order to consume the marriage, which contradicts Corollary 2.24.

□

# 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 3.1.** 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 graph is a single-threaded path. A 1STS is called *single-threaded* if it admits an infinite single-threaded derivation.

**Theorem 3.2 ([McN95, Theorem 7.4],[Moc02, Theorem 3]).** *Every non-terminating, well-behaved 1STS is single-threaded.*

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

**Definition 3.3.** A set $P$ of positions is called *connected*, if $[p, q] \subseteq P$ for all $p, q \in P$.

Intuitively, a set of positions is connected, if there are no gaps in it.

**Definition 3.4.** A position is consumed by a single-threaded path $K = (k_1, k_2, \dots)$ if it is consumed by any step from $K$.

Likewise one can define the set of positions consumed by a tree, when no other tree exists.

**Lemma 3.5.** *The set of positions consumed by any single-threaded path is connected.*

*Proof.* From the fact that the introduced $t$ and the $s$ consumed in the next step overlap. □

*Example* 3.6. In the derivation from the example 2.2, the right path consumes positions $[10, 11]$ from the first word of $D$.

**Definition 3.7 ([Moc02, Definitions 8.2 and 8.3]).** Let $S$ be a path in the chain graph of an infinite derivation, and let $k \in S$ be an active edge. Then $k$ is called *independent* if all mothers-in-law of $k$ lie on $S$. The path $S$ is called *independent* if every active edge $k \in S$ is independent.

*Example* 3.8. The left path in the chain graph of $D$ from the example 2.2 is independent. However, the right path is not - the mother-in-law of the boundary position of the redex in the last-but-one word is a vertex in the left path.

**Lemma 3.9 ([Moc02, Proposition 12]).** *If a path is independent then for all active edges $k$ on it, all mothers-in-law of the next active edge further down find themselves on* $\text{level}(k)$ *or less.*

We can accordingly "push up" every active edge on an independent path.

**Lemma 3.10.** *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.*

*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 left edge. Let $h$ be the husband of $k'$, i.e., there are $x, y \in \Sigma^*$ such that $w_{j-1} = xhsy$. Since $k' \in S$ and all edges between $k$ and $k'$ are inactive, the occurrence of $s$ is preserved during the derivation $w_i \to^* w_{j-1}$. Only the parts $xh$ and $y$ may be touched during this derivation. By Lemma 3.9, all mothers-in-law of $h$ are above level $i$, so $h$ is not touched either. Suppose that $x' \to^* x$ 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:

$$\begin{array}{ccc} w_i = x'hsy' & \xrightarrow{*} & w_{j-1} = xhsy \\ \downarrow & & \downarrow \\ w'_{i+1} = x'hty' & \xrightarrow{*} & w_j = xhty \end{array}$$

The chain graph of the new derivation $w_i \to w'_{i+1} \to \cdots \to w'_{j-1} \to w'_j = w_j$ is obtained from the given chain graph as follows. The inactive edge at level $i+1$ having source $|x'h|$ is replaced by the active edge $((|x'h|, i), (|x'|, i+1))$. This edge may have a rival. All other inactive edges are preserved, mutatis mutandis. At levels from $i+2$ to $j$, the old edges are preserved, up to the inactive edges at the indicated position of $s$, which move by $|h|$ to the left. Note that the new chain graph has indeed an independent path again. $\square$

**Lemma 3.11.** *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.*

*Proof.* We construct the $i$-the step of the target derivation and the $i$-th level of its chain graph by applying Lemma 3.10 $i$ times. $\square$

**Lemma 3.12.** *A derivation whose chain graph* contains *an infinite single-threaded path 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 another redex besides the path 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 $S$. $\square$

## 3.2 A Criterion for Single-Threaded Derivations

In this section we prove that $SOF \wedge \neg 2A$ implies single-threadedness.

**Lemma 3.13.** *Let $j$ be the highest level that contains a dependent edge, the right edge $k$ say. Let $m = (R(i), i-1)$ be the highest mother-in-law of $k$. If the chain graph of the derivation consists of only infinite non-branching paths, then the position $R(i)$ in $w_{i-1}$ has a residual in $w_j$.*

*Proof.* According to Lemma 2.16, the chain graph is a forest $(T_1, \ldots, T_K)$ for some $K > 0$. By our assumption, the $T_i$ are paths. Let $k$ be in $T_n$ for some $1 \le n \le K$. Then $n < K$, and by Corollary 2.25, $m$ is in $T_{n+1}$. By definition of mother-in-law, there is a position $p$ in the husband of

$k$ which is introduced in step $i$. Because $p$ is not consumed by steps in $]i, j]$, the position $R(i)$ in $w_i$, which is at the left of the residual of $p$, can only be consumed by steps at redex positions at the left of $p$. Their active edges are in $T_n$ by Corollary 2.25. If $R(i)$ is consumed in step $j'$ where $i < j' \leq j$, then the lowest active ancestor edge of the vertex $(R(j'), j'-1)$ has mother-in-law $m$, a contradiction to $j$ being highest. So $R(i)$ is not consumed by steps in $]i, j]$; so $R(i)$ has a residual in $w_j$. □

**Definition 3.14.** The condition $B(i)$ holds if there is both a left and a right edge at level $i$.

**Lemma 3.15 ([Moc02, Lemma 17]).** *$EI(i) \wedge B(i) \Rightarrow 2A(i)$.*

*Proof.* By $B(i)$ we have at level $i$ both a left edge and a right edge. By Lemma 2.26 we get $w_{i-1} = x's_\beta ss_\alpha y'$ and $w_i = x'st'sy'$ such that $t = \beta t'\alpha$ for some $x', y', t' \in \Sigma^*$. By definition of chain graph, both exhibited $s$ factors are rewritten later. The reduction step of the left $s$ consumes the remainder of $t$ from the left; the reduction step of the right $s$ consumes the remainder of $t$ from the right. So we have $2A(i)$. □

