JOURNAL OF COMPUTER AND SYSTEM SCIENCES 35, 285–310 (1987)

On Deciding the Confluence of a Finite String-Rewriting System on a Given Congruence Class*

FRIEDRICH OTTO†

Fachbereich Informatik, Universität Kaiserslautern, Postfach 3049, 6750 Kaiserslautern, Federal Republic of Germany

Revised July 10, 1987

In general it is undecidable whether or not a given finite string-rewriting system R is confluent on a given congruence class $[w]_R$, even when only length-reducing systems are considered. However, for finite monadic string-rewriting systems this problem becomes decidable in double exponential time. For certain subclasses of monadic string-rewriting systems algorithms of lower complexity are obtained for solving this problem.

INTRODUCTION

String-rewriting systems, also known as *semi-Thue systems*, have extensively been studied in computability theory, formal language theory, and combinatorial (semi-) group theory. A string-rewriting system R on alphabet Σ induces a *congruence* $\leftrightarrow_R^*$ on the *free monoid* Σ^* generated by Σ, and hence, the set M_R of congruence classes modulo $\leftrightarrow_R^*$ is a *monoid*. Accordingly, the ordered pair $(\Sigma; R)$ is called a *monoid-presentation* of M_R.

The fundamental decision problem associated with R is the *word problem*:

INSTANCE. Two words $u, v \in \Sigma^*$.

Question. Are u and v congruent modulo R?

It is well known that this problem is undecidable in general.

A string-rewriting system R is called *Noetherian*, if there is no infinite sequence of reductions modulo R. So if R is Noetherian, then for each word w, an irreducible word w_0 is reached within a finite number of reduction steps. Here a word is called *irreducible* if no reduction applies to it. R is called *confluent* if, for all words w_1, w_2, w_3, whenever w_1 reduces to w_2 and to w_3, then w_2 and w_3 can be reduced to a common word w_4. This property is equivalent to the *Church–Rosser property*: for all words w_1 and w_2, if w_1 and w_2 are congruent modulo R, then they can be

* This work was partially supported by a Faculty Research Award from SUNY Albany.

† Currently visiting at Department of Computer Science, State University of New York, 1400 Washington Avenue, Albany, New York 12222.

0022-0000/87 $3.00

reduced to a common word w_3. In particular, this means that no two distinct irreducible words are congruent modulo R. Thus, if R is Noetherian and confluent, then each congruence class contains a unique irreducible word, which can be taken as a *normal form* of this class. String-rewriting systems of this kind are called *complete* or *canonical*. Obviously, the word problem for a complete string-rewriting system is decidable, provided the process of reduction can be performed effectively.

Unfortunately, it is undecidable in general whether a given finite string-rewriting system R is Noetherian [13] or whether it is confluent [2]. However, if R is Noetherian, then it is decidable whether or not R is confluent by checking the *critical pairs* of R [14]. If R is not confluent, i.e., there exists a critical pair (u, v) of words such that u and v cannot be reduced to a common word w, one can try to obtain a complete system R' generating the same congruence as R by introducing $u \to v$ or $v \to u$ as an additional rule. This is essentially the idea behind the Knuth–Bendix completion procedure [18]. Obviously, not every finite string-rewriting system can be completed in a finite number of steps. In fact, there are examples of finite string-rewriting systems that are not equivalent to any finite complete system, although they have decidable word problem [15, 17, 30].

On the other hand, there are situations in which only certain restricted instances of the word problem are of interest. For example, one might be interested in characterizing the languages $[e]_R$ or $[a]_R$ $(a \in \Sigma)$. Another example is the word problem for a finitely presented group. Let Σ be a finite alphabet corresponding to a set of generators of a group G, let Σ^- be an alphabet in one-to-one correspondence to Σ such that $\Sigma \cap \Sigma^- = \emptyset$, let $\mathbf{\Sigma} := \Sigma \cup \Sigma^-$, and let R be a finite *string-rewriting system on* $\mathbf{\Sigma}$ corresponding to a set of defining relations of G [22]. Since the letter $a^- \in \Sigma^-$ is to be considered as the formal inverse of the letter $a \in \Sigma$, we assume that the following set of trivial relations $\{(aa^-, e), (a^-a, e) \mid a \in \Sigma\}$ is a subset of R. Further, the function $^{-1}\colon \mathbf{\Sigma}^* \to \mathbf{\Sigma}^*$, defined by $e^{-1} := e$, $(wa)^{-1} := a^- w^{-1}$, $(wa^-)^{-1} := aw^{-1}$ $(w \in \mathbf{\Sigma}^*, a \in \Sigma, a^- \in \Sigma^-)$, yields the *formal inverse* w^{-1} of the word w for each $w \in \mathbf{\Sigma}^*$. Here e denotes the empty word, which represents the identity of the group G.

Now two words $u, v \in \mathbf{\Sigma}^*$ describe the same element of G, i.e., $u =_G v$, if and only if $u^{-1}v =_G e$. Thus, in order to solve the word problem for G it suffices to solve the membership problem for the congruence class $[e]_R$. If R is Noetherian, then each word $w \in \mathbf{\Sigma}^*$ is reduced by R to some irreducible descendant. If, in addition, R is confluent on $[e]_R$, then $[e]_R$ contains one irreducible word only, which necessarily must be the empty word e itself. Hence, given $u, v \in \mathbf{\Sigma}^*$, we see that $u =_G v$ if and only if e is the irreducible descendant of the word $u^{-1}v$. For example, Dehn's algorithm for the word problem, which applies to certain small cancellation groups [19, 20], is of this form. Bücken and Le Chenadec [10, 19] investigated how certain restrictions upon the finite set of relations R do translate into a proof that R is confluent on $[e]_R$. These proofs are technically rather involved, thus raising the question of whether there exists a uniform way of deciding whether a given finite string-rewriting system R is confluent on the congruence class of a given word w. So instead of dealing with the global notion of confluence, one might restrict one's

attention to the more local notion of confluence on a given congruence class. This is the goal of the present paper.

We should remark that in [11] a restriction of the same kind can be found for the problem of deciding whether a given finite term-rewriting system R that is globally finite is Noetherian.

After establishing the basic definitions and notation in Section 1, we present a necessary and sufficient condition for a finite Noetherian string-rewriting system R to be confluent on a congruence class $[w]_R$ in Section 2. Then in Section 3 we prove that this problem is undecidable in general, even if only length-reducing systems are considered. In fact, there exists a fixed finite length-reducing string-rewriting system R such that it is undecidable whether R is confluent on $[w]_R$ given a word $w \in \Sigma^*$. Instead of fixing the finite string-rewriting system R we can also fix the word w to always be the empty word, and still our problem remains undecidable.

Due to the results of Section 3, the necessary and sufficient condition for a finite Noetherian string-rewriting system R to be confluent on a congruence class $[w]_R$, that is established in Section 2, is undecidable in general. However, if the systems R under consideration are monadic, i.e., the right-hand side of each rule is of length at most one, then our problem can be reduced to the equivalence problem of a special class of deterministic finite-turn pushdown automata, and therewith it becomes decidable [32]. In fact, this approach gives an algorithm of double exponential time complexity [3] in Section 4.

Notice that this result only gives an upper bound for the complexity of the problem considered. We currently do not know how good this upper bound is. In particular, we do not know about a corresponding lower bound.

In Section 5 we consider finite special string-rewriting systems, i.e., systems for which the right-hand side of each rule is the empty word. To characterize those finite special string-rewriting systems R that are confluent on $[e]_R$ a weakened version of the condition derived in Section 2 is sufficient. Since this condition can be checked in exponential time, it turns out that it is decidable in exponential time whether or not a given finite special string-rewriting system R is confluent on $[e]_R$. Moreover, if R is confluent on some congruence class $[w]_R$, then $[e]_R$ contains a finite number of irreducible words only, implying that the special word problem for R is decidable in linear time.

The final two sections are dedicated to the study of the distinguished role $[e]_R$ plays among all congruence classes of a finite special string-rewriting system R. In Section 6 we consider finite string-rewriting systems R that are homogeneous of degree 2, i.e., each rule of R is of the form (l, e) for some word l of length 2. For a system R of this kind it can be checked in polynomial time whether or not R is confluent on $[e]_R$. Further, if R is confluent on some congruence class $[w]_R$, then R is confluent on $[e]_R$ as well. However, this result does not generalize to finite string-rewriting systems that are homogeneous of degree 3.

Finally, in Section 7 we look at special string-rewriting systems R containing a single rule only, i.e., $R = \{(l, e)\}$ for some non-empty word l. Book [6] has shown

that a system of this form is equivalent to a finite length-reducing and confluent system if and only if R is confluent, which in turn holds if and only if the root $\rho(l)$ of l has no overlap. Here we prove that R is confluent if and only if it is confluent on some congruence class $[w]_R$. So for a special one-rule system R, confluence of R on an arbitrary congruence class already yields that R is a confluent system. An example finally indicates that a corresponding result does not hold for one-rule systems that are length-reducing but not special.

A problem that we do not discuss here, but which is clearly related to our investigations, is that of finding a completion procedure for completing a given string-rewriting system with respect to a given congruence class.

Further, we would like to point out that the notion of confluence on a particular congruence class is also of interest in the more general setting of term-rewriting systems [12, 14]. Here, also other notions of restricted confluence are of interest, e.g., the notion of ground confluence, i.e., confluence on the set of all ground terms.

1. Basic Definitions and Notation

Here the formal definitions of string-rewriting systems and their various properties are given that we shall need throughout this paper. We do not go into any more detail than seems appropriate for our purposes. For additional information and comments on the various notions introduced the reader is referred to the literature, in particular [5, 7, 14].

An *alphabet* Σ is a set the elements of which are called *letters*. Here we are only dealing with finite alphabets. For an alphabet Σ, Σ^* denoted the set of *words* over Σ including the *empty word* e. Thus, Σ^* is the free monoid generated by Σ under the operation of concatenation with e as identity. The set $\Sigma^* - \{e\}$ of all non-empty words over Σ is denoted by Σ^+. For $w \in \Sigma^*$, the *length* of w, denoted by $|w|$, is defined by $|e| = 0$, and $|wa| = |w| + 1$ for all $w \in \Sigma^*$, $a \in \Sigma$. For $a \in \Sigma$, the *a-length* of w, denoted by $|w|_a$, is the number of occurrences of the letter a in w. As usual the *concatenation* of words u and v is simply written as uv, and numerical superscripts are often used to abbreviate words, i.e., for all $w \in \Sigma^*$, $w^0 = e$, and $w^{n+1} = ww^n$ for $n \in \mathbf{N}$. Here $\mathbf{N}$ denotes the set of all non-negative integers.

