Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

SEKI - REPORT

A specialized completion procedure for monadic string-rewriting systems presenting groups.

K. Madlener, P. Narendran, F. Otto

SEKI Report SR-90-24

A specialized completion procedure for monadic string-rewriting systems presenting groups

K. Madlener

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
6750 Kaiserslautern
Germany

P. Narendran

Department of Computer Science
State University of New York
Albany, NY 12222
U.S.A.

F. Otto

Fachbereich Mathematik
FG Informatik
Gesamthochschule Kassel
Postfach 101380
3500 Kassel
Germany

Abstract

Based on a simplified test for determining whether a finite monadic string-rewriting system R presenting a group is confluent on the equivalence class of the unity, a procedure for completing a system of this form on $[e]_R$ is derived. The completion procedure transforms monadic presentations in monadic ones by adding monadic rules which are extracted from appropriate infinite regular sets in polynomial time. The correctness and completeness of this procedure are shown.

1. Introduction

Algebraic structures and rewrite methods for effectively computing information on such structures have been studied extensively. In the present paper we are interested in groups that are presented through certain string-rewriting systems. Our interest in groups has mainly two reasons: first of all, the theory of groups is well developed, and secondly, groups are often used as invariants for more complicated structures. For example, there are many results on structural properties of groups, and knowing about these properties may give a lot of additional information on the groups considered [11]. Also it was first for groups that rewrite methods were used for solving the word problem as exemplified by Dehn's algorithm for the class of small cancellation groups ([4],[10],[11]).

Each finitely presented group G can be presented by a finite string-rewriting system R on some alphabet Σ, i.e. G is isomorphic to the factor monoid $\mathfrak{M}_R := \Sigma^*/\overset{*}{\longleftrightarrow}_R$ of the free monoid Σ^* generated by Σ modulo the Thue congruence $\overset{*}{\longleftrightarrow}_R$ induced by R. Although we may restrict the system R to only contain special rules, i.e., one side of each rule is the empty word e, it is in general impossible to obtain much information on the Thue congruence $\overset{*}{\longleftrightarrow}_R$ or on the monoid $\mathfrak{M}_R$ from R. For example, it is even undecidable in general whether the monoid $\mathfrak{M}_R$ presented by a finite special string-rewriting system R is at all a group [16]. In fact, the undecidability of Markov properties can be carried over to the class of monoids that are presented by finite special string-rewriting systems [22].

The situation improves dramatically when attention is restricted to those finite string-rewriting systems R that are Noetherian and confluent. Let $\overset{*}{\rightarrow}_R$ denote the reduction relation induced by R, which is obtained by allowing the rules of R to be only applied only from left to right. Then R is called Noetherian if there are no infinite sequences of reductions modulo R, and it is called **confluent** if, for all $u,v \in \Sigma^*$, $u \overset{*}{\longleftrightarrow}_R v$ implies that $u \overset{*}{\rightarrow}_R w$ and $v \overset{*}{\rightarrow}_R w$ for some $w \in \Sigma^*$. Thus, if R is Noetherian and confluent, then each congruence class mod $\overset{*}{\longleftrightarrow}_R$ contains a unique "minimal" word, and given any word $u \in \Sigma^*$, the minimal word v congruent to u can be computed through rewriting. Hence, the word problem for R is decidable, but also some other problems become decidable in this setting. By restricting the syntactic form of the rules admitted, e.g. by allowing only length-reducing, monadic or special rules, stronger decidability results have been obtained, (c.f., e.g., ([2],[3]) and even some results on structural properties have been derived (see [13] for an overview).

The confluence property for finite Noetherian string-rewriting systems is decidable. Based on a confluence test Knuth and Bendix have developed a completion procedure for rewrite systems [9]. If R is a finite Noetherian string-rewriting system on Σ that is non-confluent, then this completion procedure tries to construct a finite string-rewriting system S on the same alphabet Σ such that S is equivalent to R, i.e., the two congruences $\overset{*}{\longleftrightarrow}_S$ and $\overset{*}{\longleftrightarrow}_R$ coincide, and S is Noetherian and confluent. Specialized completion procedures a la Knuth-Bendix have been developed for groups [10]. However, even if the word problem of R is decidable and if it is allowed to change the ordering used in the completion process, as well as the underlying alphabet, they will not always succeed [21]. By restricting the ordering used or the syntactic form of the rules allowed we are led to completion procedures for certain restricted classes of string-rewriting systems e.g., length-reducing, monadic or special systems. In case of success these procedures may give more information on the group considered or algorithms that are more efficient. Of course, they will not always succeed either.

If the monoid $\mathfrak{M}_R$ is a group, the word problem for R is reducible to the membership problem for the congruence class $[e]_R$. The system R is called confluent on $[e]_R$, if, for all $w \in \Sigma^*$, if $w \overset{*}{\longleftrightarrow}_R e$, implies $w \overset{*}{\rightarrow}_R e$. If a Noetherian system R is confluent on $[e]_R$, then the process of reduction mod R yields an algorithm for testing membership in $[e]_R$ and therewith for solving the word problem for R, even if R is not confluent. In fact Dehn's algorithm for the word problem can be interpreted as computing minimal words modulo a finite length-reducing system R, which in general is only confluent on $[e]_R$. As pointed out in [4],[10] this system can be computed by only using critical pairs which involve the group axioms, provided the given presentation satisfies certain small cancellation conditions. An example of a system R, which presents a group, and which is confluent on $[e]_R$, but for which no equivalent finite string- rewriting system exists, that is Noetherian and confluent, is described in detail in [7] and [12].