**Definition 3.16.** Let the *width* of a derivation be the number of trees in its chain graph.

**Lemma 3.17.** *If $\neg 2A$ holds and there is an infinite derivation of width $K$ starting from $w_0$ then there is also an infinite derivation of width $K$ or less starting from $w_0$ that does not branch in its first step.*

*Proof.* Suppose that the infinite derivation $w_0 \to w_1 \to \ldots$ satisfies $B(1)$. By Lemma 3.15 we get that a position in $w_0$ is not consumed later: $\neg EI(1)$. Then the given derivation can be split into two derivations $u_0 \to u_1 \to \ldots$ and $v_0 \to v_1 \to \ldots$, one of which is infinite by the pigeon-hole principle, such that $w_0 = u_0v_0$. If $u_0 \to u_1 \to \ldots$ is infinite then the infinite derivation

$$w_0 = u_0v_0 \to u_1v_0 \to u_2v_0 \to \ldots$$

has width $\leq K$, and $\neg B(1)$ holds. If $u_0 \to u_1 \to \ldots$ is finite then the infinite derivation

$$w_0 = u_0v_0 \to u_0v_1 \to u_0v_2 \to \ldots$$

has width $\leq K$, and $\neg B(1)$ holds. □

**Lemma 3.18.** *If $\neg 2A$ holds and there is an infinite derivation of width $K$ then there is also an infinite derivation of width $K$ or less that does not branch.*

*Proof.* Using Lemma 3.17 $i$ times, we show that all branches can be eliminated from the first $i$ levels. Thus we approximate an infinite derivation whose chain graph does not branch. □

**Lemma 3.19.** *If $\neg 2A$ holds and there is an infinite derivation of width $K$ then there is also an infinite derivation that contains $K$ or less infinite non-branching paths.*

*Proof.* By Lemma 3.18, there is a derivation $D : d_0 \to d_1 \to \ldots$ containing $K$ non-branching paths. If all of them are infinite, then we are done. Otherwise let $n$ be the length of the longest finite path in $D$. The derivation $D' : d_{n+1} \to d_{n+2} \to \ldots$ contains only infinite paths. □

**Lemma 3.20.** *If $\neg 2A$ holds and there is an infinite derivation then there is a derivation that contains an infinite independent path.*

*Proof.* Let $\neg 2A$ hold. We prove the claim by induction on the width, $K$, of the infinite derivation $D$. For $K = 1$ the claim follows by Lemma 3.18. Let $K > 1$. By Lemma 3.18 we may assume that the chain graph of $D$ does not branch and that it consists of $K$ infinite paths, $S_1, S_2, \ldots, S_K$. If there are no dependent edges then we are done.

Otherwise let $k$ be the topmost dependent edge. By symmetry let us assume that $k$ is a right edge. Let $1 \leq n \leq K$ be defined by $k \in S_n$ and let $j = \text{level}(k)$. Let $m = (R(i), i-1)$ denote the highest mother-in-law of $k$. The vertex $m$ is in $S_{n+1}$ by Corollary 2.25 and by the choice of $D$. Moreover, $n < K$. By Lemma 3.13, $R(i)$ has a residual in $w_j$. Let $u, v \in \Sigma^*$ be defined by $w_{i-1} = usv$ and $w_i = utv$. We have $R(i) = |u|$. Next let $x, y, z \in \Sigma^*$ be given by $w_{j-1} = xsyz$ and $w_j = xtyz$ such that $|xty|$ is the residual of $R(i)$ in $w_j$. Because $k$ is a right edge, we moreover have $z = pq$ and $s = \alpha yp$ for some $p, q \in \Sigma^*$ and $\alpha \in \text{OVL}(t, s)$. Let us focus on the derivation $w_i = utv \to^* xsypq = w_{j-1}$. Its steps left of $R(i)$ and its steps right of $R(i)$ yield derivations $u \to^* xsy$ and $tv \to^* pq$, respectively. By dropping the steps in $u \to^* xsy$ from $D$ we get the infinite derivation

$$D' : xsysv \to xsytv \to^* xsypq \to xtypq = xt_\alpha sq = w_j \to w_{j+1} \to \ldots.$$

We claim that $D'$ has width less than $K$. To this end let $i'$ denote the lowest level of a mother-in-law of $k$ in $D$. We have $i \leq i' \leq j$. The step at level $i'$ completes the factor $p$ needed for the $s$ in $w_j$. There is no left edge at level $i'$ by Lemma 2.32. In $D'$ there is a corresponding step; at its level there is a left edge to the $s$ in $w_j$ and, if there is a right edge at level $i'$ in $D$, also a right edge. The remainder of the paths $S_n$ and $S_{n+1}$ from $D$ now become the left and right branches of a tree in $D'$. The other paths are either preserved or deleted. Hence $D'$ has width less than $K$. □

**Theorem 3.21.** *If a 1STS satisfies $SOF \wedge \neg 2A$ and admits an infinite derivation then it is single-threaded.*

*Proof.* The claim follows from Lemmas 3.20 and 3.12. □

### 3.3 An Application of Single-Threadedness

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

**Lemma 3.22 ([Moc02, Proposition 34]).** *Let $w_0 \to w_1 \to \ldots$ be an infinite single-threaded derivation and let $EI(i)$ hold for some $i > 0$. If $k_i$ is a left edge then $2B(i)$ holds. If $k_i$ is a right edge then $2C(i)$ holds.*

*Proof.* To fix our attention, suppose that $k_1$ is a left edge. Hence positions introduced in $i$-th step are consumed from the left. By induction we can show that any step consuming these position from the right would have to consume rest of them from the left before. Such step hence must not exist, due to the definition of consumption from the right. □

During the remainder of this section we assume that the given 1STS $s \to t$ satisfies $SOF \wedge \neg 2A \wedge \neg 2C$.