A *string-rewriting system* R on Σ is a subset of $\Sigma^* \times \Sigma^*$, the elements of which are called (*rewrite*) *rules*. Usually they are written as pairs (l, r), but sometimes we shall also write them as $l \to r$ to increase readability. For a string-rewriting system R on Σ, $\mathrm{dom}(R) = \{l \mid \exists r \in \Sigma^*\colon\ (l, r) \in R\}$ is the *domain* of R, and $\mathrm{range}(R) = \{r \mid \exists l \in \Sigma^*\colon (l, r) \in R\}$ is its *range*. The *single step reduction relation* $\to_R$ induced by R is the following relation over Σ^*: for $u, v \in \Sigma^*$, $u \to_R v$ if and only if $\exists x, y \in \Sigma^*$, $(l, r) \in R$: $u = xly$ and $v = xry$. So $u \to_R v$ holds if and only if u can be transformed into v by a single application of a rule of R. The *reduction relation* $\to_R^*$ induced by R is the reflexive transitive closure of the relation $\to_R$. By $\to_R^+$ we denote the transitive closure of $\to_R$, by $\leftrightarrow_R$ we denote the symmetric closure of $\to_R$, and $\leftrightarrow_R^*$ denotes the equivalence relation generated by $\to_R$. Note that $\leftrightarrow_R^*$ is a con-

gruence on Σ^*, i.e., it is compatible with the operation of concatenation. Hence, if $u, v \in \Sigma^*$ satisfy $u \leftrightarrow_R^* v$, one says that u and v are *congruent* (*modulo* R). The *congruence class* $\{v \in \Sigma^* | u \leftrightarrow_R^* v\}$ of u is denoted by $[u]_R$.

If $u \to_R^* v$ one says that u *reduces to* v, u is an *ancestor* of v, and v is a *descendant* of u (modulo R). If a word u has no descendant except itself, then it is *irreducible*, otherwise, it is *reducible* (modulo R). $\mathrm{IRR}(R)$ denotes the set of all irreducible words. For $u \in \Sigma^*$, $\langle u \rangle_R = \{w \in \Sigma^* | w \to_R^* u\}$ is the *set of ancestors* of u, and $\Delta_R^*(u) = \{v \in \Sigma^* | u \to_R^* v\}$ is the *set of descendants* of u. Finally, for a language $L \subseteq \Sigma^*$,

$$[L]_R = \bigcup_{u \in L} [u]_R, \quad \langle L \rangle_R = \bigcup_{u \in L} \langle u \rangle_R, \quad \text{and} \quad \Delta_R^*(L) = \bigcup_{u \in L} \Delta_R^*(u).$$

A string-rewriting system R on Σ is called

— *Noetherian* if there is no infinite sequence of reductions $u_1 \to_R u_2 \to_R \cdots$;

— *locally confluent* if, for all $u, v, w \in \Sigma^*$, $u \to_R v$ and $u \to_R w$ imply that there exists some $z \in \Sigma^*$ such that $v \to_R^* z$ and $w \to_R^* z$;

— *confluent* if, for all $u, v, w \in \Sigma^*$, $u \to_R^* v$ and $u \to_R^* w$ imply that there exists some $z \in \Sigma^*$ such that $v \to_R^* z$ and $w \to_R^* z$;

— *complete* if it is Noetherian and confluent.

It is undecidable in general whether or not a given finite string-rewriting system R is Noetherian [13] or confluent [2]. On the other hand, if R is Noetherian, then R is confluent if and only if it is locally confluent [25].

Let R be a string-rewriting system on Σ. If there are (not necessarily distinct) rules $(l_1, r_1), (l_2, r_2) \in R$ such that $l_1 = x l_2 y$ for some $x, y \in \Sigma^*$ or $l_1 x = y l_2$ for some $x, y \in \Sigma^*$, $0 < |y| < |l_1|$, then these rules are said to *overlap*. The pair of words $(r_1, x r_2 y)$ or $(r_1 x, y r_2)$, respectively, is then called a *critical pair* of R. We say that a critical pair (u, v) can be *resolved*, if u and v have a common descendant, i.e., $\Delta_R^*(u) \cap \Delta_R^*(v) \neq \emptyset$; otherwise, it is called *unresolvable*. Note that if (u, v) is a critical pair of R, then $u \leftarrow_R w \to_R v$ for some word $w \in \Sigma^*$. Thus, for R to be locally confluent, it is necessary that all critical pairs of R can be resolved. As it turns out this condition is also sufficient [26], and if R is finite and Noetherian, then it is effectively decidable. Hence, in this situation it is decidable whether or not R is confluent.

Let R be a string-rewriting system on Σ, and let $w \in \Sigma^*$. R is called *confluent on* $[w]_R$ if, for all $u, v, x \in [w]_R$, $u \to_R^* v$ and $u \to_R^* x$ imply that $\Delta_R^*(v) \cap \Delta_R^*(x) \neq \emptyset$. If R is Noetherian, then this means that $[w]_R$ contains exactly one irreducible word w_1, which can be considered as the normal form of $[w]_R$. Thus, if R is Noetherian, $w \in \Sigma^*$, and $w_1 \in \Sigma^*$ is an irreducible descendant of w, then R is confluent on $[w]_R$ if and only if $[w_1]_R = \langle w_1 \rangle_R$, i.e., each word that is congruent to w_1 can be reduced to w_1. In this paper we are interested in the following decision problem: *Confluence on a given Congruence Class* (CCC).

INSTANCE. A finite string-rewriting system R on Σ, and a word $w \in \Sigma^*$.

Question. Is R confluent on $[w]_R$?

In what follows we will mainly be dealing with finite length-reducing string-rewriting systems. Here a string-rewriting system R on Σ is called *length-reducing*, if $|l| > |r|$ holds for each rule $(l, r) \in R$. Obviously, a length-reducing string-rewriting system R is Noetherian. Thus, if R is also finite, then it is decidable whether or not R is confluent [9]. In fact, this decision can be made in polynomial time [9, 16].

2. A CHARACTERIZATION THEOREM

Let R be a finite string-rewriting system on Σ. Then $\mathrm{UCP}(R) = \{(u, v) | (u, v)$ is a critical pair of R such that $\Delta_R^*(u) \cap \Delta_R^*(v) = \emptyset\}$ is the *set of unresolvable critical pairs* of R. This set is finite, and if R is Noetherian, then it can effectively be computed from R.

In general, the process of reduction $\to_R$ is inherently ambiguous. To obtain a stronger version of the intended characterization theorem, we consider a restricted notion of reduction.

One source of ambiguity for $\to_R$ is the fact that distinct rules of R may have the same left-hand side. To eliminate these ambiguities we choose a subsystem R_1 of R as follows. Let $<$ be the lexicographic ordering on Σ^* induced by a linear ordering of Σ. For each word $l \in \mathrm{dom}(R)$, let $r(l) := \min\{r \in \Sigma^* | (l, r) \in R\}$, where the minimum is taken with respect to the ordering $<$. Then $R_1 := \{(l, r(l)) | l \in \mathrm{dom}(R)\}$ is a subsystem of R such that $\mathrm{dom}(R_1) = \mathrm{dom}(R)$, which implies $\mathrm{IRR}(R_1) = \mathrm{IRR}(R)$, and distinct rules of R_1 have distinct left-hand sides. If R is Noetherian, then so is R_1, since the reduction relation $\to_{R_1}$ is a restriction of the reduction relation $\to_R$. However, even this restricted reduction relation is still inherently ambiguous. To resolve the remaining ambiguities we consider leftmost reductions.

Here, a reduction $u \to_R v$ is called *leftmost* if $u = xly$, $v = xry$, $(l, r) \in R_1$, and whenever $u = x_1 l_1 y_1$ with $l_1 \in \mathrm{dom}(R)$, then xl is a proper prefix of $x_1 l_1$, or $xl = x_1 l_1$, and x is a proper prefix of x_1, or $x = x_1$ and $l = l_1$. We write $u \to_{R,L} v$ if $u \to_R v$ is left-most, and by $\to_{R,L}^*$ we denote the reflexive transitive closure of $\to_{R,L}$.

Observe that for each reducible word $u \in \Sigma^*$, there exists a unique word $v \in \Sigma^*$ such that $u \to_{R,L} v$. Thus, the process of leftmost reduction is unambiguous. Obviously, the subsystem R_1 used to define the notion of leftmost reduction can effectively be obtained from R and Σ. Further, given a word $w \in \Sigma^*$, one can determine an irreducible word $w_1 \in \Sigma^*$ such that $w \to_{R,L}^* w_1$.

Given three words u, w, $w_1 \in \Sigma^*$ such that $w_1 \in \mathrm{IRR}(R)$ and $w \to_{R,L}^* w_1$, let $L_u(w)$ denote the language $L_u(w) = \{x \# y |\ x, y \in \mathrm{IRR}(R),\ xuy \to_{R,L}^* w_1\}$, where $\#$ is an additional letter not in Σ. Roughly speaking, $x \# y \in L_u(w)$ if (x, y) is an irreducible

context of u in $[w]_R$. Using these sets we can formulate our characterization theorem as follows.

THEOREM 2.1. *Let R be a finite Noetherian string-rewriting system on Σ, and let $w \in \Sigma^*$. Then the following two statements are equivalent:*

(i) *The system R is confluent on $[w]_R$.*

(ii) $\forall (u, v) \in \mathrm{UCP}(R)$: $L_u(w) = L_v(w)$.

Proof. Without loss of generality we assume that $w \in \mathrm{IRR}(R)$, i.e., $L_u(w) = \{x \# y \mid x, y \in \mathrm{IRR}(R), xuy \to_{R,L}^* w\}$.

(i) $\Rightarrow$ (ii) Assume that R is confluent on $[w]_R$, i.e., for all $z \in \Sigma^*$, if $z \leftrightarrow_R^* w$, then $z \to_R^* w$. In fact, we even have $z \to_{R,L}^* w$ in this situation.

Let $(u, v) \in \mathrm{UCP}(R)$. Then $u \neq v$, but $u \leftrightarrow_R^* v$ according to the definition of critical pair. If $x \# y \in L_u(w)$, then $x, y \in \mathrm{IRR}(R)$ and $xuy \to_{R,L}^* w$. Hence, $xuy \leftrightarrow_R^* w$ implying $xvy \leftrightarrow_R^* w$, which in turn yields $xvy \to_{R,L}^* w$. Thus, $x \# y \in L_v(w)$, i.e., $L_u(w) \subseteq L_v(w)$. By symmetry we obtain $L_u(w) = L_v(w)$.

(ii) $\Rightarrow$ (i) Assume that R is not confluent on $[w]_R$.

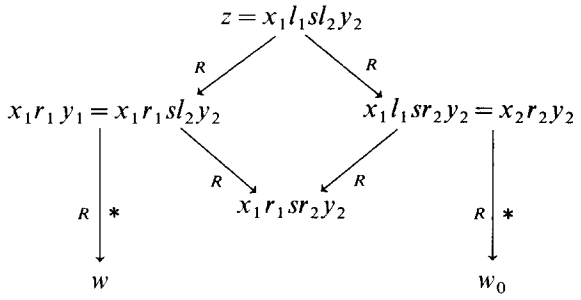
CLAIM 1. There exists a word $z \in \Sigma^*$ such that $\Delta_R^*(z) \cap \mathrm{IRR}(R) \supsetneq \{w\}$.