The property of confluence on a given congruence class is undecidable in general even for finite length-reducing string-rewriting systems, and for finite monadic string-rewriting systems only algorithms of doubly exponential time complexity are known for deciding this property [18]. So far it is only for finite special string-rewriting systems that this property has been shown to be tractable in [19]. Based on this result a completion procedures may now be developed which, given a finite special string-rewriting system R that is not confluent on $[e]_R$, tries to compute a special system S which is equivalent to R and confluent on $[e]_R$. For the case of finite special string-rewriting systems presenting groups such a specialized completion procedure is described in [20].

Here we consider the case of finite monadic string-rewriting systems that present groups, i.e., each rule is of the form (ℓ,b) with $\ell \in \Sigma^+$ and $b \in \Sigma \cup \{e\}$. For example, let $\Sigma = \{a,b,c\}$ and $R = \{(ab,e),(ba,e),(c^2,e),(acb,c)\}$. Obviously, this system presents a group. In fact, this group is isomorphic to the direct product of the free group F_1 of rank 1 and the cyclic group $\mathbb{Z}_2$ of order 2, and hence, it cannot be presented by a finite monadic and confluent string-rewriting system on any set of generators [13]. However, let $R_o := R \cup \{(bca,c)\}$. Then R_o is a finite monadic system that is equivalent to R. Obviously, R_o is not confluent either, but for all $w \in \Sigma^*$, if $w \overset{*}{\longleftrightarrow}_{R_o} e$, then $w \overset{*}{\rightarrow}_{R_o} e$, i.e. R_o is confluent on $[e]_{R_o}$. Hence, the process of reduction mod R_o gives a linear-time algorithm for solving the word problem for R. In fact many decision problems can be solved, when they are restricted to the class of finite monadic string-rewriting systems R that present groups, and that are confluent on $[e]_R$. For example,

all problems which can be expressed through linear sentences in the sense of Book [2] can be solved in a uniform way in this setting [14]. In addition, the class of groups that can be presented by these systems is strictly larger than the class of groups that can be presented by finite length-reducing and confluent string-rewriting systems. In fact, the structural and language theoretical properties of this class of groups are also well-known, since it is exactly the class of context-free groups [1], which has also been characterized as the class of finite extensions of finitely generated free groups [15].

Here we present a procedure which, given a finite monadic string-rewriting system R on Σ such that the monoid $\mathfrak{M}_R$ is a group as input, tries to construct a finite monadic system S on Σ such that S is equivalent to R and confluent on $[b]_R$ for all $b \in \Sigma \cup \{e\}$. This procedure consists of two subroutines called CONTEXT_RESOLVING and NORMALIZATION, where the former introduces new monadic and special rules to make the system confluent on the relevant equivalence classes, while the latter deletes superfluous rules in order to keep the system as small as possible. It is shown that this procedure either terminates with a finite monadic system S, or it enumerates an infinite monadic system S. In either case, S is equivalent to R and confluent on $[b]_R$ for all $b \in \Sigma \cup \{e\}$. In addition, our procedure terminates whenever there exists a finite monadic system that is equivalent to R and that is confluent on $[e]_R$. Thus, we have a completion procedure that is correct and complete.
This paper is organized as follows. After establishing the necessary notation in Section 2, we derive some conditions in Section 3 that are necessary and sufficient to guarantee confluence on [e] for finite monadic string-rewriting systems R presenting groups. Since these conditions can be verified in polynomial time, we thus have a polynomial-time algorithm for deciding confluence on $[e]_R$ for this class of string-rewriting systems. In Section 4 the announced completion procedure is presented together with the necessary proofs and some examples. Finally, in Section 5 we point out the relation to the notion of symmetrized group-presentation as it is considered in small cancellation theory ([10],[11]). Also some possible improvements of our completion procedure and some problems for future research are mentioned.

2. Preliminary results

After establishing notation we present some basic results about reductions in finite monadic string-rewriting systems. We assume the reader to be familiar with the foundations of automata theory as presented for example by Hopcroft and Ullman [6], and the theory of string-rewriting systems. Therefore, we repeat only those definitions and results which will be used here. For more details and background information, Book's seminal paper [3] may be consulted.

Let Σ be a finite alphabet. Then Σ^* denotes the set of words over Σ including the empty word e. A **monadic string-rewriting** system R on Σ is a subset of $\Sigma^+ \times (\Sigma \cup \{e\})$, where $\Sigma^+ = \Sigma^* \setminus \{e\}$ denotes the set of non-empty words over Σ. The elements (ℓ, b) of R are called (rewrite) **rules**. If $b = e$, the rule is called **special**. For all $u,v \in \Sigma^*$ and $(\ell,b) \in R$, $u\ell v \rightarrow_R ubv$, i.e., $\rightarrow_R$ is the **single-step reduction relation** induced by R. Its reflexive and transitive closure $\xrightarrow{*}_R$ is the **reduction relation** induced by R. For $u,v \in \Sigma^*$, if $u \xrightarrow{*}_R v$, then u is an **ancestor** of v, and v is a **descendant** of u. By $\langle v \rangle_R$ we denote the set of all ancestors of v, and $\Delta^*_R(u)$ denotes the set of all descendants of u. For a subset $L \subseteq \Sigma^*$, $\langle L \rangle = \bigcup_{u \in L} \langle u \rangle$, and $\Delta^*(L) = \bigcup_{u \in L} \Delta^*(u)$.