**Lemma 3.23 ([Moc02, Lemma 35]).** *Let $k$ be a right edge at level $i$ in the chain graph of a derivation. Then the position $\alpha(k)$ in $V_{i-1}$ is not consumed later.*

*Proof.* By contradiction. Suppose that there is a right edge $k$ at level $i$ and $\alpha(k)$ is consumed later. By Lemma 3.5 all positions between $\alpha(k)$ and $R(i+1)$ are also consumed; hence $EI(i)$ holds. By Lemma 3.22, we get $2C(i)$, a contradiction. □

**Definition 3.24 ([Moc02, Definition 10.5]).** To a 1STS $s \to t$, we assign a generalized pushdown automaton [Sal69] $\mathcal{A}(s,t)$ whose transitions will correspond to rewrite steps in a given derivation. The input alphabet and the stack alphabet are $\Sigma$ 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

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

If $SOF$ holds, there is at most one transition of each sort from $(x,y)$. In other words, $x'$, $\beta$ are unique, and $y'$, $\alpha$ are unique. 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*.

**Theorem 3.25 ([Moc02, Lemma 37]).** *If $\mathcal{A}(s,t)$ admits an infinite computation then there is an infinite derivation.*

*Proof.* One shows that for all $x, x' \in \Sigma^{<|t|}$ and $y, y' \in \Sigma^*$,

$$(x,y) \vdash (x',y') \quad \text{implies} \quad xsy \to x'sy' \quad \text{or} \quad xsy \to xx'sy'.$$

□

**Definition 3.26.** We say that $\mathcal{A}(s,t)$ 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 3.27.** *The automaton $\mathcal{A}(s,t)$, put on an infinite, single-threaded derivation, admits an infinite computation.*

*Proof.* We prove that $\mathcal{A}(s,t)$ 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 $D : w_0 \to w_1 \to \dots$ 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 (unique) $x' \in \Sigma^*$ 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 (unique) $y' \in \Sigma^*$ and $\alpha \in \mathrm{OVL}(t,s)$. By Lemma 3.23 we know that 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. □

**Theorem 3.28.** *Let $s \to t$ be an 1STS that satisfies $SOF \wedge \neg 2A \wedge \neg 2C$, and let $s \to t$ admit an infinite derivation. Then $\mathcal{A}(s,t)$ admits an infinite computation.*

*Proof.* Let there be an infinite derivation. By Theorem 3.21, there is an infinite, single-threaded derivation, $w_0 \to w_1 \to \dots$. In order to work with Lemma 3.27, 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 2.21, 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 3.23 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 $D : w'_1 \to w'_2 \to \dots$ is again an infinite, single-threaded derivation. Moreover $|t_\alpha| < |t|$ holds for the left context $t_\alpha$ of its first rewrite step. By Lemma 3.27, the automaton $\mathcal{A}(s,t)$ put on $D$ admits an infinite computation. □

*Example* 3.29. Consider the well-behaved system (taken from [McN95]) $abcd \to cdcdbabab$ 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 3.30.** The uniform halting problem of pushdown automata is the following problem:

> **Given:** A pushdown automaton $(\Sigma, Z, Q, \vdash, q_0, z_0)$
> **Question:** Are there $(x,y) \in Q \times Z^*$ such that $(x,y)$ initiates an infinite computation?

**Theorem 3.31 ([Tuc80]).** *The uniform halting problem of pushdown automata is decidable.*

**Theorem 3.32.** *The uniform termination problem is decidable for the class of 1STS $s \to t$ that satisfy $SOF \wedge \neg 2A \wedge (\neg 2B \vee \neg 2C)$.*

*Proof.* By symmetry we may assume $\neg 2C$. By Theorem 3.25 and 3.28, the uniform termination problem is reduced to the uniform halting problem of pushdown automata which is decidable by Theorem 3.31. □

# 4 Condition $2A$ is Decidable

In order to prove that Condition $2A$ is decidable for SOF 1SRSs, we first characterize it by the existence of two peculiar single-threaded derivations. Then we will show that the existence of such derivations is decidable and so that $2A$ is decidable.

## 4.1 A Characterization of $2A$

Suppose that the system $s \to t$ satisfies $2A$. We consider a derivation $W : w_0 \to w_1 \to \cdots \to w_n$ that has the following properties:

1. The derivation $W$ exhibits $2A$;

2. The number $n$ is the least among all derivations that satisfy Item 1;

3. The length of $w_0$ ($|w_0|$) is the least among all derivations that satisfy Items 1 and 2.

Next we extract the existence of two certain single-threaded derivations from $W$.

By the definition of $2A$, there is some $1 \le i \le n-1$ such that $2A(i)$ holds. We have $2A(1)$, as otherwise $w_i \to w_{i+1} \to \dots w_n$ is a shorter derivation that exhibits $2A$. Suppose that $u, v \in \Sigma^*$ are such that $w_0 = usv \to utv = w_1$. Then $2A(1)$ says that all positions in $]|u|, |ut|[$ are consumed during $w_1 \to w_2 \to \cdots \to w_n$, some from the left and some from the right. Because $W$ is shortest, some of these positions are consumed in the last step.