Proof. Since R is not confluent on $[w]_R$, there is a word $v \in \mathrm{IRR}(R)$ such that $v \neq w$ but $v \leftrightarrow_R^* w$. Hence, there exist an integer $m \geqslant 1$ and words $w_0, w_1, \ldots, w_m \in \Sigma^*$ such that $w = w_0 \leftrightarrow_R w_1 \leftrightarrow_R \cdots \leftrightarrow_R w_m = v$. Since $w, v \in \mathrm{IRR}(R)$, we conclude that $m \geqslant 2$, $w_1 \to_R w_0$ and $w_{m-1} \to_R w_m$. Let $k := \max\{i \mid w_i \to_R^* w\}$. If $k = m-1$, then $\Delta_R^*(w_k) \cap \mathrm{IRR}(R) \supseteq \{w, v\} \supsetneq \{w\}$. If $k < m-1$, then $w_k \to_R^* w$, $w_k \to_R w_{k+1}$, but $w_{k+1} \not\to_R^* w$. Since R is Noetherian, $\Delta_R^*(w_{k+1}) \cap \mathrm{IRR}(R) \neq \emptyset$. Hence, $\Delta_R^*(w_k) \cap \mathrm{IRR}(R) \supseteq \{w\} \cup (\Delta_R^*(w_{k+1}) \cap \mathrm{IRR}(R)) \supsetneq \{w\}$. This proves claim 1. ∎

Since R is Noetherian, we obtain a well-founded ordering $<$ on Σ^* by defining $x > y$ if and only if $x \to_R^+ y$. Let $z \in \Sigma^*$ be a fixed minimal word (with respect to $>$) such that $\Delta_R^*(z) \cap \mathrm{IRR}(R) \supsetneq \{w\}$. Since z is minimal with this property, we see that, whenever $z > z_1$, then either w is the only irreducible descendant of z_1, or w is not a descendant of z_1 at all.

CLAIM 2. *For all factorizations $z = x_1 l_1 y_1 = x_2 l_2 y_2$, where $x_1 l_1 y_1 \to_R x_1 r_1 y_1 \to_R^* w$ and $x_2 l_2 y_2 \to_R x_2 r_2 y_2 \to_R^* w_0$, $w_0 \in \mathrm{IRR}(R) - \{w\}$, the distinguished occurrences of l_1 and l_2 in z overlap, and their overlap yields an unresolvable critical pair $(u, v) \in \mathrm{UCP}(R)$.*

Proof. Since $\Delta_R^*(z) \cap \mathrm{IRR}(R) \supsetneq \{w\}$, z has factorizations of the above form. Now assume that the distinguished occurrences of l_1 and l_2 in z do not overlap, i.e., $|x_1 l_1| \leqslant |x_2|$ or $|x_2 l_2| \leqslant |x_1|$. Without loss of generality we may assume the former. Then there is a word $s \in \Sigma^*$ such that $z = x_1 l_1 s l_2 y_2$. Hence, we have the following situation:



Since $x_1 r_1 y_1 < z$, and since $w \in \Delta_R^*(x_1 r_1 y_1) \cap \mathrm{IRR}(R)$, we conclude that $\Delta_R^*(x_1 r_1 y_1) \cap \mathrm{IRR}(R) = \{w\}$. Thus, $x_1 r_1 s r_2 y_2 \to_R^* w$, implying that $\Delta_R^*(x_2 r_2 y_2) \cap \mathrm{IRR}(R) \supsetneq \{w\}$. Since $x_2 r_2 y_2 < z$, this contradicts the choice of z. Hence, the distinguished occurrences of l_1 and l_2 in z overlap, i.e., $l_1 = s l_2 t$, or $l_1 t = s l_2$ with $0 < |s| < |l_1|$. In the first case we have the critical pair $(r_1, s r_2 t)$, in the latter case we have the critical pair $(r_1 t, s r_2)$. If this critical pair can be resolved, then $x_1 r_1 y_1$ and $x_2 r_2 y_2$ have a common descendant, thus leading to the same contradiction as above. Therefore, the resulting critical pair (u, v) is unresolvable, i.e., $(u, v) \in \mathrm{UCP}(R)$. ∎

Let $z_1 \in \mathrm{IRR}(R)$ such that $z = x_1 l_1 y_1 \to_{R,L} x_1 r_1 y_1 \to_{R,L}^* z_1$, and let $z_2 \in \mathrm{IRR}(R)$ be defined as follows: If $z_1 \neq w$, then $z_2 := w$, otherwise let $z_2 \in \Delta_R^*(z) \cap \mathrm{IRR}(R)$ with $z_2 \neq w$. Then we have the following situation:

$$z = x_1 l_1 y_1 \to_{R,L} x_1 r_1 y_1 \to_{R,L}^* z_1 \in \mathrm{IRR}(R),$$
$$z = x_2 l_2 y_2 \to_R x_2 r_2 y_2 \to_R^* z_2 \in \mathrm{IRR}(R),$$

where $z_1 \neq z_2$ and $w \in \{z_1, z_2\}$.

By Claim 2 the occurrences of l_1 and l_2 in z overlap giving an unresolvable critical pair $(u, v) \in \mathrm{UCP}(R)$, i.e.,

$$z = xsy \to_{R,L} xuy \to_{R,L}^* z_1 \in \mathrm{IRR}(R),$$
$$z = xsy \to_R xvy \to_R^* z_2 \in \mathrm{IRR}(R),$$

where $z_1 \neq z_2$ and $w \in \{z_1, z_2\}$.

In particular, Claim 2 implies that $x, y \in \mathrm{IRR}(R)$. Now we distinguish two cases.

(i) Assume that $z_1 = w$. Then $xuy \to_{R,L}^* z_1 = w$, i.e., $x \# y \in L_u(w)$. However, $x \# y \notin L_v(w)$, since $w \notin \Delta_R^*(xvy)$ due to the choice of z.

(ii) Assume that $z_1 \neq w$. Then $z_2 = w$. Since $xvy < z$, this means that $\Delta_R^*(xvy) \cap \mathrm{IRR}(R) = \{w\}$ implying $xvy \to_{R,L}^* z_2 = w$. Hence, $x \# y \in L_v(w)$. However, $x \# y \notin L_u(w)$, since $xuy \to_{R,L}^* z_1 \neq w$.

Thus, in any case $L_u(w) \neq L_v(w)$. This completes the proof of Theorem 2.1. ∎

Thus, in order to decide whether a given Noetherian string-rewriting system R is confluent on a given congruence class $[w]_R$, it suffices to compare the languages $L_u(w)$ and $L_v(w)$, $(u, v) \in \mathrm{UCP}(R)$. Since the language equivalence problem is undecidable for many classes of languages, it is not clear how helpful this characterization theorem is in practice. The following investigations can be seen as a step towards determining the borderline between those situations for which the criterion given by the characterization theorem is undecidable, respectively intractible, and those situations where it is decidable, respectively tractable.

3. Length-Reducing String-Rewriting Systems

Our next result states that the criterion for confluence on a given congruence class that is given by Theorem 2.1 is undecidable in general, even for length-reducing systems.

THEOREM 3.1. *There exists a finite length-reducing string-rewriting system R on Σ such that the following problem is undecidable*:

INSTANCE. A word $w \in \Sigma^*$.

Question. Is R confluent on $[w]_R$?

The proof of this result is based on a construction taken from Ó'Dúnlaing's dissertation ([27], see also [28, Theorem 4.1.1]), that has also been used in [23, 24, 29] to prove various undecidability results concerning finite length-reducing string-rewriting systems.

Proof. Let Σ be a finite alphabet, let $L \subseteq \Sigma^*$ be a language over Σ, that is recursively enumerable, but nonrecursive, and let $M = (\Sigma, Q, q_0, \delta)$ be a single-tape Turing machine accepting L. Then one can effectively construct a finite length-reducing string-rewriting system $R(M)$ on alphabet $\Gamma \supsetneq \Sigma$ such that $R(M)$ is confluent, and the reduction sequences induced by $R(M)$ correspond to computations of M. Thus, the string-rewriting system $R(M)$ simulates the Turing machine M in a certain way.

Let $\Sigma_b := \Sigma \cup \{b\}$, where b denotes the *blank symbol* used by M, let $\Pi := \Sigma_b \cup \{¢, \$\}$, where $¢$ and $\$$ are additional new letters used as *markers*, let $Q_1 := Q_p \cup Q_s$, where $Q_P = \{p_0, p_1, ..., p_n\}$ and $Q_s = \{s_0, s_1, ..., s_n\}$ are disjoint copies of the set Q of states of M, and let $D := \{\langle ap_i\rangle, \langle s_i a\rangle \mid a \in \Pi, i \in \{0, 1, ..., n-1\}\} \cup \{\langle A\rangle, \langle B\rangle\}$ be a set of letters called *dummy symbols*.

Without loss of generality we may assume that the Turing machine M has exactly one *accepting state* q_n, and that M halts if and only if it enters this state. So p_n and s_n can be regarded as the representatives of this unique accepting state. Let $\mathrm{CONFIG} := \{\$\} \cdot (\Sigma_b \cup D)^* \cdot Q_1 \cdot (\Sigma_b \cup D)^* \cdot \{¢\}$, and $\mathrm{HALTING} := \{\$\} \cdot (\Sigma_b \cup D)^* \cdot \{p_n, s_n\} \cdot (\Sigma_b \cup D\})^* \cdot \{¢\}$. Then the elements of CONFIG can be considered as descriptions of possible configurations of the Turing machine M interspersed with occurrences of dummy symbols, while the elements of HALTING

correspond to possible halting configurations of M. Finally, let $\Gamma := \pi \cup Q_1 \cup D \cup \{\langle C\rangle\}$. Then $R(M)$ is the finite string-rewriting system containing the following two groups of rules:

(1) *Rules to simulate M.*

$$\left.\begin{array}{l} \langle ap_i\rangle\, cs_j \to ap_i \\ \langle s_i a\rangle\, cs_j \to s_i a \end{array}\right\} \quad \text{if} \quad \delta(q_i, a) = (q_j, c, R),$$

$$\left.\begin{array}{l} p_j c\langle ap_i\rangle \to ap_i \\ p_j c\langle s_i a\rangle \to s_i a \end{array}\right\} \quad \text{if} \quad \delta(q_i, a) = (q_j, c, L),$$

$$\left.\begin{array}{l} \$\langle \$p_i\rangle\, cs_j \to \$p_i \\ \langle s_i ¢\rangle\, cs_j ¢ \to s_i ¢ \end{array}\right\} \quad \text{if} \quad \delta(q_i, b) = (q_j, c, R),$$

$$\left.\begin{array}{l} \$p_j c\,\langle \$p_i\rangle \to \$p_i \\ p_j c\,\langle s_i ¢\rangle\, ¢ \to s_i ¢ \end{array}\right\} \quad \text{if} \quad \delta(q_i, b) = (q_j, c, L),$$

(2) *Rules to deal with dummy symbols.*

$$p_i\langle d\rangle\langle A\rangle \to \langle d\rangle\, p_i,$$
$$\langle B\rangle\langle d\rangle\, s_i \to s_i\langle d\rangle,$$
$$\langle B\rangle\, p_i s_j \to \langle C\rangle,$$
$$p_i s_j\langle A\rangle \to \langle C\rangle,$$
$$\langle C\rangle\langle A\rangle \to \langle C\rangle,$$
$$\langle B\rangle\langle C\rangle \to \langle C\rangle$$

for all $a, c \in \Sigma_b$, $q_i \in Q$, $p_i \in Q_p$, $s_j \in Q_S$, and $\langle d\rangle \in D$. Since $q_n \in Q$ is the accepting state, $\delta(q_n, a)$ is undefined for all $a \in \Sigma_b$ implying that neither p_n nor s_n occurs on the right-hand side of any of the rules of group (1).