Since we allow monadic rules of the form (a,b), where $a,b \in \Sigma$ are both letters, we will always assume a fixed ordering $>$ on Σ, and for each rule of this form, we will require that $a > b$. This slightly extends the usual definition of monadic systems, but $\xrightarrow{*}_R$ is terminating and the usual properties still hold . For example, if R is finite and monadic, and if $L \subseteq \Sigma^*$ is a regular set that is given through a non-deterministic finite state acceptor (nfsa) $\mathfrak{A}$, then the set $\Delta_R^*(L)$ is regular as well, and an nfsa $\mathfrak{B}$ for this set can be constructed in polynomial time [2]. If there is no word $y \in \Sigma^*$ such that $x \to_R y$, then x is called **irreducible**, otherwise, it is **reducible**. If R is finite, then the set IRR(R) of irreducible words is regular, and from R a deterministic finite state acceptor (dfsa) for this set can be obtained in polynomial time [5]. By $\xleftrightarrow{*}_R$ we denote the equivalence relation induced by $\to_R$, which is actually a congruence on Σ^*. It is called the **Thue congruence** generated by R. For $w \in \Sigma^*$, $[w]_R = \{u \in \Sigma^* \mid u \xleftrightarrow{*}_R w\}$ is the **congruence class** of w mod R. The set $\{[w]_R \mid w \in \Sigma^*\}$ of congruence classes forms a **monoid** $\mathfrak{M}_R$ under the operation $[u]_R \circ [v]_R = [uv]_R$ with identity $[e]_R$. This monoid is uniquely determined (up to isomorphism) by Σ and R, and hence, whenever M is a monoid that is isomorphic to $\mathfrak{M}_R$, we call the ordered pair $(\Sigma;R)$ a (**monoid-**) **presentation** of M with generators Σ and defining relations R. Two systems R and S on the same alphabet Σ are called **equivalent** if they generate the same Thue congruence, i.e. $\xleftrightarrow{*}_R = \xleftrightarrow{*}_S$ and $\mathfrak{M}_R = \mathfrak{M}_S$. The monoid $\mathfrak{M}_R$ is a **group** if and only if, for each letter $a \in \Sigma$, there exists a word $u_a \in \Sigma^*$ such that $au_a \xleftrightarrow{*}_R e$. In this case there exists a function $^{-1}: \Sigma^* \to \Sigma^*$ such that for all $w \in \Sigma^*$, w^{-1} is a formal inverse of w, i.e., $ww^{-1} \xleftrightarrow{*}_R w^{-1}w \xleftrightarrow{*}_R e$. In fact , for every letter $a \in \Sigma$ a candidate w_a for u_a of length less than $(\max\{|\ell|: (\ell,b) \in R\})^{|\Sigma|}$ can be computed from R [17].

A subset $L \subseteq \Sigma^*$ is **closed under cyclic permutations** if $uv \in L$ implies $vu \in L$ for all $u,v \in \Sigma^*$. The set $[e]_R$ is closed under cyclic permutations if $\mathfrak{M}_R$ is a group.

A string-rewriting system R on Σ is **confluent on $[w]_R$** for some word $w \in \Sigma^*$, if there exists a word $w_o \in IRR(R)$ such that $\Delta_R^*([w]_R) \cap IRR(R) = \{w_o\}$. Thus, R is confluent on $[w]_R$ if all words in that class reduce to the same irreducible word, which then can serve as a normal form for this class. R will be called **weakly confluent** if it is confluent on $[b]_R$ for all $b \in \Sigma \cup \{e\}$, and it will be called **e-confluent** if it is confluent on $[e]_R$.

From now on we will assume that the monoid $\mathfrak{M}_R$ is a group. This group is called **context-free** if the set $[e]_R \subseteq \Sigma^*$ is a context-free language. This property is independent of the actually chosen finite presentation. The importance of monadic string-rewriting systems that are weakly confluent or e-confluent is due to their relation to context-free groups and on the decidability of the linear sentences of Book for this class [14]. Autebert, Boasson, and Senizergues [1] established the following fundamental result on context-free groups.

Theorem 2.1 [Autebert et al. 87]. A finitely generated group $\mathfrak{G}$ is context-free if and only if it has a presentation of the form $(\Sigma;R)$, where R is a finite, monadic and weakly confluent string-rewriting system on Σ.

An algebraic characterization of this class of groups has been given by Muller and Schupp [15] using the concept of virtually free groups (i.e. groups which contain a finitely generated free subgroup of finite index). So it is fairly easy to construct examples of such groups. They all have presentations by finite and confluent string-rewriting systems, provided appropriate

orderings are used, which in general are not length-compatible. Note that the class of groups that can be presented by finite, monadic and confluent systems is a proper subclass of the context-free groups. (See the example of Section 1 and [13]).