Now let $d, e, f \in \Sigma^*$ be such that $t = def$, and $|ud|$ and $|ude|$ are the leftmost and rightmost positions, respectively, of $w_1$ in $]|u|, |ut|[$ that have residuals in $w_{n-1}$. Suppose that $y, z \in \Sigma^*$ are such that $w_{n-1} = ysz \to ytz = w_n$. Since the last step consumes all remaining positions, we get $g, h \in \Sigma^*$ such that $s = geh$ and $|yg|$ and $|yge|$ are the residuals of $|ud|$ and $|ude|$, respectively. We can now summarize what we know about the form of $W$ by

$$usv \to utv = udefv \to^{n-2} ygehz = ysz \to ytz$$

where during the $n-2$ steps in the middle, the factor $e$ is not touched.

As the derivation $udefv \to \cdots \to ygehz$ consumes no positions between $|ud|$ and $|ude|$, every step therein takes place either left of the residual of $|ud|$ or right of the residual of $|ude|$. The steps may be collected in two derivations $L : ud \to^i yg$ and $R : fv \to^j hz$ such that $i + j = n - 2$. Because positions both from the left and from the right are consumed, $d$ and $f$ are non-empty, and $i$ and $j$ are non-zero. By definition of consumption of $|yg|$ and $|yge|$, the strings $g$ and $h$ are non-empty.

We continue by deriving further properties about $L$ and $R$. We will do it only for $L$; for $R$ there is a symmetric proof.

The redex of the first step in $L$ overlaps $d$; otherwise this step can be skipped from $W$ whence $W$ is not minimal. Because $d$ is a prefix of $t$, we get $u', d' \in \Sigma^*$ such that $u = u's_\alpha$ and $d = \alpha d'$ for some $\alpha \in \mathrm{OVL}(t, s)$. The contractum of the last step in $L$ overlaps $g$; otherwise this step can be skipped from $W$. Because $g$ is a prefix of $s$, we get $y', g' \in \Sigma^*$ such that $y = y't_{\beta'}$ and $g = \beta' g'$ for some $\beta' \in \mathrm{OVL}(s, t)$. If $i = 1$ then $u' = y'$ and $d' = g'$.

**Lemma 4.1.** *The chain graph of $L$ is a tree.*

*Proof.* Suppose that the width of $L$ is greater than 1. Then $u'sd'$ has another active factor $s$, the root of another tree. Since $d'$ is a factor of $t$, the other active factor is in $u'$. By definition this $s$ is reduced during $L$; we can drop this reduction step from $W$ and still have $2A(1)$. Hence $W$ is not minimal. □

**Lemma 4.2.** *The chain graph of $L$ is single-threaded.*

*Proof.* For a proof by contradiction let $L$ contain a step

$$ps_\beta ss_\alpha q \to ps_\beta ts_\alpha q = pst'sq$$

for some $\alpha \in \mathrm{OVL}(t, s)$, $\beta \in \mathrm{OVL}(s, t)$, and $p, q, t' \in \Sigma^*$ such that $t = \beta t' \alpha$ and both $s$ are reduced later. We consider three cases:

- If some position between $|ps|$ and $|pst'|$ is never consumed, then we can shorten $W$ by not reducing the left factor $s$.

- All positions between $|ps|$ and $|pst'|$ are consumed before the final step of $W$. In this case $2A$ is exhibited by a derivation shorter than $W$.

- Some position between $|ps|$ and $|pst'|$ is consumed by the last step of $W$. Let one such position be $|psx|$ where $x, x' \in \Sigma^*$ are such that $t' = xx'$. The steps right of $|psx|$ can be collected in a derivation $x'sq \to^* w$ for some $w \in \Sigma^*$. By $|s| < |t|$ we get $|w| \geq |s|$; hence positions left and right of $w$ cannot be consumed in the same step. This contradicts the assumption that both $|psx|$ and $|psxw|$ are consumed in the last step.

□

We call a derivation $w_1 \to^+ w_n$ *full* if all inner positions of $w_1$ are consumed during it. A full derivation is also called an *overlap closure* [GKM83, GZ99] or a *fully seamed* derivation [KKST01].

**Lemma 4.3.** *The derivation $L$ is full.*

*Proof.* If $L$ is not full, there is an inner position in $u'$ that is not consumed during $L$. Let $u' = pq$ for some $p, q \in \Sigma^+$ such that $|p|$ is not consumed. Then no reduction step takes place inside $p$; hence the prefix $p$ can be dropped from the starting string of $W$. □

In a similar fashion one gets $v', f', z', h' \in \Sigma^*$ such that $v = s_\beta v'$ and $f = f'\beta$ for some $\beta \in \mathrm{OVL}(s,t)$, and $z = t_{\alpha'} z'$ and $h = h'\alpha'$ for some $\alpha' \in \mathrm{OVL}(t,s)$, and the derivation $R$ is full and single-threaded.

Let us summarize the proof of this section. Note that we do not assume $SOF$.

**Theorem 4.4.** *The system $s \to t$ satisfies 2A if, and only if, there are $d, e, f, g, h, u, v, y, z \in \Sigma^*$, $\alpha, \alpha' \in \mathrm{OVL}(t,s)$, $\beta, \beta' \in \mathrm{OVL}(s,t)$, and full, single-threaded derivations*

$$L : usd \to^+ ytg$$
$$R : fsv \to^+ htz$$

*such that $t = \beta def\alpha$ and $s = \alpha' geh\beta'$.*

*Example* 4.5. The derivation $D$ from the example 2.2 decomposes as follows:

$$L : aabbabb \to^+ abbabbaabba$$
$$R : aabbabb \to^+ abbabbaabba$$

$$\beta = ab \quad u = \varepsilon \quad d = b \quad y = abb \quad g = \varepsilon \quad f = \varepsilon \quad v = b$$
$$h = abb \quad z = \varepsilon \quad e = \varepsilon \quad \beta = ab \quad \alpha = aabba \quad \alpha' = a \quad \beta' = ab$$

## 4.2 A Hierarchy

With each 1STS we can associate a hierarchy of sets, which comprises the possible components of derivations. Moreover, it is useful in the proof of decidability of the condition 2A.