String-rewriting system $R(M)$ satisfies the following properties [27, 29]:

(1) $R(M)$ is finite, length-reducing, and confluent,

(2) $\mathrm{dom}(R(M)) \subseteq \Gamma^2 \cup \Gamma^3 \cup \Gamma^4$ and $\mathrm{range}(R) \subseteq \Gamma \cup \Gamma^2$,

(3) $\{\$s_0\} \cdot \Sigma^* \cdot \{¢\} \subseteq \mathrm{IRR}(R(M))$,

(4) if $w \in \mathrm{CONFIG}$ and $w \to^*_{R(M)} z$, then $z \in \mathrm{CONFIG}$,

(5) if $z \in \mathrm{CONFIG}$ and $w \to^*_{R(M)} z$, then $w \in \mathrm{CONFIG}$,

(6) M halts on input $x \in \Sigma^*$ if and only if $\exists w \in \mathrm{HALTING}$: $w \to^*_{R(M)} \$s_0 x¢$.

Thus, from the choice of M we obtain the following equivalence for all $x \in \Sigma^*$:

$$x \in L \text{ if and only if } \$s_0 x¢ \in \Delta^*_{R(M)}(\mathrm{HALTING})$$
$$\text{if and only if } \langle \$s_0 x¢\rangle_{R(M)} \cap \mathrm{HALTING} \neq \emptyset.$$

Further, we see that $[\$s_0 x¢]_{R(M)} \subseteq \mathrm{CONFIG}$ for each $x \in \Sigma^*$.

We now modify the above construction in order to prove that the decision problem CCC is undecidable in general. Let § be another new letter, let $\Delta := \Gamma \cup \{\S\}$, and let $R_L := R(M) \cup \{(p_n a, \S), (s_n a, \S) \mid a \in \Sigma_b \cup D \cup \{¢\}\}$. Then R_L is a finite length-reducing string-rewriting system on Δ. Obviously, R_L is not confluent, since none of the critical pairs that result from overlaps between left-hand sides of rules of $R(M)$ and left-hand sides of rules from $\{(p_n a, \S), (s_n a, \S) \mid a \in \Sigma_b \cup D \cup \{¢\}\}$ can be resolved.

Now let $x \in \Sigma^*$. If $x \in L$, then $\langle \$s_0 x¢ \rangle_{R_L} \cap \text{HALTING} = \langle \$s_0 x¢ \rangle_{R(M)} \cap \text{HALTING} \neq \emptyset$, i.e., there exist words $z_1, z_2 \in (\Sigma_b \cup D)^*$ and $a \in \{p_n, s_n\}$ such that $\$z_1 a z_2 ¢ \to^*_{R(M)} \$s_0 x¢$. On the other hand, $\$z_1 a z_2 ¢ \to_{R_L} \$z_1 \S z_3$ for some $z_3 \in (\Sigma_b \cup D \cup \{¢\})^*$, and since $\$s_0 x¢$ and $\$z_1 \S z_3$ are irreducible, this means that R_L is not confluent on $[\$s_0 x¢]_{R_L}$. If $x \notin L$, then $\langle \$s_0 x¢ \rangle_{R_L} \cap \text{HALTING} = \langle \$s_0 x¢ \rangle_{R(M)} \cap \text{HALTING} = \emptyset$, and hence for all $(u, v) \in \text{UCP}(R_L)$, $L_u(\$s_0 x¢) = \emptyset = L_v(\$s_0 x¢)$. Hence, by Theorem 2.1, R_L is confluent on $[\$s_0 x¢]_{R_L}$. This completes the proof of Theorem 3.1. ∎

What happens, if, instead of the string-rewriting system R, the congruence class considered is fixed? To conclude this section, we look at the following restricted version of the decision problem CCC:

Confluence on Class of the Empty Word (CCEW)

INSTANCE. A finite string-rewriting system R on Σ.

Question. Is R confluent on $[e]_R$?

Let R_L be the finite length-reducing string-rewriting system on alphabet $\Delta \supseteq \Sigma \cup \{\$, s_0, ¢\}$ constructed in the proof of Theorem 3.1. For $x \in \Sigma^*$, let $R_L(x)$ denote the string-rewriting system $R_L(x) := R_L \cup \{(\$s_0 x¢, e)\}$. Since the rule $(\$s_0 x¢, e)$ does not introduce any new critical pairs, we can conclude the following from the properties of R_L stated before:

$R_L(x)$ is confluent on $[e]_{R_L(x)}$

if and only if R_L is confluent on $[\$s_0 x¢]_{R_L}$

if and only if $x \notin L$.

Thus, it is undecidable whether or not $R_L(x)$ is confluent on $[e]_{R_L(x)}$. Since $R_L(x)$ can easily be obtained from x, we have the following undecidability result.

THEOREM 3.2. *The decision problem CCEW is undecidable in general, even when it is restricted to length-reducing string-rewriting systems.*

4. FINITE MONADIC STRING-REWRITING SYSTEMS

A string-rewriting system R on Σ is called *monadic* if it is length-reducing and $\text{range}(R) \subseteq \Sigma \cup \{e\}$. During this section we shall only be dealing with finite monadic string-rewriting systems.

The equivalence problem for languages of the form $L_u(w)$ (cf. Theorem 2.1) is undecidable in general, even for finite length-reducing systems (Theorem 3.1). However, if R is finite and monadic, then each language of the form $L_u(w)$ is a deterministic 1-turn context-free language, as we shall see in the following. In fact, given a finite monadic string-rewriting system R on Σ and two words $u, w \in \Sigma^*$, a deterministic 1-turn pushdown automaton (d1pda) $A(u, w)$ recognizing the language $L_u(w)$ can be constructed effectively. This implies that condition (ii) of Theorem 2.1 is effectively decidable in this situation due to Valiant's result on the equivalence problem for deterministic finite-turn pushdown automata [32].

For constructing the d1pda $A(u, w)$ we need the following observations regarding reduction sequences generated by finite monadic string-rewriting systems. Since they are easily verified, no proofs are given.

LEMMA 4.1. *Let R be a finite monadic string-rewriting system on Σ, let $x, y \in \mathrm{IRR}(R)$, and let $k \in \mathbf{N}$, $w_0, w_1, \ldots, w_k \in \Sigma^*$ such that $xy = w_0 \to_R w_1 \to_R \cdots \to_R w_k$. Then for each $i \in \{0, 1, \ldots, k-1\}$ there exist words $z_i, z_i' \in \Sigma^*$ and a rule $(l_i, r_i) \in R$ such that the following conditions are satisfied:*

(i) $w_i = z_i l_i z_i'$,

(ii) $w_{i+1} = z_i r_i z_i'$,

(iii) *z_0 is a proper prefix of x, and for $i > 0$, z_i is a prefix of z_{i-1}, and*

(iv) *z_0' is a proper suffix of y, and for $i > 0$, z_i' is a suffix of z_{i-1}'.*

LEMMA 4.2. *Let R be a finite monadic string-rewriting system on Σ^*, let $x \in \mathrm{IRR}(R)$, and let $y \in \Sigma^*$. If $xy \to_R^* w = zlz'$ for some words $z, z' \in \Sigma^*$ and $l \in \mathrm{dom}(R)$, then $|l| + |z'| \leqslant |y| + \mu - 1$, where $\mu := \max\{|l| \mid l \in \mathrm{dom}(R)\}$.*

Thus, if $xy = w_0 \to_R w_1 \to_R \cdots \to_R w_k$ for some words $x \in \mathrm{IRR}(R)$ and $y \in \Sigma^*$, then at each step of this reduction sequence a rule of R is applied to the suffix of length $|y| + \mu - 1$ of the word under consideration. In particular, w_k is irreducible if and only if its suffix of length $|y| + \mu - 1$ is irreducible.

Now we are prepared to prove the main technical lemma of this section.

LEMMA 4.3. *There is an algorithm for solving the following task:*

Input. *A finite monadic string-rewriting system R on Σ, and two words $u, w \in \Sigma^*$.*

Output. *A deterministic 1-turn pushdown automaton $A(u, w)$ that recognizes the language $L_u(w)$.*

Proof. According to the remarks proceeding Theorem 2.1 we can effectively determine a subsystem R_1 of R satisfying the following two conditions:

(i) $\mathrm{dom}(R_1) = \mathrm{dom}(R)$,

(ii) no two distinct rules of R_1 have the same left-hand side.

Then $\mathrm{IRR}(R) = \mathrm{IRR}(R_1)$, and the system R_1 induces the unambiguous process of leftmost reduction $\to_{R,L}$.

Also from R we can construct a deterministic finite state acceptor (dfsa) B accepting the language $\mathrm{IRR}(R)$ [4]. Finally, without loss of generality we may assume that w is irreducible, which means that $L_u(w) = \{x \# y \mid x, y \in \mathrm{IRR}(R), xuy \to_{R,L}^* w\}$.

We now give an informal description of the dlpda $A(u, w)$. Let $\mu = \max\{|l| \mid l \in \mathrm{dom}(R) \cup \{w\}\}$, and let $\lambda = \mu + |u|$. As *input alphabet* and as *stack alphabet* we choose $\Sigma_0 := \Sigma \cup \{\#\}$, where $\#$ also serves as the *start symbol* marking the bottom of the pushdown store. $A(u, w)$ can store two words $x, y \in \Sigma^*$ of length $|x| \leqslant \lambda$ and $|y| \leqslant \mu$ in its finite control, so that each state q' of $A(u, w)$ can be thought of as a triple (q, x, y), where q is from a finite set Q of *proper states* and $x, y \in \Sigma^*$ satisfy $|x| \leqslant \lambda$, $|y| \leqslant \mu$. In the initial state q_0' the word u is stored in the left-hand part of the finite control, i.e., $q_0' = (q_0, u, e)$.

At each time the situation of $A(u, w)$ can be described by a 4-tuple (w_1, w_2, w_3, w_4), where

— $w_1 \in \Sigma^*$ such that $\# w_1$ is the contents of the pushdown store,

— $w_2 \in \Sigma^*$, $|w_2| \leqslant \lambda$, is the word stored in the left-hand part of the finite control,

— $w_3 \in \Sigma^*$, $|w_3| \leqslant \mu$, is the word stored in the right-hand part of the finite control,

— $w_4 \in \Sigma_0^*$ is the remaining input, the initial letter of w_4 being the actual input symbol.

$A(u, w)$ has three stages: READ, REDUCE LEFT, REDUCE RIGHT. These stages are executed one after another in the given order.

READ: $A(u, w)$ starts with the situation $(e, u, e, x \# y)$. Now the input is read letter by letter and copied onto the pushdown store until the first occurrence of the letter $\#$ is found, i.e., until the situation $(x, u, e, \# y)$ is reached. While doing this copying $A(u, w)$ simulates the dfsa B. If B is not in an accepting state when the first occurrence of the letter $\#$ is encountered, i.e., the word x copied onto the pushdown store is not irreducible, then $A(u, w)$ enters a distinguished *failure state* q_f, otherwise $A(u, w)$ proceeds with stage REDUCE LEFT.

REDUCE LEFT: $A(u, w)$ reduces the word xu to some irreducible word x_1 by computing a leftmost reduction $xu = w_0 \to_{R,L} w_1 \to_{R,L} \cdots \to_{R,L} w_k = x_1$ using the rules of the subsystem R_1. At step i of this reduction sequence $(i = 0, 1, ..., k-1)$ the suffix of length λ of the word w_i is stored in the left-hand part of the finite control, if $|w_i| \geqslant \lambda$, otherwise, all of w_i is stored there. By Lemma 4.2 this means that each reduction step is performed on the word stored in the left-hand part of the finite control of $A(u, w)$. Thus, an upper part of the contents of the pushdown store is read while this reduction sequence is being computed, but no letter is written onto the pushdown store. In addition, the input is not being used at all during this stage,

i.e., all the moves described above are ε-moves. On reaching the irreducible word x_1 $A(u, w)$ enters stage REDUCE RIGHT.