Example 2.2. a) Let $\Sigma = \{a,b,c\}$ and $R = \{(ab,e),(ba,e),(c^2,e),(aca,c)\}$. Then $(\Sigma;R)$ presents a group which is an extension of the free group of rank 1 and the cyclic group $\mathbb{Z}_2$. R is not confluent, since $ac \overset{*}{\longleftrightarrow}_R cb$ and $ca \overset{*}{\longleftrightarrow}_R bc$. It is neither confluent on $[e]_R$, since $cbcb \overset{*}{\longleftrightarrow}_R e$, nor on the congruence class of any letter, because $bcb \overset{*}{\longleftrightarrow}_R c$, $cac \overset{*}{\longleftrightarrow}_R b$ and $cbc \overset{*}{\longleftrightarrow}_R a$. By adding the rule (bcb,c) we get a system which is confluent on [e] and [b], and if (cac,b) and (cbc,a) are also taken, we get a weakly confluent system. (See also Example 4.5).
b) Let $\Sigma = \{a,b,c\}$ and $R = \{(ab,e),(ba,e),(c^3,e),(c^2ac,a),(c^2bc,b)\}$. Then $(\Sigma;R)$ presents a group isomorphic to $\mathbb{Z}\times\mathbb{Z}_3$, the direct product of $\mathbb{Z}$ with the cyclic group of order 3. For all $n \geq 1$, $ca^nc^2b^n \overset{*}{\longleftrightarrow}_R e \overset{*}{\longleftrightarrow}_R cb^nc^2a^n$. Since $bc^2a \overset{*}{\longleftrightarrow}_R c^2 \overset{*}{\longleftrightarrow}_R ac^2b$, no factor u of $ca^nc^2b^n$ or $cb^nc^2a^n$ satisfying $1 < |u| \leq n$ is congruent to any letter. Thus, there is no finite monadic system S that is both equivalent to R and confluent on $[e]_R$. Since $\mathbb{Z}\times\mathbb{Z}_3$ is a context-free group, there must be a monadic presentation of this group which is confluent on [e]. In fact, by introducing a new letter d and the rules $(c^2,d),(cd,e),(dc,e),(d^2,c)$ together with $\{(axb,x),(bxa,x) \mid x \in \{c,d\}\}$ we get a different presentation of $\mathbb{Z}\times\mathbb{Z}_3$ for which confluence on [e] and even weak confluence can be shown.

Confluence on one equivalence class is much harder to decide than confluence everywhere. In fact, in [18] it is shown that this property is undecidable even for length-reducing systems, while for monadic systems it may be decided using the decidability of the equivalence problem for finite-turn deterministic pushdown automata. For stating this result in detail we need some more notation. Let R be a finite monadic string-rewriting system on Σ, and let (ℓ_1,b_1) and (ℓ_2,b_2) be two rules of R. If $\ell_1 = x\ell_2y$ for some $x,y \in \Sigma^*$, or if $\ell_1x = y\ell_2$ for some $x,y \in \Sigma^*$ satisfying $0 < |y| < |\ell_1|$, then the pair (b_1,xb_2y), respectively (b_1x,yb_2), is called a **critical pair** of R. By UCP(R) we denote the set $\{(x,y) \mid (x,y)$ is a critical pair of R such that $\Delta_R^*(x) \cap \Delta_R^*(y) = \emptyset\}$ of **unresolvable critical pairs** of R. Obviously, this set can be computed in polynomial time. For $w \in IRR(R)$ we define a language $L_u(w)$ as follows: $L_u(w) = \{x\#y \mid x,y \in IRR(R), xuy \overset{*}{\rightarrow}_R w\}$. Here # is an additional letter not in Σ. Then $x\#y \in L_u(w)$ if (x,y) is an irreducible context of u in $\langle w\rangle_R$. Using sets of this form the confluence on $[w]_R$ can be characterized as follows:

Proposition 2.3 [Otto 87]. Let R be a monadic string-rewriting system on Σ, and let $w \in IRR(R)$. Then the following two statements are equivalent:
(i) The system R is confluent on $[w]_R$.
(ii) $\forall (u,v) \in UCP(R)$: $L_u(w) = L_v(w)$.

Since the sets $L_u(w)$ are in general context-free languages, this characterization will not be useful in a completion procedure. In the next section we will derive an easier test by using the fact that $\mathfrak{M}_R$ is a group.
As a first simplification we would like to keep the system R as small as possible. A string-rewriting system R is called **reduced** if, for each rule $(\ell,r) \in R$, $r \in IRR(R)$ and $\ell \in IRR(R\setminus\{(\ell,r)\})$. In general there need not exist a reduced system which is equivalent to a given one. However,