**Definition 4.6.** For $i \geq 0$, the sets $S_i$ and $T_i$ for the system $s \to t$ are defined as follows:

$$\begin{aligned} T_0 &= \{t_\alpha \mid \alpha \in \mathrm{OVL}(s,t)\} \\ S_0 &= \{s_\alpha \mid \alpha \in \mathrm{OVL}(t,s)\} \\ T_n &= \{v \mid t = \beta wv, \beta \in \mathrm{OVL}(s,t), w \in S_{n-1}^*\} \\ S_n &= \{v \mid s = \alpha wv, \alpha \in \mathrm{OVL}(t,s), w \in T_n^*\} \end{aligned}$$

The sets $S$ and $T$ are defined by $S = \bigcup_{i=0}^{\infty} S_n$ and $T = \bigcup_{i=0}^{\infty} T_n$.

The intuitive meaning of these sets is: An element of $S$ is a proper suffix of $s$ that may be consumed during a derivation. An element of $T$ is a proper suffix of $t$ that may be introduced during a derivation.

**Lemma 4.7.** *For all $i > 0$, we have $T_{i-1} \subseteq T_i \subseteq T \subseteq \mathrm{Suf}(t)$ and $S_{i-1} \subseteq S_i \subseteq S \subseteq \mathrm{Suf}(s)$.*

Since $\mathrm{Suf}(t)$ and $\mathrm{Suf}(s)$ are finite, there exists $i$ such that for all $j > 0$, $S_{i+j} = S_i \wedge T_{i+j} = T_i$. Let $k$ be the least such $i$. By definition then $S = S_k$ and $T = T_k$.

**Definition 4.8.** We call $k+1$ the complication level (col) of the system $s \to t$.

Sets $S$ and $T$ are computable from the system $s \to t$. We only mention at this point that there is a dual hierarchy for prefixes.

*Example* 4.9. Consider the system $aabbab \to abbaabba$ from Example 2.2. Then:

$$\begin{array}{rcl} T_0 & = & \{baabba\} \\ S_0 & = & \{abbab, b\} \\ T_1 & = & \{baabba, aabba\} \\ S_1 & = & S_0 \\ T_2 & = & T_1 \end{array}$$

The complication level of this system is 2. After the second step in the example derivation, only $baabba \in T$ remains from the contractum.

**Lemma 4.10.** *If SOF holds then the empty string is neither in $S$ nor in $T$.*

*Proof.* We show only the proof for $T$; the proof for $S$ is similar. If $\varepsilon \in T$ then $t = s_1 s_2 \ldots s_n$ for some $s_1, \ldots, s_n \in \mathrm{Suf}(s)$. Let $\alpha \in \mathrm{OVL}(t, s)$. Then there exist $x \in \Sigma^*, y \in \Sigma^+$ and $1 \leq i \leq n$ such that $t = s_1 \ldots s_i x \alpha$ and $xy = s_{i+1}$. The word $y$ is a suffix of $s_{i+1}$ and so of $s$, and it is a prefix of $\alpha$ and so of $s$. Hence $y \in \mathrm{OVL}(s, s)$, contradicting $SOF$. □

### 4.3 Justification Forests

We will apply the hierarchy in the proof of decidability of the condition $2A$. First we have to address the question whether a string in $S$ or $T$ can actually be consumed or introduced, respectively, by a derivation. We introduce the notion of justification tree to exhibit why a string is in $S$ or in $T$, and to establish a correspondence between justification trees and derivations that may consume or introduce that string in the way prescribed.

**Definition 4.11.** We will define the set $JT$ of the *justification trees* for our system. The nodes in the trees will be pairs $(x, y)$ such that $x \in S \cup T$, $y \in \{S, T\}$. The possible values of $y$ will serve only as labels. The set $JT$ is defined as the smallest set such that:

- For each $s_\alpha \in S_0$, the tree comprised of the single node $(s_\alpha, S)$ is in $JT$.
- For each $t_\alpha \in T_0$, the tree comprised of the single node $(t_\alpha, T)$ is in $JT$.
- If $\alpha \in \mathrm{OVL}(t, s)$, $s = \alpha t_1 t_2 \ldots t_n s'$, and $J_1, J_2, \ldots, J_n$ are justification trees whose roots are $(t_1, T), (t_2, T), \ldots, (t_n, T)$, respectively, then the tree with the root $(s', S)$ and children $J_1, J_2, \ldots, J_n$ is in $JT$.

- If $\beta \in \mathrm{OVL}(s,t)$, $t = \beta s_1 s_2 \ldots s_n t'$, and $J_1, J_2, \ldots, J_n$ are justification trees whose roots are $(s_1, S), (s_2, S), \ldots, (s_n, S)$, respectively, then the tree with the root $(t', T)$ and children $J_1, J_2, \ldots, J_n$ is in $JT$.

**Definition 4.12.** An *S-tree* is a justification tree with the root $(z, S)$ for some $z \in \Sigma^+$. An *S-forest* is a non-empty, finite sequence of S-trees. A *T-tree* is a justification tree with root $(z, T)$ for some $z \in \Sigma^+$. A *T-forest* is a non-empty, finite sequence of T-trees. A *justification forest* is an S-forest or a T-forest.