REDUCE RIGHT: When entering stage REDUCE RIGHT $A(u, w)$ is in the following situation: $(w_1, w_2, e, \#y)$, where $w_1 w_2 = x_1 \in \mathrm{IRR}(R)$, $|w_2| < \lambda$ implying $w_1 = e$.

Now the letter $\#$ that has already been encountered at the end of stage READ is deleted from the input tape. Then $A(u, w)$ reduces the word $x_1 y$ to some irreducible word x_2 by computing a leftmost reduction $x_1 y = v_0 \to_{R,L} v_1 \to_{R,L} \cdots \to_{R,L} v_l = x_2$ provided y does not contain any occurrences of the letter $\#$, i.e., $y \in \Sigma^*$. If an occurrence of the letter $\#$ is encountered while y is being read, then $A(u, w)$ immediately enters its failure state q_f.

The above mentioned reduction sequence is computed as follows. $A(u, w)$ reads the input y letter by letter. While reading each letter $A(u, w)$ performs the following two actions in parallel:

(1) The dfsa B is being simulated on the input y, i.e., the actual state of B is part of the actual state of $A(u, w)$.

(2) Assume that (w_1, w_2, y_1, ay_2) is the actual situation, where $|w_2| \leqslant \lambda$, $|w_2| < \lambda$ implying $w_1 = e$, $|y_1| \leqslant \mu$, $w_1 w_2 y_1 \in \mathrm{IRR}(R)$, and $a \in \Sigma$.

Case 2.1. $|y_1| < \mu$, and $w_2 y_1 a \in \mathrm{IRR}(R)$. Then the letter a is appended to the word y_1, i.e., we obtain the situation $(w_1, w_2, y_1 a, y_2)$.

Case 2.2. $|y_1| = \mu$, and $w_2 y_1 a \in \mathrm{IRR}(R)$. Then $A(u, w)$ enters its failure state q_f.

Case 2.3. $w_2 y_1 a$ is reducible. Then $w_2 y_1 a = zl$ for some word $z \in \Sigma^*$, $|z| \leqslant |w_2|$, and some rule $(l, r) \in R_1$ such that $w_2 y_1 a = zl \to_{R,L} zr$. $A(u, w)$ performs this reduction step, which yields the situation (w_1, z, r, y_2). Now the left-hand part of the finite control is refilled by reading letters from the pushdown store, i.e., we obtain the situation $(w_1', w_2' z, r, y_2)$, where $w_1 = w_1' w_2'$, $|w_2' z| \leqslant \lambda$, $|w_2' z| < \lambda$ implying $w_1' = e$.

If $w_2' zr$ is reducible, then another leftmost reduction step is applied. This process is repeated until a situation (w_1'', w_2'', r', y_2) is reached such that $|w_2''| \leqslant \lambda$, $|w_2''| < \lambda$ implying $w_1'' = e$, $r' \in \mathrm{range}(R_1)$, and $w_2'' r' \in \mathrm{IRR}(R)$.

$A(u, w)$ accepts if and only if after reading all of the input $A(u, w)$ is in a situation of the form (e, w_1, w_2, e), where $w_1 w_2 = w$, and the actual state of B that is part of the actual state of $A(u, w)$ is accepting.

Obviously, $A(u, w)$ is a deterministic pushdown automaton a formal definition of which is effectively computable from R_1 and the words u and w by using the informal description given above. While $A(u, w)$ is executing stage READ, the length of the contents of the pushdown store is strictly increasing; later on it is nonincreasing. Thus, $A(u, w)$ is a d1pda. It remains to prove the following claim.

CLAIM. *$A(u, w)$ accepts on input x_0 if and only if $x_0 \in L_u(w)$.*

Proof. "$\Leftarrow$" Let $x_0 \in L_u(w)$. Then there are words $x, y \in \mathrm{IRR}(R)$ such that

$x_0 = x\#y$ and $xuy \to^*_{R,L} w$. This leftmost reduction sequence can be written as $xuy = w_0 y \to_{R,L} w_1 y \to_{R,L} \cdots \to_{R,L} w_k y = v_0 \to_{R,L} v_1 \to_{R,L} \cdots \to_{R,L} v_l = w$, since it proceeds from left to right. Thus, on input x_0 $A(u, w)$ reaches stage REDUCE RIGHT while being in the situation

$$(w_1', w_2', e, \#y),\ w_1' w_2' = w_k,\ |w_2'| \leqslant \lambda,\ |w_2'| < \lambda \qquad \text{implying } w_1' = e.$$

Since $w_k \in \mathrm{IRR}(R)$ and since $y \in \mathrm{IRR}(R)$, Lemma 4.1 implies that the remaining reduction steps are always applied at the border between w_k and y. Since $\mu \geqslant \max\{|l| \mid l \in \mathrm{dom}(R)\}$, this means that Case 2.2 cannot occur during this computation, i.e., $A(u, w)$ computes the whole leftmost reduction sequence given above. Thus, $A(u, w)$ accepts on input x_0.

"$\Rightarrow$" Let $x_0 \in \Sigma_0^*$ such that $A(u, w)$ accepts on input x_0. Then $x_0 = x\#y$ for some words $x, y \in \mathrm{IRR}(R)$. Thus, $A(u, w)$ reaches stage REDUCE RIGHT while being in the situation

$$(w_1, w_2, e, \#y),\ w_1 w_2 = x_1 \in \mathrm{IRR}(R),\ |w_2| \leqslant \lambda,\ |w_2| < \lambda \qquad \text{implying} \quad w_1 = e$$
$$\text{and} \qquad xu \to^*_{R,L} x_1.$$

Since $A(u, w)$ accepts eventually, Case 2.2 does not occur while $A(u, w)$ is executing stage REDUCE RIGHT. Thus, $A(u, w)$ computes a leftmost reduction sequence $x_1 y \to^*_{R,L} w$ implying that $xuy \to^*_{R,L} x_1 y \to^*_{R,L} w$. Hence, $x_0 = x\#y \in L_u(w)$. ▮

This completes the proof of Lemma 4.3. ▮

Let R be a finite monadic string-rewriting system on Σ, and let $w \in \Sigma^*$. From R we can effectively compute the finite set $\mathrm{UCP}(R)$. Now for each pair $(u, v) \in \mathrm{UCP}(R)$, dlpdas $A(u, w)$ and $A(v, w)$ can be determined effectively such that $A(u, w)$ recognizes the language $L_u(w)$, while $A(v, w)$ recognizes the language $L_v(w)$ (Lemma 4.3). By Valiant's result [32] it is decidable whether or not $A(u, w)$ and $A(v, w)$ are equivalent, i.e., whether or not $L_u(w) = L_v(w)$. Hence, Theorem 2.1 gives the following result.

THEOREM 4.4. *The following problem is effectively decidable*:

INSTANCE. A finite monadic string-rewriting system R on Σ, and a word $w \in \Sigma^*$.

Question. Is R confluent on $[w]_R$?

From R the set $\mathrm{UCP}(R)$ can be constructed in polynomial time (cf., e.g., [9]). Also the subsystem R_1 can be obtained from R in polynomial time. Given w we can determine the word $w_1 \in \mathrm{IRR}(R)$ such that $w \to^*_{R,L} w_1$ in polynomial time as well [5]. However, for constructing the dlpda $A(u, w)$ from u, w_1, and R_1, we will in general need exponential time, since $A(u, w)$ may be of exponential size.

In general, the equivalence of two deterministic finite-turn pushdown automata can be tested in double exponential time [3]. However, due to their specific form

the equivalence of $A(u, w)$ and $A(v, w)$ is decidable in time bounded above by an exponential function in the size of $A(u, w)$ and $A(v, w)$ (cf. the proof of Theorem 5.2 of [3]). Thus, our solution to the decision problem stated in Theorem 4.4 is double exponentially time bounded. Since we do not yet see a way of improving this time bound in general, we shall consider restricted instances of this decision problem in the next sections.

5. Finite Special String-Rewriting Systems

A finite string-rewriting system R on Σ is called *special*, if it is length-reducing and $\text{range}(R) = \{e\}$. Since the special string-rewriting systems form a restricted class of monadic string-rewriting systems, problem CCC is decidable in double exponential time for this class of finite string-rewriting systems. Here we want to show that at least the problem CCEW becomes decidable in exponential time under this restriction.

Recall that for a string-rewriting system R on Σ, $\text{UCP}(R)$ denotes the set of all critical pairs of R that cannot be resolved. Further, let μ_R denote the integer $\mu_R := \max\{|l| \mid l \in \text{dom}(R)\}$. In what follows the condition (C) for string-rewriting systems will play an important role:

(C) $\forall (u, v) \in \text{UCP}(R)\, \forall p, q \in \Sigma^*: |p|,\ |q| \leqslant \mu_R^2$ implies that $puq \to_R^* e$ if and only if $pvq \to_R^* e$.

Obviously, condition (C) is a weakened version of condition (ii) of Theorem 2.1 applied to $w = e$. Thus, if a finite special string-rewriting system R on Σ is confluent on $[e]_R$, then R satisfies condition (C). But also the converse implication holds as shown by the following lemma.

Lemma 5.1. *Let R be a finite special string-rewriting system on Σ. If R satisfies condition* (C), *then R is confluent on $[e]_R$.*

Proof. Assume that R satisfies condition (C), but R is not confluent to $[e]_R$. Then there exists a word $z \in \Sigma^+$ of shortest length such that $\Delta_R^*(z) \cap \text{IRR}(R) \supsetneq \{e\}$ (cf. proof of Theorem 2.1). Also from the proof of Theorem 2.1 we conclude that $z = pwq$ for some words $p, q \in \text{IRR}(R)$ and some word $w \in \Sigma^*$ such that $w \to_R u$ and $w \to_R v$ for a critical pair $(u, v) \in \text{UCP}(R)$. Thus, we may assume the following situation:

$$\begin{array}{ccc} z = pwq & \to_R & puq \to_R^* e \\ {\scriptstyle R}\downarrow & & \\ pvq & \not\to_R^* & e. \end{array}$$

Since $puq \to_R^* e$, we can factor p, u, and q as $p = p_1 p_2 x$, $u = u_1 u_2 u_3$, and $q = y q_2 q_3$ such that

— $xu_1 \to_R^* e$ implying $|x| \leqslant |u_1| \cdot (\mu_R - 1) \leqslant \mu_R \cdot (\mu_R - 1)$,

— $u_3\, y \to_R^* e$ implying $|y| \leqslant |u_3| \cdot (\mu_R - 1) \leqslant \mu_R \cdot (\mu_R - 1)$,

— $p_2 u_2 q_2 = e$ or $(p_2 u_2 q_2, e) \in R$ implying $|p_2 q_2| < \mu_R$, and

— $p_1 q_3 \to_R^* e$.