such a reduced system exists when the ordering on Σ^* is total or when the system R is confluent on $[r]_R$ for every right-hand side r of R. The same holds for monadic systems R as we shall see.
Let R be a finite monadic string-rewriting system on Σ. By replacing every right-hand side by an irreducible descendant of it, we get a finite monadic system R_1 on Σ, which can be obtained in polynomial time from R, such that $IRR(R_1) = IRR(R)$, $\xrightarrow{*}_{R_1} \subseteq \xrightarrow{*}_R$ and $\xleftrightarrow{*}_{R_1} = \xleftrightarrow{*}_R$. Also R_1 is weakly confluent or e-confluent if and only if R has this property. Recall that each right-hand side of a rule of R_1 is irreducible, but still R_1 need not be reduced.
So assume that there are two rules $(\ell_1,b_1),(\ell_2,b_2) \in R_1$ such that $\ell_1 = x\ell_2y$ for some $x,y \in \Sigma^*$. Let $R_2 := R_1 \setminus \{(\ell_1,b_1)\}$. Then $\xrightarrow{*}_{R_2} \subseteq \xrightarrow{*}_{R_1}$ and $IRR(R_2) = IRR(R_1)$. In general R_1 and R_2 will not be equivalent. Nevertheless, if R_1 is weakly confluent, then it is confluent on [e] and [b] for every right-hand side b of a rule. In particular, for all $u \in \Sigma^*$, $u \xleftrightarrow{*}_{R_1} e$ or $u \xleftrightarrow{*}_{R_1} b$ implies $u \xrightarrow{*}_{R_1} e$, respectively $u \xrightarrow{*}_{R_1} b$. Since R_2 is terminating, there exists a word $v \in IRR(R_2)$ such that $u \xrightarrow{*}_{R_2} v$. Because of $\rightarrow_{R_2} \subseteq \rightarrow_{R_1}$ and $IRR(R_2) = IRR(R_1)$, we see that $v = e$, respectively $v = b$. Hence $[e]_{R_1} = [e]_{R_2}$ and $[b]_{R_1} = [b]_{R_2}$, and R_2 is also weakly confluent. But then $\ell_1 \xrightarrow{*}_{R_2} b_1$, so R_2 and R_1 are equivalent (in fact $\xrightarrow{*}_{R_1} = \xrightarrow{*}_{R_2}$).
If R_1 is only e-confluent, then as above $[e]_{R_1} = [e]_{R_2}$, and R_2 is also e-confluent. If $b_1 = e$, then again R_2 is equivalent to R_1. So let $b_1 \in \Sigma$. Since $\mathfrak{M}_{R_1}$ is a group, $b_1^{-1}\ell \xleftrightarrow{*}_{R_1} e$ and $b_1b_1^{-1} \xleftrightarrow{*}_{R_1} e$, so $b_1^{-1}\ell \xrightarrow{*}_{R_2} e$ and $b_1b_1^{-1} \xrightarrow{*}_{R_2} e$. Thus, we get $\ell \xleftrightarrow{*}_{R_2} b_1b_1^{-1}\ell \xleftrightarrow{*}_{R_2} b_1$, i.e. R_2 is equivalent to R_1.

Thus, if R_1 is weakly confluent or e-confluent, we obtain a reduced finite monadic system R_2 that is equivalent to R_1 and that has the same confluence property, by simply deleting those rules $(\ell,b) \in R_1$ one by one, for which ℓ is reducible by some other rule of R_1. Notice that the conditions

$$\ell_1 \xrightarrow{*}_{R_2} b_1 \qquad \text{and} \qquad b_1^{-1}\ell \xrightarrow{*}_{R_2} e \quad \text{and} \quad b_1b_1^{-1} \xrightarrow{*}_{R_2} e$$

can be verified in polynomial time. If they are not satisfied, then from the above discussion we can conclude that either R_1 is not weakly confluent or R_1 is not e-confluent, respectively. So when testing weak- or e-confluence of a monadic system R we may assume that R is reduced. When completing such a system, $xb_2y \xleftrightarrow{*} b_1$ can always be oriented so that a monadic rule results. If no letter is congruent to e or to a different letter, then, if $\ell_1 \not\xrightarrow{*}_{R_2} b_1$, then (xb_2y,b_1) has to be added to preserve the congruence, and then the resulting system R_2 even satisfies $\xrightarrow{*}_{R_1} \subseteq \xrightarrow{*}_{R_2}$. This will not be the case if a letter becomes reducible to a different letter or to e.

3. A polynomial test for e-confluence

Let R be a finite reduced monadic string-rewriting system on Σ such that $\mathfrak{M}_R$ is a group. If R is confluent on [e], then for each $a \in \Sigma$, there exists a word $u_a \in \Sigma^*$ such that $au_a \xrightarrow{*}_R e$. In fact, $au_a \xrightarrow{i}_R e$ for some $i \leq |\Sigma|$, and hence, $|u_a| \leq |\Sigma| \cdot (\mu-1)$, where $\mu = \max\{|\ell| \mid (\ell,b) \in R\}$. Further, if $au \xrightarrow{*}_R e$, then $ua \xrightarrow{*}_R e$, since [e] is closed under cyclic permutations, and therewith $\langle e \rangle$ has also this property. If any one of these conditions is not satisfied, then R is not e-confluent. It is easy to see that R is e-confluent iff for all $a \in \Sigma$ and $w \in [a^{-1}] \cap IRR(R)$: $aw \xrightarrow{*}_R e$. This set might not be easy to construct, so we will use an approximation of it, namely the set of right-inverses which will play a central role in a test for e-confluence.

Definition 3.1. For $u \in \Sigma^*$, let $RI_R(u)$, the set of **right inverses of** u, consist of all words v such that $uv \xrightarrow{*}_R e$, and no step of the reduction sequence is performed entirely within u or within v. To be precise

$RI_R(u) = \{v \in \Sigma^* \mid \exists k \geq 1\ \exists u_1,\dots,u_k, v_1,\dots,v_k \in \Sigma^*$ with $u = u_k\dots u_1$, $v = v_1\dots v_k$, $(u_1v_1,a_1) \in R$, $(u_2a_1v_2,a_2) \in R,\dots,(u_ka_{k-1}v_k,e) \in R$ and $u_i \neq e \neq v_i$ if $a_i = e\}$.

If $u,v \in IRR(R)$ and $uv \xrightarrow{*}_R e$, then $v \in RI_R(u)$ since R is monadic. $RI_R(u)$ may be infinite, but it is easy to compute.