**Definition 4.13.** For a justification tree $X$ with the root $(x, Y)$ we define $W(X)$ as the word $x$. For a justification forest $F = (J_1, \ldots, J_n)$ we define $W(F)$ as the word $W(J_1)W(J_2)\ldots W(J_n)$.

In a manner of speaking, the justification tree $Q$ is a proof that $W(Q)$ is in the respective set ($S$ or $T$). The correspondence is established in the following lemma:

$$\begin{tikzcd} (aabba, T) \arrow[d] \\ (b, S) \end{tikzcd}$$

Figure 4: A justification tree for the word *aabba* from example 4.9

**Lemma 4.14.** *For all $x \in \Sigma^*$ we have:*

1. *$x \in S$ if and only if there is some S-tree $F$ such that $W(F) = x$.*
2. *$x \in T$ if and only if there is some T-tree $F$ such that $W(F) = x$.*

*Proof.* We prove the "if" part by structural induction on $F$ and the "only if" part by induction on the proof that $x$ is the element of the respective set. □

But more importantly, some of the justification forests F correspond to single-threaded derivations. T-forests correspond to derivations $xt \to^+ ytz$, where $z = W(F)$. S-forests correspond to derivations $utv \to^+ wt$, where $u = W(F)$. The correspondence will be established in the following definition.

**Definition 4.15.** The set $DF$ is the smallest set consisting of pairs $(F, D)$, where $F$ is a justification forest and $D$ is a derivation, such that:

1. If $\alpha \in \mathrm{OVL}(t,s)$, then $((J), ts_\alpha \to t_\alpha t) \in DF$ where $J$ is comprised of the single node $(s_\alpha, S)$.
2. If $\alpha \in \mathrm{OVL}(s,t)$, then $((J), s_\alpha t \to tt_\alpha) \in DF$ where $J$ is comprised of the single node $(t_\alpha, T)$.
3. If $((J_1, J_2, \ldots, J_n), utv \to^+ wt) \in DF$, and $((J'), u'tv' \to^+ w't) \in DF$, then:
   (a) If $u' = w''w$ for some $w''$, then $((J_1, \ldots, J_n, J'), D) \in DF$ where $D : w''utvv' \to^+ w''wtv' = u'tv' \to^+ w't$.
   (b) If $w = w''u'$ for some $w''$, then $((J_1, \ldots, J_n, J'), D) \in DF$ where $D : utvv' \to^+ wtv' = w''u'tv' \to^+ w''w't$.
4. If $((J_1, J_2, \ldots, J_n), xt \to^+ ytz) \in DF$, and $((J'), x't \to^+ y'tz') \in DF$, then:

(a) If $y' = y''x$ for some $y''$, then $((J_1, \ldots, J_n, J'), D) \in DF$ where $D : x't \rightarrow^+ y'tz' = y''xtz' \rightarrow^+ y''ytzz'$.

(b) If $x = y''y'$ for some $y''$, then $((J_1, \ldots, J_n, J'), D) \in DF$ where $D : y''x't \rightarrow^+ y''y'tz' = xtz' \rightarrow^+ xytzz'$.

5. Let $\alpha \in \mathrm{OVL}(t, s)$, $((J_1, J_2, \ldots, J_n), xt \rightarrow^+ ytz) \in DF$ and $s_\alpha = zs'$ for some $s'$. The derivation $D$ is defined by $D : xts' \rightarrow^+ ytzs' = yts_\alpha \rightarrow yt_\alpha t$. The tree $J$ is defined to comprise a root $(s', S)$ and children $(J_1, \ldots, J_n)$. Then $((J), D) \in DF$, provided that $J$ is a justification tree.

6. Let $\alpha \in \mathrm{OVL}(s, t)$, $((J_1, J_2, \ldots, J_n), utv \rightarrow^+ wt) \in DF$ and $t_\alpha = vt'$ [2] for some $t'$. The derivation $D$ is defined as: $D : us_\alpha t \rightarrow utt_\alpha = utvt' \rightarrow^+ wtt'$. The tree $J$ is defined to comprise a root $(t', T)$ and children $(J_1, \ldots, J_n)$. Then $((J), D) \in DF$, provided that $J$ is a justification tree.

**Definition 4.16.** A forest $F$ *describes* a derivation $D$, if $(F, D) \in DF$.

We do not know whether or not all justification forests describe derivations.

Now let us try to convey the meaning of $DF$. First we note that $DF$ is a partial function, and that the problem

**Given:** a justification forest $F$
**Wanted:** is there a derivation $D$ such that $(F, D) \in DF$?

is decidable, and its solutions $D$ are computable. We may so understand $F$ as a potential representative of $D$. The next three lemmas make precise the correspondence between justification forests and the derivations they may describe.

**Lemma 4.17.** *If $(F, utv \rightarrow^+ wt) \in DF$ then $F$ is an $S$-forest and $W(F) = v$. If $(F, xt \rightarrow^+ ytz) \in DF$ then $F$ is a $T$-forest and $W(F) = z$.*

*Proof.* By induction on the proof that $(F, D) \in DF$. □

By Lemma 4.17, the trees $J$ in Cases 5 and 6 are always justification trees.

**Lemma 4.18.** *If SOF holds then there is no full derivation $d_0 = xt \rightarrow^+ d_n$.*

*Proof.* Suppose that there is a full derivation. Then all inner positions from the rightmost $t$ in $d_0$ are consumed from the left and that hence $\varepsilon \in T$, in contradiction to Lemma 4.10. □

**Lemma 4.19.** *Let SOF hold. Every full, single-threaded derivation $utv \rightarrow^+ wt$ is described by some $S$-forest. Every full, single-threaded derivation $xt \rightarrow^+ ytz$ is described by some $T$-forest.*

*Proof.* By induction on the length, $n$, of the derivation. If $n = 1$, then $v = s_\alpha$ for some $\alpha \in \mathrm{OVL}(t, s)$, and the forest comprised of the single node $(s_\alpha, S)$ describes $ts_\alpha \rightarrow t_\alpha t$. Likewise, $z = t_\beta$ for some $\beta \in \mathrm{OVL}(s, t)$, and the forest comprised of the single node $(t_\beta, T)$ describes $s_\beta t \rightarrow tt_\beta$.