Assume that $p_1 q_3 \neq e$, and let $z' := p_2 xwyq_2$. Then $|z'| < |z|$ and

$$z' = p_2 xwyq_2 \to_R p_2 xuyq_2 = p_2\, xu_1\, u_2 u_3\, yq_2 \to_R^* e$$

$$\Big|_R$$

$$p_2\, xvyq_2.$$

Hence, by the choice of z, $p_2\, xvyq_2 \to_R^* e$ which implies that $pvq = p_1 p_2\, xvyq_2 q_3 \to_R^* p_1 q_3 \to_R^* e$, thus contradicting the choice of (u, v). Hence, $p_1 q_3 = e$, and therefore $|p| = |p_2 x| = |p_2| + |x| \leqslant \mu_R + \mu_R \cdot (\mu_R - 1) = \mu_R^2$ and $|q| = |yq_2| = |y| + |q_2| \leqslant \mu_R \cdot (\mu_R - 1) + \mu_R = \mu_R^2$. Now $(u, v) \in \mathrm{UCP}(R)$ and $puq \to_R^* e$ yield $pvq \to_R^* e$ by condition (C) again giving the same contradiction. Thus, R is confluent on $[e]_R$. ∎

So we see that a finite special string-rewriting system R is confluent on $[e]_R$ if and only if it satisfies condition (C). Given R, the set $\mathrm{UCP}(R)$ and the constant μ_R can be computed in polynomial time. Also for each $(u, v) \in \mathrm{UCP}(R)$ and $p, q \in \Sigma^*$, $|p|, |q| \leqslant \mu_R^2$, it can be determined in polynomial time whether or not $puq \to_R^* e$ or $pvq \to_R^* e$, respectively, holds. Since exponentially many of these tests must be performed, we obtain the following result.

THEOREM 5.2. *The following problem is decidable in exponential time:*

INSTANCE. A finite special string-rewriting system R on Σ.

Question. Is R confluent on $[e]_R$?

For a special string-rewriting system R, the congruence class $[e]_R$ is obviously distinguished among all the congruence classes of R. Thus, it is not surprising that properties of $[e]_R$ play a role when the confluence of R on $[w]_R$ is considered for some nonempty word $w \in \mathrm{IRR}(R)$.

For $w \in \mathrm{IRR}(R) - \{e\}$, we formulate the following two conditions that a finite special string-rewriting system R on Σ may or may not satisfy:

(C1(w)) $\forall (u, v) \in \mathrm{UCP}(R)\ \forall p, q \in \Sigma^*$: $|p|, |q| \leqslant |w| + \mu_R^2$ implies that $puq \to_R^* w$ if and only if $pvq \to_R^* w$.

(C2(w)) $\forall x \in \mathrm{IRR}(R)\ \forall w_1, w_2 \in \Sigma^*$: $x \leftrightarrow_R^* e$ and $w = w_1 w_2$ imply that $w_1 x w_2 \to_R^* w$.

Condition (C1(w)) is the adopted version of condition (C), while (C2(w)) puts a condition on all those irreducible words that are in $[e]_R$. As can be seen easily con-

ditions (C1(w)) and (C2(w)) are necessary for a finite special string-rewriting system R to be confluent on $[w]_R$, where $w \in \mathrm{IRR}(R)$. That these conditions are also sufficient is shown by the following lemma.

LEMMA 5.3. *Let R be a finite special string-rewriting system on Σ, and let $w \in \Sigma^+$ be irreducible. If R satisfies conditions* (C1(w)) *and* (C2(w)), *then R is confluent on* $[w]_R$.

Proof. Let R satisfy conditions (C1(w)) and (C2(w)), but assume that R is not confluent on $[w]_R$. Then there exists a word $z \in \Sigma^+$ of shortest length such that $\Delta_R^*(z) \cap \mathrm{IRR}(R) \supsetneq \{w\}$. Further, z can be factored as $z = psq$, where $p, q \in \mathrm{IRR}(R)$, and $s \in \Sigma^*$ is such that $s \to_R u$ and $s \to_R v$ for a critical pair $(u, v) \in \mathrm{UCP}(R)$. Thus, we may assume the following situation:

$$\begin{array}{ccc} z = psq & \to_R & puq \to_R^* w \\ {\scriptstyle R}\downarrow & & \\ pvq & \not\to_R^* & w. \end{array}$$

Because of (C1(w)) we can conclude that $|p| > |w| + \mu_R^2$ or $|q| > |w| + \mu_R^2$.

On the other hand, since R is a special string-rewriting system, and (u, v) is a critical pair of R, we have $|u| \leqslant \mu_R - 1$. Hence, during the reduction sequence $puq \to_R^* w$, the word u is totally cancelled, which means that we can factor p, u, and q as $p = w_1 p_1 p_2 x$, $u = u_1 u_2 u_3$, and $q = y q_2 q_3 w_2$ such that

— $x u_1 \to_R^* e$ implying $|x| \leqslant |u_1| \cdot (\mu_R - 1)$,

— $u_3\, y \to_R^* e$ implying $|y| \leqslant |u_3| \cdot (\mu_R - 1)$,

— $p_2 u_2 q_2 = e$ or $(p_2 u_2 q_2, e) \in R$ implying $|p_2 q_2| \leqslant \mu_R - 1$,

— $p_1 q_3 \to_R^* e$, and

— $w_1 w_2 = w$.

Now, $p_1 p_2 xvyq_2 q_3 \leftrightarrow_R^* p_1 p_2 xuyq_2 q_3 = p_1 p_2 x u_1 u_2 u_3\, yq_2 q_3 \to_R^* e$ while $p_1 p_2 xvyq_2 q_3 \to_R^* t$ for some word $t \in \mathrm{IRR}(R)$. Hence, $t \in \mathrm{IRR}(R) \cap [e]_R$, and therefore, $w_1 t w_2 \to_R^* w$ by (C2(w)). Thus, $pvq = w_1 p_1 p_2 xvyq_2 q_3 w_2 \to_R^* w_1 t w_2 \to_R^* w$ contradicting the choice of (u, v). Thus, R is confluent on $[w]_R$. ∎

So a finite special string-rewriting system R is confluent on $[w]_R$ for some irreducible word w if and only if R satisfies conditions (C1(w)) and (C2(w)). Condition (C2(w)) now induces the following necessary condition for R to be confluent on $[w]_R$.

LEMMA 5.4. *Let R be a finite special string-rewriting system on Σ, and let $w \in \mathrm{IRR}(R)$. If R is confluent on $[w]_R$, then $[e]_R$ contains no irreducible word z of length $|z| > \mu_R \cdot |w|$.*

Proof. Let R be confluent on $[w]_R$, and let $z \in \mathrm{IRR}(R) \cap [e]_R$. By condition (C2($w$)) this means that $zw \to_R^* w$. Since R is special, and since w and z are irreducible, we can factor w and z as $w = w_1 w_2$, $z = z_1 z_2$ such that

— $z_2 w_1 \to_R^* e$ implying that $|z_2| \leqslant |w_1| \cdot (\mu_R - 1)$, and

— $z_1 w_2 = w = w_1 w_2$ implying that $z_1 = w_1$.

Hence, $|z| = |z_1| + |z_2| \leqslant |w_1| + |w_1| \cdot (\mu_R - 1) = |w_1| \cdot \mu_R \leqslant |w| \cdot \mu_R$. ∎

From Lemma 5.4 we can immediately conclude the following.

COROLLARY 5.5. *Let R be a finite special string-rewriting system on Σ. If the intersection $\mathrm{IRR}(R) \cap [e]_R$ is infinite, i.e., if there are infinitely many irreducible words that are congruent to the empty word modulo R, then R is not confluent on any congruence class $[w]_R$.*

As condition (C) also condition (C1(w)) can be decided in exponential time for each finite special string-rewriting system R on Σ and each irreducible word $w \in \Sigma^+$. If $\mathrm{IRR}(R) \cap [e]_R$ is finite, then condition (C2(w)) can be checked in time that is polynomial in the size of R and w, and that is linear in the size of the intersection $\mathrm{IRR}(R) \cap [e]_R$. Thus, we obtain the following result.

THEOREM 5.6. *For each finite special string-rewriting system R on Σ, there exists an algorithm that solves the following decision problem in exponential time*:

INSTANCE. A word $w \in \Sigma^*$.

Question. Is R confluent on $[w]_R$?

Proof. If the intersection $\mathrm{IRR}(R) \cap [e]_R$ is infinite, then R is not confluent on any congruence class $[w]_R$ by Corollary 5.5. Thus, the corresponding algorithm always gives the answer "no." So let $\mathrm{IRR}(R) \cap [e]_R$ be finite, i.e., $\mathrm{IRR}(R) \cap [e]_R = \{x_1, x_2, ..., x_n\}$ for some integer $n \in \mathbf{N}$. In linear time an irreducible descendant w_1 of w can be computed [5]. Now R is confluent on $[w]_R$ if and only if R satisfies conditions (C1(w_1)) and (C2(w_1)), which can be determined in exponential time according to the remarks stated above. ∎

Obviously, condition (C2(w)) is redundant whenever R is known to be confluent on $[e]_R$. So under this assumption we get the following result corresponding to Theorem 5.2.

COROLLARY 5.7. *The following problem is decidable in exponential time*:

INSTANCE. A finite special string-rewriting system R on Σ that is confluent on $[e]_R$, and a word $w \in \Sigma^*$.

Question. Is R confluent on $[w]_R$?

Observe that a finite special string-rewriting system R on Σ that is confluent on $[e]_R$ may be nonconfluent on some other congruence class $[w]_R$ as shown by the following example.

EXAMPLE 5.8. Let $\Sigma = \{a, b, c\}$, and $R = \{(ab, e), (ba, e), (ac, e), (ca, e)\}$. Then R is a finite special string-rewriting system on Σ, and since $b \leftrightarrow_R bac \leftrightarrow_R c$ with $b, c \in \mathrm{IRR}(R)$, R is not confluent on $[b]_R$. On the other hand, it can be seen easily that $\langle e \rangle_R = \{z \in \Sigma^* \mid |z|_a = |z|_b + |z|_c\} = [e]_R$, i.e., for $z \in \Sigma^*$, $z \to_R^* e$ if and only if $z \leftrightarrow_R^* e$. Thus, R is confluent on $[e]_R$.

We conclude this section with an observation concerning the special word problem for a finite special string-rewriting system R on Σ. Here the *special word problem for* R is the following decision problem:

INSTANCE. A word $z \in \Sigma^*$.

Question. Does $z \leftrightarrow_R^* e$ hold?

If a finite special string-rewriting system R is confluent on $[e]_R$, then the special word problem for R can be solved in linear time. Now Lemma 5.4 gives the following generalization of this result.

COROLLARY 5.9. *Let R be a finite special string-rewriting system on Σ. If R is confluent on $[w]_R$ for some word $w \in \Sigma^*$, then the special word problem for R is decidable in linear time.*

Proof. If R is confluent on $[w]_R$ for some word $w \in \Sigma^*$, then the intersection $\mathrm{IRR}(R) \cap [e]_R$ is finite, i.e., $\mathrm{IRR}(R) \cap [e]_R = \{x_1, x_2, ..., x_n\}$. On input $z \in \Sigma^*$, an irreducible descendant z_1 of z modulo R can be determined in linear time. Now $z \leftrightarrow_R^* e$ if and only if $z_1 = x_i$ for some $i \in \{1, 2, ..., n\}$. Since the set $\{x_1, x_2, ..., x_n\}$ depends on R only but not on z, this test can be performed in linear time. ∎

6. FINITE HOMOGENEOUS STRING-REWRITING SYSTEMS

In this section we are mainly concerned with finite string-rewriting systems that are homogeneous of degree 2. Here, a finite string-rewriting system R on Σ is called *homogeneous of degree* k if it is special and $(l, e) \in R$ implies $|l| = k$. It is *homogeneous* if it is homogeneous of degree k for some k.