Lemma 3.2. For every $u \in \Sigma^*$, $RI_R(u)$ is a regular set. From R and u a nfsa for this set can be constructed in polynomial time.

Proof: Let F be the set of all proper factors of left-hand sides of rules in R. Define a nfsa $\mathfrak{A}(u) = (Q,\Sigma,\delta,q_0,q_a)$ as follows:

- $Q = \{(u_1,\ell_1) \mid \exists u_2 \in \Sigma^*\ \exists b \in \Sigma \cup \{e\}$: $u_1 = u_2b$, u_2 is a prefix of u and $\ell_1 \in F\}$
- $q_0 := (u,e)$ - $q_a := (e,e)$ and
- $(u_2,\ell_2) \in \delta((u_1,\ell_1),a)$ iff $(u_1 = u_2$ and $\ell_2 = \ell_1a \in F)$ or $\ell_1a \in F$ and $\exists u_3,u_4 \in \Sigma^*$: $u_1 = u_3u_4$, $u_4 \neq e$, $u_4\ell_1a \xrightarrow{+}_R b$ and $u_2 = u_3b$, $\ell_2 = e$ for some $b \in \Sigma \cup \{e\}$.

Then $\mathfrak{A}(u)$ can be constructed in polynomial time and $(e,e) \in \delta((u,e),v)$ iff $v \in RI_R(u)$. So $L(\mathfrak{A}(u)) = RI_R(u)$. ∎

Now we are ready to formulate a test for e-confluence.

Theorem 3.3. R is confluent on $[e]$ iff the following conditions are satisfied:

(i) $\forall a \in \Sigma$: $\Delta_R^*(RI_R(a)\cdot a) \cap IRR(R) = \{e\}$ if $RI_R(a) \neq \emptyset$ and

(ii) $\forall (p,q) \in UCP(R)\ \forall p_1 \in \Delta_R^*(p)\ \forall q_1 \in \Delta_R^*(q)$:
$\Delta_R^*(q\cdot RI_R(p_1)) \cap IRR(R) = \{e\} = \Delta_R^*(p\cdot RI_R(q_1)) \cap IRR(R)$ if $RI_R(p_1) \neq \emptyset \neq RI_R(q_1)$.

Proof: Assume that R is confluent on $[e]_R$.

(i) Let $a \in \Sigma$ and $v \in RI_R(a)$, i.e. $av \xrightarrow{*}_R e$. Since $\mathfrak{M}_R$ is a group, $va \xleftrightarrow{*}_R e$ and $\Delta_R^*(va) \subseteq [e]$. Because R is confluent on $[e]_R$, $\Delta_R^*(va) \cap IRR(R) = \{e\}$ holds.

(ii) Let $(p,q) \in UCP(R)$, let $p_1 \in \Delta_R^*(p)$ and let $v \in RI_R(p_1)$, i.e. $p_1v \xrightarrow{*}_R e$. Then $qv \xleftrightarrow{*}_R p_1v \xleftrightarrow{*}_R e$ and since R is confluent on $[e]_R$, $\Delta_R^*(qv) \cap IRR(R) = \{e\}$.

So the conditions are necessary.

To prove the converse implication, assume that conditions (i) and (ii) are satisfied.

Claim 1: $\langle e\rangle_R$ is closed under cyclic permutations.

Proof: Assume that this is not the case and let $x \in \Sigma^*$ be of minimal length and minimal with respect to $\xrightarrow{*}_R$, such that $x \xrightarrow{*}_R e$, but $x' \not\xrightarrow{*}_R e$ for some cyclic permutation x' of x. Then there exists a cyclic permutation $y = az$ of x, where $a \in \Sigma$ and $z \in \Sigma^*$, such that $y = az \xrightarrow{*}_R e$, but $za \not\xrightarrow{*}_R e$.

In the reduction sequence $az \xrightarrow{*}_R e$ no step is entirely performed within a or within z, since otherwise x would not be minimal. Hence, $z \in RI_R(a)$. By condition (i) this implies that $za \xrightarrow{*}_R e$, contradicting our choice. Thus, $\langle e\rangle_R$ is closed under cyclic permutations. ∎

Claim 2: R is confluent on $[e]_R$.
Proof: Let $(p,q) \in UCP(R)$. By Proposition 2.3 we must show that, for all $x,y \in IRR(R)$, $xpy \xrightarrow{*}_R e$ iff $xqy \xrightarrow{*}_R e$. By Claim 1, $\langle e \rangle$ is closed under cyclic permutations, i.e. $pyx \xrightarrow{*}_R e$. Since R is monadic, this means that there exist words $p_1 \in \Delta_R^*(p)$ and $w \in \Delta_R^*(yx)$ such that $p_1 w \xrightarrow{*}_R e$, and each step in this reduction straddles the boundary between p_1 and w. Hence $w \in RI_R(p_1)$. By condition (ii) this implies that $qw \xrightarrow{*}_R e$ and so $qyx \xrightarrow{*}_R qw \xrightarrow{*}_R e$. Again by Claim 1 this means that $xqy \xrightarrow{*}_R e$. By symmetry we obtain: $xpy \xrightarrow{*}_R e$ iff $xqy \xrightarrow{*}_R e$. Thus, R is in fact confluent on $[e]_R$. ∎
So conditions (i) and (ii) guarantee confluence on [e]. ∎