Suppose now that the claim holds for all derivations of length less than $n$.

[2] Note that this implies that $v \neq \varepsilon$.

- Let $s'$ be the prefix of $v$ which is consumed first in one step during $D : utv \to^n wt$. Then $v = s'v'$ for some $v'$, and $D$ can be split into 3 parts:

$$\begin{aligned} D' &= & ut &\to^i y'tz' \\ D'' &= & y'tz's' = u's &\to u't \\ D''' &= & u'tv' &\to^j wt \end{aligned}$$

  We choose $y'$ and $z'$ so that $t$ in $y'tz'$ is created in the $i$-th step. Since $i < n$, from the inductive hypothesis we get some T-forest $G'$ describing $D'$. Since $D$ is single-threaded, we get $tz's' = t_\alpha \alpha s_\alpha$ for some $\alpha \in \mathrm{OVL}(t, s)$. So $u' = y't_\alpha$ and $s_\alpha = z's'$. By definition of $DF$ the S-tree $J$, comprising the root $(s', S)$ and the children $G'$, describes the derivation $uts' \to^{i+1} u't$. Since $j < n$, then from the inductive hypothesis we get a T-forest $(G_1, \ldots, G_m)$ describing the derivation $D'''$. From the definition of $DF$ we get the forest $(G_1, \ldots, G_m, J)$ describing $D$.

- Let $t'$ denote the longest suffix of $t$ that is untouched during the derivation $E = xt \to^n ytz$. The string $t'$ is non-empty; otherwise the string $x't$ for some $x'$ occurs during $E$ in contradiction to Lemma 4.18. We have $z = z't'$. Since $t'$ is maximal, there is a reduction step $e_i = wst' \to wtt' = e_{i+1}$.

  If $i = 0$ then $t' = t_\alpha$ for some $\alpha \in \mathrm{OVL}(s, t)$, and we have a derivation $wt \to^{n-1} ytz'$, which by inductive hypothesis is described by some T-forest $(H_1, \ldots, H_m)$. Since the T-tree $J$, comprising the single node $(t_\alpha, T)$, describes the step $e_0 \to e_1$, the forest $(H_1, \ldots, H_m, J)$ describes $E$.

  If $i > 0$ then let $e_0 = xt = usvt' \to utvt' = e_1$. We have then two derivations: $utv \to^+ wt$ and $wt \to^+ ytz'$. Both are shorter and give us, by inductive hypothesis, an S-forest $H$ describing the former derivation, and a T-forest $(I_1, \ldots, I_m)$ describing the latter. By definition of $DF$, the T-tree $K$, rooted by $(t', T)$ and having the children $H$, describes the derivation $e_0 = xt \to^+ wtt'$. We conclude that the forest $(I_1, \ldots, I_m, K)$ describes the derivation $E$.

□

## 4.4 Solving Certain Derivability Problems

Finally we use the framework of justification forests to solve the derivability problems in Theorem 4.4.

**Definition 4.20.** The problems

> **Given:** $v \in \Sigma^+$
> **Question:** Are there $u, w \in \Sigma^*$ such that there is a single-threaded derivation $utv \to^+ wt$ that consumes all of $utv$?

and

> **Given:** $z \in \Sigma^+$
> **Question:** Are there $x, y \in \Sigma^*$ such that there is a single-threaded derivation $xt \to^+ ytz$ that produces all of $ytz$?

are denoted by $\underline{u}tv \to^+ \underline{w}t$ and $\underline{x}t \to^+ yt\underline{z}$, respectively. The problem

> **Given:** $v, z \in \Sigma^+$
> **Question:** Are there $u, y \in \Sigma^*$ such that there is a single-threaded derivation $utv \to^+ ytz$ that consumes all of $utv$ and produces all of $ytz$?

| $2A$ | $2B$ | $2C$ | Status | Comments |
|---|---|---|---|---|
| 0 | 0 | 0 | decidable | well-behaved [McN97] |
| 0 | 1 | 0 | decidable | Theorem 3.32 |
| 0 | 0 | 1 | decidable | Theorem 3.32 |
| 0 | 1 | 1 | open | impossibility conjectured [KKST01] |
| 1 | $*$ | $*$ | open | |

Figure 5: Overview of the uniform termination problem of self-overlap-free 1STS

is denoted by $\underline{u}tv \to^+ \underline{y}tz$.

**Lemma 4.21.** *The problems $\underline{u}tv \to^+ \underline{w}t$ and $\underline{x}t \to^+ \underline{y}tz$ are decidable, and their sets of solutions, $(u, w)$ and $(x, y)$, respectively, are computable.*

*Proof.* If $\underline{u}tv \to^+ \underline{w}t$ has a solution $(u, w)$ then there is a single-threaded derivation $utv \to^+ wt$ that consumes all of $utv$. By Lemma 4.19, this derivation is described by some justification forest $F$. By Lemma 4.17, $W(F) = v$.