We now give three conditions that a finite string-rewriting system R on Σ that is homogeneous of degree 2 may or may not satisfy:

(C1) $\forall a, b, c, d \in \Sigma$: $[(ab, e), (bc, e), (cd, e) \in R \Rightarrow (ad, e) \in R]$.

(C2) $\forall a, b, c, d \in \Sigma$: $[(ab, e), (cd, e), (db, e) \in R \Rightarrow (ac, e) \in R]$.

(C3) $\forall a, b, c, d \in \Sigma$: $[(ab, e), (cd, e), (ac, e) \in R \Rightarrow (db, e) \in R]$.

LEMMA 6.1. *Let R be a finite string-rewriting system on Σ that is homogeneous of degree 2. R is confluent on $[e]_R$ if and only if it satisfies conditions* (C1) *to* (C3).

Proof. It is straightforward to verify that R satisfies conditions (C1) to (C3) if R is confluent on $[e]_R$. To prove the converse implication we assume that R satisfies conditions (C1) to (C3), and let $w \in \Sigma^*$ such that $w \leftrightarrow_R^* e$. We must show that $w \rightarrow_R^* e$. Since $w \leftrightarrow_R^* e$, there are an integer $m \geqslant 0$ and words $w_0, w_1, \ldots, w_m \in \Sigma^*$ such that $w = w_m \leftrightarrow_R w_{m-1} \leftrightarrow_R \cdots \leftrightarrow_R w_1 \leftrightarrow_R w_0 = e$. We proceed by induction on m.

If $m = 0$, then $w = e$. If $m > 0$, then $w = w_m \leftrightarrow_R w_{m-1} \leftrightarrow_R^* e$. By induction hypothesis we have $w_{m-1} \rightarrow_R^* e$. Now either $w_m \rightarrow_R w_{m-1}$ in which case we are done, or $w_{m-1} \rightarrow_R w_m$. So let $w_m = uv$ and $w_{m-1} = uabv$ for some words $u, v \in \Sigma^*$ and a rule $(ab, e) \in R$. In the reduction sequence $w_{m-1} = uabv \rightarrow_R^* e$ we can single out those steps that involve the distinguished occurrences of the letters a and b. Thus, we obtain $w_{m-1} = uabv \rightarrow_R^* u_1 abv_1 \rightarrow_R^i u_2 v_2 \rightarrow_R^* e$, where

(i) $i = 1$, $u_1 = u_2$, and $v_1 = v_2$, or

(ii) $i = 2$, $u_1 = u_2 c$, and $v_1 = dv_2$ with $(ca, e), (bd, e) \in R$, or

(iii) $i = 2$, $u_1 = u_2 cd$, and $v_1 = v_2$ with $(da, e), (cb, e) \in R$, or

(iv) $i = 2$, $u_1 = u_2$, and $v_1 = cdv_2$ with $(bc, e), (ad, e) \in R$.

Now $w = w_m = uv \rightarrow_R^* u_1 v_1$. Using conditions (C1) to (C3) it is easily verified that in each of the above cases $u_1 v_1 \rightarrow_R^* e$.

This completes the proof of Lemma 6.1. ∎

For a finite homogeneous string-rewriting system R of degree 2, conditions (C1) to (C3) can be checked in polynomial time. So we obtain the following result.

THEOREM 6.2. *The following problem is decidable in polynomial time*:

INSTANCE. A finite string-rewriting system R on Σ that is homogeneous of degree 2.

Question. Is R confluent on $[e]_R$?

In the previous section we observed that if a finite special string-rewriting system R on Σ is confluent on some congruence class $[w]_R \neq [e]_R$, then this fact induces some restrictions on the behavior of the reduction $\rightarrow_R$ on $[e]_R$ (Lemma 5.4). Here we shall see that for finite homogeneous systems of degree 2 these restrictions are even stronger.

THEOREM 6.3. *Let R be a finite homogeneous string-rewriting system of degree 2 on Σ. If R is confluent on $[w]_R$ for some $w \in \Sigma^*$, then R is also confluent on $[e]_R$.*

Proof. Let $w \in \mathrm{IRR}(R) - \{e\}$, and let R be confluent on $[w]_R$, but assume that R is not confluent on $[e]_R$. Then by Lemma 6.1 there are letters $a, b, c, d \in \Sigma$ such that

(i) $(ab, e), (bc, e), (cd, e) \in R$ but $(ad, e) \notin R$, or

(ii) $(ab, e), (cd, e), (db, e) \in R$ but $(ac, e) \notin R$, or

(iii) $(ab, e), (cd, e), (ac, e) \in R$ but $(db, e) \notin R$.

Since R is confluent on $[w]_R$, we have $w_1 u w_2 \to_R^* w$ for all $w_1, w_2 \in \Sigma^*$ such that $w = w_1 w_2$ and all $u \in [e]_R$.

(i) $ad \leftrightarrow_R abcd \leftrightarrow_R^* e$, i.e., $ad \in [e]_R$. Thus, for all $w_1, w_2 \in \Sigma^*$, if $w = w_1 w_2$, then $w_1 adw_2 \to_R w = w_1 w_2$. Since $(ad, e) \notin R$, we have $w_1 = v_1 g$ for some letter $g \in \Sigma$ with $(ga, e) \in R$ or $w_2 = hv_2$ for some letter $h \in \Sigma$ with $(dh, e) \in R$. In the first case we observe that $w_1 adw_2 = v_1 gadw_2 \to_R v_1 dw_2 = w = w_1 w_2 = v_1 gw_2$, which yields $d = g$; in the second case $w_1 adw_2 = w_1 adhv_2 \to_R w_1 av_2 = w = w_1 w_2 = w_1 hv_2$ implying $a = h$. Thus, $(da, e) \in R$, and w_1 ends in d or w_2 begins with a. Since this holds for all words $w_1, w_2 \in \Sigma^*$ satisfying $w = w_1 w_2$, we conclude that $a = d$, which in turn yields $ad = da$, i.e., $(ad, e) \in R$, a contradiction.

In cases (ii) and (iii) a contradiction is reached in an analogous manner, thus completing the proof of Theorem 6.3. ▌

Thus, if a finite homogeneous string-rewriting system R of degree 2 is confluent on any of its congruence classes at all, then it is confluent on $[e]_R$. Example 5.8 shows that the converse implication does not hold: R may be confluent on $[e]_R$, although there is some word $w \in \Sigma^*$ such that R is not confluent on $[w]_R$. Hence, Theorem 6.3 expresses a specific property of the congruence class $[e]_R$. The following example shows that this result cannot be generalized to the class of all finite string-rewriting systems that are homogeneous of degree 3.

EXAMPLE 6.4. Let $\Sigma = \{a, b\}$, and $R = \{(aab, e), (baa, e)\}$. Then R is a finite homogeneous string-rewriting system of degree 3 on Σ. Since $aba \in \mathrm{IRR}(R)$, while $aba \leftrightarrow_R baaaba \leftrightarrow_R baa \leftrightarrow_R e$, R is not confluent on $[e]_R$. However, we shall see that R is confluent on $[a]_R$. First, observe that, for all $w \in \Sigma^*$, if $w \leftrightarrow_R^* a$, then $|w|_a = 2 \cdot |w|_b + 1$. Further, for all $w \in \Sigma^*$, if $|w|_a = 2 \cdot |w|_b + 1 \geqslant 3$, then w contains a factor of the form aab or baa, i.e., w is reducible modulo R.

Now let $w \in \Sigma^*$ such that $|w|_a = 2 \cdot |w|_b + 1$, and let $u \in \mathrm{IRR}(R)$ such that $w \to_R^* u$. Then $|u|_a = 2 \cdot |u|_b + 1$ as can be seen easily from the form of the rules of R, and $|u|_a < 3$, i.e., $|u|_b = 0$ and $|u|_a = 1$. Hence, $u = a$, and so $[a]_R = \{w \in \Sigma^* \mid |w|_a = 2 \cdot |w|_b + 1\} = \langle a \rangle_R$. Thus, R is confluent on $[a]_R$.

7. Special String-Rewriting Systems with a Single Rule

A word w is *primitive* if there are no word x and integer $k > 1$ such that $w = x^k$; otherwise, w is *imprimitive*. In either case, the shortest x such that $w = x^k$ is the *root* of w, denoted $\rho(w)$. For a non-empty word w, let $\mathrm{OVL}(w) = \{y \mid$ there exist $r, s \neq e$ such that $w = ur = su\}$, and let $\mathrm{ov}(w)$ be the longest word in the set $\mathrm{OVL}(w)$. Thus,

OVL(w) consists of all proper self-overlaps of w (including the empty word), and ov(w) is the longest proper self-overlap of w, so $0 \leqslant |\mathrm{ov}(w)| < |w|$. The word w is said to have *overlap* if $\mathrm{ov}(w) \neq e$.

Let $R = \{(l, e)\}$ be a special string-rewriting system on Σ. Then according to Book [6] R is confluent if and only if $\rho(l)$ has no overlap. Using this observation we can prove the following converse of Theorem 6.3 for special one-rule systems.

LEMMA 7.1. *Let R be a special string-rewriting system with a single rule. If R is confluent on $[e]_R$, then R is a confluent string-rewriting system.*

Proof. Let $R = \{(l, e)\}$, and assume that R is not confluent. Then by Book's result mentioned above the root $\rho(l)$ of the word l has overlap, i.e., $l = ((xy)^k x)^n$ for some words x, $y \in \Sigma^*$ satisfying $xy \neq yx$ and some integers k, $n \geqslant 1$ [21].

Consider the word $w := ((xy)^k x)^{n-1}(xy)^k(xy)^k x((xy)^k x)^{n-1}x$. Then we have the following two reduction sequences starting with w:

$$w = ((xy)^k x)^{n-1}(xy)^k(xy)^k x((xy)^k x)^{n-1}x$$
$$\to_R ((xy)^k x)^{n-1}(xy)^k x \to_R e$$

and

$$w = ((xy)^k x)^{n-1}(xy)^k x(yx)^k((xy)^k x)^{n-1}x$$
$$\to_R (yx)^k((xy)^k x)^{n-1}x = ((yx)^k x)^n \neq ((xy)^k x)^n,$$

since $xy \neq yx$. Hence, $((yx)^k x)^n \in \mathrm{IRR}(R)$ implying that R is not confluent on $[e]_R$. ∎

Since the converse implication is obvious, we see that a special string-rewriting system R with a single rule is confluent if and only if it is confluent on $[e]_R$. The aim of the following investigations is to extend this result to all congruence classes $[w]_R$. To this end we develop a sequence of lemmas.

For a non-empty word $w \in \Sigma^*$, let $\pi(w) = |w| - |\mathrm{ov}(w)|$ denote the *period* of w, let $x := \mathrm{res}(w)$, the *residue* of w, be the prefix of w of length equal to the remainder when $|w|$ is divided by $\pi(w)$, and let $k = \lfloor |w|/\pi(w) \rfloor \geqslant 1$. Then there is a word $y \in \Sigma^+$ such that $z = xy$ is primitive, $\mathrm{ov}(w) = z^{k-1}x$, and $w = z^k x = (xy)^k x$ [31, Lemma 3.2].