According to the discussion at the end of Section 2, we may assume that R is reduced. Thus, if $(p,q) \in UCP(R)$, then there exist words $x,y \in \Sigma^*$ and rules $(\ell_1,b_1),(\ell_2,b_2) \in R$ such that $\ell_1 x = y\ell_2$, $0 < |y| < |\ell_1|$, $p = b_1 x$ and $q = y b_2$. In particular, y is a proper prefix of ℓ_1, and x is a proper suffix of ℓ_2, i.e. x and y are both irreducible. If $b_1 = e$, then p is irreducible, and if $b_2 = e$, then q is irreducible. Otherwise, the sets $\Delta_R^*(p)$ and $\Delta_R^*(q)$ are of size bounded from above by $\mu \cdot |\Sigma|$, where $\mu = \max\{|\ell| \mid (\ell,b) \in R\}$. Hence, to verify the conditions (i) and (ii) of Theorem 3.3 only a polynomially bounded number of tests must be performed. Since we can construct nfsa's recognizing the involved testsets in polynomial time, we obtain the following result.

Corollary 3.4: The following problem is decidable in polynomial time.
INSTANCE: A finite monadic string-rewriting system R on Σ such that the monoid $\mathfrak{M}_R$ is a group.
QUESTION: Is R confluent on $[e]_R$?

Let us now consider the problem of deciding weak confluence of such a system. We may assume that R is confluent on [e] and reduced, and so we only have to check the confluence of R on the congruence classes of irreducible letters. Let b be such a letter and let b^{-1} be an irreducible inverse of b. Then $RI_R(b^{-1}) \cap IRR(R) = [b] \cap IRR(R)$.

The inclusion $RI_R(b^{-1}) \cap IRR(R) \subseteq [b] \cap IRR(R)$ is clear. So let $w \in [b] \cap IRR(R)$. Then $b^{-1}w \xleftrightarrow{*}_R b^{-1}b \xleftrightarrow{*}_R e$. Since R is monadic and confluent on [e], and since both b^{-1} and w are irreducible, we get $w \in RI_R(b^{-1})$ i.e., $w \in RI_R(b^{-1}) \cap IRR(R)$. Thus, we have the following characterization.

Theorem 3.5. R is weakly confluent iff conditions (i) and (ii) of Theorem 3.3 and
(iii) $\forall a \in \Sigma \cap IRR(R)$: $RI_R(a^{-1}) \cap IRR(R) = \{a\}$ for some irreducible inverse a^{-1} of a
are satisfied.

Since e-confluence is decidable in polynomial time, and since (iii) of Theorem 3.5 is also decidable in polynomial time we get:

Corollary 3.6: The following problem is decidable in polynomial time.
INSTANCE: A finite monadic string-rewriting system R on Σ such that the monoid $\mathfrak{M}_R$ is a group.
QUESTION: Is R weakly confluent ?

One might ask whether e-confluence implies weak confluence. This is not the case as shown by Example 2.2a). But the existence of an e-confluent monadic system R implies the existence of a weakly confluent monadic system R' which is equivalent to R. In fact, R' may be constructed in polynomial time from R. W.l.o.g. we may assume that R is reduced. Thus, for each reducible letter b, R contains exactly one rule with left-hand side b, and this letter does not occur in any other rule. Let a be the smallest irreducible letter (Σ is ordered) such that R is not confluent on [a], and let a^{-1} be an irreducible inverse of a. Then $RI_R(a^{-1}) \cap IRR(R) = [a] \cap IRR(R)$ properly contains {a}. This set is finite in this case, since otherwise we would have irreducible words $ux^nv \overset{*}{\longleftrightarrow}_R uv \overset{*}{\longleftrightarrow}_R a$. Since $\mathfrak{M}_R$ is a group, and since R is confluent on [e], this would imply that $x^n \overset{*}{\longleftrightarrow}_R e$ contradicting the fact that ux^nv is irreducible. So let $w_1,...,w_k$ be the irreducible words in [a] different from a, which may be computed in polynomial time. Then $R \cup \{(w_i,a) \mid i = 1,...,k\}$ is monadic, e-confluent and also confluent on [a]. This process may be iterated with the next irreducible letter on which the resulting reduced system is not confluent. For the resulting system R' we obtain $\overset{*}{\rightarrow}_R \subseteq \overset{*}{\rightarrow}_{R'}$, R' is equivalent to R, and R' is weakly confluent.

4. The completion procedure

Based on our confluence test, we now present a procedure which on input a finite monadic string-rewriting system R_0 presenting a group, tries to construct a weakly confluent monadic system R that is equivalent to R_0. This procedure contains two main subroutines: NORMALIZATION and CONTEXT_RESOLVING. The first one realizes the reduction process explained at the end of Section 2. The second one introduces new rules if necessary based on the test of Theorem 3.5. There are three types of regular sets which may contribute new rules depending on the condition actually checked:

$E_a := \Delta_R^*(RI_R(a) \cdot a) \cap IRR(R) \setminus \{e\}$ for $a \in \Sigma$
$L_{a^{-1}} := RI_R(a^{-1}) \cap IRR(R) \setminus \{a\}$ for irreducible $a \in \Sigma$
$Sp_i := \Delta_R^*(q \cdot RI_R(p_i)) \cap IRR(R) \setminus \{e\}$, and
$Sq_i := \Delta_R^*(p \cdot RI_R(q_i)) \cap IRR(R) \setminus \{e\}$
for $(p,q) \in UCP(R)$ and $p_i \in \Delta_R^*(p)$, $q_i \in \Delta_R^*(q)$.