There are only finitely many justification forests $F$ such that $W(F) = v$, so we enumerate them all. For each such $F$, we check whether $F$ describes a derivation $utv \to^+ wt$ for any $(u, w)$. In case of success, we can extract the solution $(u, w)$. □

**Lemma 4.22.** *The problem $\underline{u}tv \to^+ \underline{y}tz$ is decidable, and its set of solutions $(u, y)$ is computable.*

*Proof.* Suppose that $D : utv \to^+ ytz$ is a single-threaded derivation that consumes all of $utv$ and introduces all of $ytz$. Since $utv$ is consumed, there is a step $ws \to wt$ during $D$. We can split $D$ into $D' : utv \to^+ wt$ and $D'' : wt \to^+ ytz$. Now let $u''$ be the longest prefix of $utv$ that stays untouched during $D'$. In other words, there are $u', w' \in \Sigma^*$ such that $u = u''u'$, $w = u''w'$, and $u'tv \to^+ w't$ where all of $u'tv$ is consumed. Hence the problem $\underline{u}tv \to^+ \underline{w}t$ has the solution $(u', w')$. Likewise let $y''$ be the longest prefix of $ytz$ that stays untouched during $D''$. Then the problem $\underline{x}t \to^+ \underline{y}tz$ has the solution $(x', y')$. Then $D$ is of the form

$$u''u'tv \to^+ u''w't = y''x't \to^+ y''y'tz.$$

Now the shorter of $u'', y''$ is a prefix of the longer of $u'', y''$. If both $u''$ and $y''$ are non-empty strings then there is a non-empty prefix of $u''u'tv$ which is never consumed, a contradiction to our premise. So either $u''$ is empty, or $y''$ is empty. In the former case we use the equation $w' = y''x'$ to compute $y''$ from $w', x'$; in the latter case we use the equation $u''w' = x'$ to compute $u''$ from $w', x'$. Either way we get the solution $u = u''u', y = y''y'$. □

**Theorem 4.23.** *The Condition $2A$ is decidable for SOF 1SRSs.*

*Proof.* By Lemma 4.4 we reduce $2A$. We enumerate all $\alpha, \alpha' \in \mathrm{OVL}(t, s)$, $\beta, \beta' \in \mathrm{OVL}(s, t)$, $d, f, g, h \in \Sigma^+$ such that $t = \beta d e f \alpha$ and $s = \alpha' g e h \beta'$. Then we solve the problems $\underline{u}td \to^+ \underline{y}tg$ and $ft\underline{v} \to^+ ht\underline{z}$ using Lemma 4.22 and its dual, respectively. □

## 5 Conclusion

We have shown that one-rule Semi-Thue Systems (1STS) that satisfy $\neg 2A \wedge \neg 2C$, according to our classification, can be simulated by pushdown automata. Therefore their uniform termination problem is decidable.

1STSs that satisfy $\neg 2A \wedge \neg 2B$ can be solved in a symmetric way. This leaves the case $\neg 2A \wedge 2B \wedge 2C$ which Kobayashi et al. conjecture to be empty [KKST01, page 603], and the widely unexplored $2A$ region which contains the gentle and tame 1STS among others.

**Acknowledgements:** We are grateful to Robert McNaughton for his encouraging us to do this investigation. The first of the authors is grateful to Paweł Urzyczyn for his infinite patience.

## References

[Ges02a] Alfons Geser. Decidability of termination of grid string rewriting rules. *SIAM J. Comput.*, 31(4):1156–1168, 2002.

[Ges02b] Alfons Geser. *Is termination decidable for string rewriting with only one rule?* Habilitation thesis, Wilhelm-Schickard-Institut, Universität Tübingen, Germany, January 2002. 201 pages.

[Ges02c] Alfons Geser. Loops of superexponential lengths in one-rule string rewriting. In Sophie Tison, editor, *Proc. Rewriting Techniques and Applications*, LNCS 2378, pages 267–280. Springer, 2002.

[GKM83] J. V. Guttag, Deepak Kapur, and David R. Musser. On proving uniform termination and restricted termination of rewriting systems. *SIAM J. Comput.*, 12:189–214, 1983.

[GZ99] Alfons Geser and Hans Zantema. Non-looping string rewriting. *Theoret. Informatics Appl.*, 33(3):279–301, 1999.

[KKST01] 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.

[Kur90] Winfried Kurth. *Termination und Konfluenz von Semi-Thue-Systemen mit nur einer Regel.* Dissertation, Technische Universität Clausthal, Germany, 1990.

[McN94] 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.

[McN95] 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.

[McN97] Robert McNaughton. Semi-Thue Systems with an Inhibitor. *J. Automated Reasoning*, 26:409–431, 1997.

[Moc02] Wojciech Moczydłowski, Jr. Jednoregułowe systemy przepisywania słów. Masters thesis, Warsaw University, Poland, 2002.

[MS96] Yuri Matiyasevitch and Geraud Sénizergues. Decision problems for semi-Thue systems with a few rules. In *Proc. Logic in Computer Science*, pages 523–531, New Brunswick, NJ, July 1996. IEEE Computer Society Press.

[Sal69] Arto Salomaa. *Theory of Automata*, volume 100 of *Intl. Series of Monographs in Pure and Applied Mathematics*. Pergamon Press, 1969.

[Sén96] Geraud Sénizergues. On the termination problem for one-rule Semi-Thue Systems. In Harald Ganzinger, editor, *Proc. Rewriting Techniques and Applications*, LNCS 1103, pages 302–316. Springer, 1996.

[STKK97] Kayoko Shikishima-Tsuji, Masashi Katsura, and Yuji Kobayashi. On termination of confluent one-rule string rewriting systems. *Information Processing Letters*, 61(2):91–96, 1997.

[Tuc80] 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.

[ZG00] Hans Zantema and Alfons Geser. A complete characterization of termination of $0^p1^q \to 1^r0^s$. *Applicable Algebra in Engineering, Communication, and Computing*, 11(1):1–25, 2000.