For what follows let $R = \{(l, e)\}$ be a *fixed special one-rule string-rewriting system on Σ such that R is not confluent.* Then the root $\rho(l)$ of the word l has overlap, i.e., $\rho(l) = (xy)^k x$, where $x := \mathrm{res}(\rho(l)) \in \Sigma^+$, $y \in \Sigma^+$ such that xy is primitive with $|xy| = \pi(\rho(l))$, and $k := \lfloor |\rho(l)|/\pi(\rho(l)) \rfloor \geqslant 1$. Hence, l has the form $l = ((xy)^k x)^n$ for some integer $n \geqslant 1$. Further, let $l_1 := ((yx)^k x)^n$. Since xy is primitive, we have $xy \neq yx$, and so $l \neq l_1$. However, $l_1 \leftrightarrow_R^* e$ according to the proof of Lemma 7.1.

LEMMA 7.2. *Let $c, d \in \Sigma^+$ such that $l = cd$ and $l_1 = dc$. Then $c = ((xy)^k x)^m x$ and $d = (yx)^k((xy)^k x)^{n-1-m}$ for some $m \in \{0, 1, \ldots, n-1\}$.*

Proof. If $n=1$, we have $cd=l=(xy)^kx=x(yx)^k$ and $dc=l_1=(yx)^kx$. Since $d\neq e\neq(yx)^k$, and since l is a primitive word, we obtain $c=x$ and $d=(yx)^k$ from Lemma 3.1(a) of [31]. If $n>1$, then $cd=l=((xy)^kx)^n$ implies that $c=((xy)^kx)^mw_1$ and $d=w_2((xy)^kx)^{n-1-m}$ for some $w_1, w_2\in\Sigma^*$ satisfying $w_1w_2=(xy)^kx$ and some integer $m\in\{0, 1,..., n-1\}$. Hence, $((yx)^kx)^n=l_1=dc=w_2((xy)^kx)^{n-1}w_1=w_2(w_1w_2)^{n-1}w_1=(w_2w_1)^n$, which yields $w_2w_1=(yx)^kx$. Since $l\neq l_1$, this implies $w_1\neq e\neq w_2$. Now as in the case $n=1$ we can conclude that $w_1=x$ and $w_2=(yx)^k$, thus completing the proof of Lemma 7.2.

Since $l_1\leftrightarrow_R^* e$, $pl_1q\leftrightarrow_R^* pq$ for all words $p, q\in\Sigma^*$. The following lemma, which is easily derived from Lemma 7.2, gives a necessary condition for $pq\in\mathrm{IRR}(R)$ to satisfy $pl_1q\to_R pq$.

LEMMA 7.3. *If $p, q\in\Sigma^*$ satisfy $pq\in\mathrm{IRR}(R)$ and $pl_1q\to_R pq$, then*

(i) *$p=p_1((xy)^k x)^mx$ for some $m\in\{0, 1, ..., n-1\}$ and $p_1\in\Sigma^*$ not ending in $(xy)^kx$, or*

(ii) *$q=(yx)^k((xy)^kx)^mq_1$ for some $m\in\{0, 1, ..., n-1\}$ and $q_1\in\Sigma^*$ not beginning with $(xy)^kx$.*

Finally, we need the following lemma from combinatorics on words, which is easily verified.

LEMMA 7.4. *Let $u, v\in\Sigma^+$ such that $uv\neq vu$. Then there is a word $v_1\in\Sigma^+$ satisfying the following two conditions:*

— $v\in\{v_1\}\cdot\Sigma^*$,

— for all $m\geqslant 1$, $v_1^m\notin\Sigma^*\cdot\{u\}$.

Now we are ready to prove the main lemma of this section.

LEMMA 7.5. *Let R be as above. Then R is not confluent on any congruence class $[w]_R$.*

Proof. Since xy is primitive, $xy\neq yx$. Hence by Lemma 7.4, there exists an initial factor $y_1\in\Sigma^+$ of y such that, for all $m\geqslant 1$, $y_1^m\notin\Sigma^*\cdot\{x\}$.

Let $w\in\mathrm{IRR}(R)$. If yx is not an initial factor of w, then $l_1w\not\to_R w$ according to Lemma 7.3, and hence, R is not confluent on $[w]_R$. So assume that yx is an initial factor of w. Then w can be factored as $w=y_1^rw_1$ for some integer $r\geqslant 1$ such that y_1 is not an initial factor of w_1. If R is confluent on $[w]_R$, then $y_1^rl_1w_1\to_R y_1^rw_1=w$. Since $y_1^r\notin\Sigma^*\cdot\{x\}$, i.e., x is not a suffix of y_1^r, Lemma 7.3 implies that $w_1=(yx)^k((xy)^kx)^mw_2$ for some $m\in\{0, 1, ..., n-1\}$, which in turn yields that y_1 is an initial factor of w_1, a contradiction. Thus, R is not confluent on $[w]_R$.

Since each congruence class of R contains at least one irreducible word, we see that R is not confluent on any congruence class $[w]_R$. ∎

Thus, we have the following characterization theorem for special one-rule string-rewriting systems that are confluent, where the equivalence of (i), (ii), and (iv) is taken from Book [6].

THEOREM 7.6. *Let $R = \{(l, e)\}$ be a special one-rule string-rewriting system on Σ. Then the following statements are equivalent:*

(i) *There exists a finite length-reducing and confluent string-rewriting system R_0 on Σ such that R and R_0 are equivalent.*

(ii) *R is confluent.*

(iii) *R is confluent on $[w]_R$ for some word $w \in \Sigma^*$.*

(iv) *The root $\rho(l)$ of the word l has no overlap.*

Since given l it can be determined in linear time whether or not $\rho(l)$ has overlap [1], the decision problem CCC can be solved in linear time, when it is restricted to special string-rewriting systems having a single rule only.

We conclude with an example indicating that Theorem 7.6 does not generalize to the class of all one-rule systems.

EXAMPLE 7.7. Let $\Sigma = \{a, b\}$, and $R = \{(aba, b)\}$. Then $bba \leftarrow_R ababa \rightarrow_R abb$, and since $abb, bba \in \mathrm{IRR}(R)$, this shows that R is not confluent on $[abb]_R$.

Let $L := \{a^m b a^m \mid m \geqslant 0\}$. Then obviously, $L \subseteq \langle b \rangle_R$. On the other hand, one can check easily that $[b]_R \subseteq L$. Hence, R is confluent on $[b]_R$.

ACKNOWLEDGMENTS

The author thanks Professor R. V. Book and Professor K. Madlener for many fruitful discussions regarding the results presented in this paper.

REFERENCES

1. J. AVENHAUS AND K. MADLENER, String matching and algorithmic problems in free groups, *Rev. Colombiana Mat.* **14** (1980), 1–16.
2. G. BAUER AND F. OTTO, Finite complete rewriting systems and the complexity of the word problem, *Acta Inform.* **21** (1984), 521–540.
3. C. BEERI, An improvement on Valiant's decision procedure for equivalence of deterministic finite turn pushdown machines, *Theoret. Comput. Sci.* **3** (1976), 305–320.
4. J. BERSTEL, Congruences plus que parfaites et langages algébrique, *in* "Séminaire d'Informatique Théorique (1976–77), Institut de Programmation," pp. 123–147.
5. R. V. BOOK, Confluent and other types of Thue systems; *J. Assoc. Comput. Mach.* **29** (1982), 171–182.
6. R. V. BOOK, A note on special Thue systems with a single defining relation, *Math. Systems Theory* **16** (1983), 57–60.

7. R. V. Book, Thue systems and the Church–Rosser property: Replacement systems, specification of formal languages, and presentations of monoids; *in* "Combinatorics on Words: Progress and Perspectives" (L. Cummings, Ed.), pp. 1–38, Academic Press, New York/London, 1983.
8. R. V. Book, M. Jantzen, and C. Wrathall, Monadic Thue systems, *Theoret. Comput. Sci.* **19** (1982), 231–251.
9. R. V. Book and C. Ó'Dúnlaing, Testing for the Church–Rosser property, *Theoret. Comput. Sci.* **16** (1981), 223–229.
10. H. Bücken, "Anwendung von Reduktionssystemen auf das Wortproblem in der Gruppentheorie," dissertation, Aachen, 1979.
11. J. V. Guttag, D. Kapur, and D. R. Musser, On proving uniform termination and restricted termination of rewriting systems, *SIAM J. Comput.* **12** (1983), 189–214.
12. G. Huet, Confluent reductions: Abstract properties and applications to term rewriting systems, *J. Assoc. Comput. Mach.* **27** (1980), 797–821.
13. G. Huet and D. S. Lankford, "On the Uniform Halting Problem for Term Rewriting Systems, Lab. Rep. 283, INRIA, Le Chesnay, France, 1978.
14. G. Huet and D. Oppen, Equations and rewrite rules; *in* "Formal Language Theory: Perspectives and Open Problems" (R. V. Book, Ed.), pp. 349–405, Academic Press, New York/London, 1980.
15. M. Jantzen, A note on a special one-rule semi-Thue system, *Inform. Process. Letters* **21** (1985), 135–140.
16. D. Kapur, M. S. Krishnamoorthy, R. F. McNaughton, and P. Narendran, An $O(|T|^3)$ algorithm for testing the Church-Rosser property of Thue systems, *Theoret. Comput. Sci.* **35** (1985), 109–114.
17. D. Kapur and P. Narendran, A finite Thue system with decidable word problem and without equivalent finite canonical system, *Theoret. Comput. Sci.* **35** (1985), 337–344.
18. D. Knuth and P. Bendix, Simple word problems in universal algebras, *in* "Computational Problems in Abstract Algebra" (J. Leech, Ed.), pp. 263–297, Pergamon, New York, 1970.
19. P. LeChenadec, "Caninical Forms in Finitely Presented Algebras," Pitman, Wiley, New York, 1986.
20. R. C. Lyndon and P. E. Schupp, "Combinatorial Group Theory," Springer, New York/London, 1977.
21. R. C. Lyndon and M. Schützenberger, The equation $a^M = b^N c^P$ in a free group, *Michigan Math. J.* **9** (1962), 289–298.
22. W. Magnus, A. Karrass, and D. Solitar, "Combinatorial Group Theory," 2nd rev. ed., Dover, New York, 1976.
23. P. Narendran and C. Ó'Dúnlaing, Cancellativity in finitely presented semigroups, *J. Symbolic Comput.*, in press.
24. P. Narendran, C. Ó'Dúnlaing, and H. Rolletschek, Complexity of certain decision problems about congruential languages, *J. Comput. System Sci.* **30** (1985), 343–358.
25. M. H. A. Newman, On theories with a combinatorial definition of equivalence, *Ann. of Math.* **43** (1943), 223–243.
26. M. Nivat (with M. Benois), Congruences parfaites et quasi-parfaites, *in* "Seminaire Dubreil, 25^e Année, 1971–72, 7-01-09."
27. C. Ó'Dúnlaing, "Finite and Infinite Regular Thue Systems," Ph.D. dissertation, Dept. of Math., Univ. of California, Santa Barbara, 1981.
28. C. Ó'Dúnlaing, Infinite regular Thue systems, *Theoret. Comput. Sci.* **25** (1983), 171–192.
29. F. Otto, Some undecidability results for non-monadic Church–Rosser Thue systems, *Theoret. Comput. Sci.* **33** (1984), 261–278.
30. F. Otto, Finite canonical rewriting systems for congruences generated by concurrency relations, *Math. Systems Theory*, in press.
31. F. Otto and C. Whathall, A note on Thue systems with a single defining relation, *Math. Systems Theory* **18** (1985), 135–143.
32. L. G. Valiant, The equivalence problem for deterministic finite-turn pushdown automata, *Inform. and Control* **25** (1974), 123–133.