Since these sets might be infinite, we have to determine a finite number of special and monadic rules which can reduce all the computed divergences. For doing this, notice that E_a, Sp_i and Sq_i are subsets of [e], and that $L_{a^{-1}} \subseteq [a]$. From the nfsa's for these sets a finite number of simple accepting paths and of simple loops which generate all accepting paths may be determined in polynomial time. Since we have a group, the irreducible words corresponding to simple loops are equivalent to e. (The argument for this being similar to the one at the end of Section 3). In the nfsa for $L_{a^{-1}}$ the irreducible words corresponding to simple accepting paths are equivalent to a, so they lead to proper monadic rules.
Let GENSPATH and GENSLOOP be procedures which compute the irreducible words corresponding to the simple paths, respectively simple loops, when applied to a nfsa accepting one of the above sets. Since the subroutine CONTEXT_RESOLVING may introduce new rules which destroy the property of being reduced and also add new unresolved critical pairs, we have to keep applying both subroutines until a stable system is obtained.

Procedure 4.1:

INPUT: A finite monadic string-rewriting system R on an ordered alphabet Σ such that the monoid $\mathfrak{M}_R$ is a group.

begin $i \leftarrow 0$; $R_i \leftarrow R$;
NORMALIZATION: Reduce right-hand sides using first applicable rule;
while $\exists \ell_1,\ell_2,x,y \in \Sigma^*$: $\ell_2 = x\ell_1 y \wedge (\ell_1,b_1) \in R_i \wedge (\ell_2,b_2) \in R_i$
do
begin $R_i \leftarrow R_i \setminus \{(\ell_2,b_2)\}$;
if $b_2 \notin \Delta^*_{R_i}(xb_1y)$ **then** $R_i \leftarrow R_i \cup \{\langle xb_1y,b_2\rangle\}$;
Reduce right-hand sides
end

comment: At this point the system R_i is reduced. Here, $\langle xb_1y,b_2\rangle$ is the monadic rule resulting from this pair using the ordering on Σ if both sides are letters.

CONTEXT_RESOLVING: For all $a \in \Sigma \cap IRR(R_i)$ compute irreducible inverse a^{-1};
Compute $UCP(R_i)$; $R_i' \leftarrow \emptyset$;
For all $a \in \Sigma$ **do** $\{E_a \leftarrow \Delta^*_R(RI_R(a)\cdot a) \cap IRR(R)\setminus\{e\}$;
$\hat{E}_a \leftarrow GENSPATH(E_a) \cup GENSLOOP(E_a)$;
$R_i' \leftarrow R_i' \cup \{(\ell,e) \mid \ell \in \hat{E}_a\}\}$
For all $a \in \Sigma \cap IRR(R_i)$ **do** $\{L_{a^{-1}} \leftarrow RI_R(a^{-1}) \cap IRR(R)\setminus\{a\}$;
$\hat{M}_{a^{-1}} \leftarrow GENSPATH(L_{a^{-1}})$;
$R_i' \leftarrow R_i' \cup \{\langle \ell,a\rangle \mid \ell \in \hat{M}_{a^{-1}}\}$;
$\hat{L}_{a^{-1}} \leftarrow GENSLOOP(L_{a^{-1}})$;
$R_i' \leftarrow R_i' \cup \{(\ell,e) \mid \ell \in \hat{L}_{a^{-1}}\}\}$
For all $(p,q) \in UCP(R_i)$ **do** $\{$For all $p_i \in \Delta^*_R(p)$, $q_i \in \Delta^*_R(q)$ **do**
$\{Sp_i \leftarrow \Delta^*_R(q\cdot RI_R(p_i)) \cap IRR(R)\setminus\{e\}$;
$Sq_i \leftarrow \Delta^*_R(p\cdot RI_R(q_i)) \cap IRR(R)\setminus\{e\}$;
$\hat{S}p_i \leftarrow GENSPATH(Sp_i) \cup GENSLOOP(Sp_i)$;
$\hat{S}q_i \leftarrow GENSPATH(Sq_i) \cup GENSLOOP(Sq_i)$;
$R_i' \leftarrow R_i \cup \{(\ell,e) \mid \ell \in \hat{S}p_i \cup \hat{S}q_i\}\}\}$

comment: The new rules are now collected in R_i', all left- and right-hand sides of the rules in R_i' are R_i-irreducible.

if $R_i' \neq \emptyset$ **then** $\{R_{i+1} \leftarrow R_i \cup R_i'$;
$i \leftarrow i+1$;
goto NORMALIZATION$\}$

comment: At this point R_i is weakly confluent and reduced

OUTPUT: R_i
end

We claim that the above procedure determines a finite monadic system R_i that is weakly confluent and that is equivalent to R whenever an equivalent e-confluent monadic system exists. Otherwise it enumerates an infinite monadic system R_∞ having both these properties.
The only place in the procedure where rules are deleted is in the NORMALIZATION subroutine. If a rule is deleted here it might be replaced by smaller rules (where rules are compared first by their left-hand sides using length-lex-ordering and for equal left-hand sides by comparing their right-hand sides). In fact for any proof using the deleted rule there is a strictly smaller proof in the resulting system using the induced proof ordering. So, when applied to some system R, NORMALIZATION always terminates with a reduced system R' which is equivalent to R and $IRR(R') \subseteq IRR(R)$. So a rule which was once deleted will never be introduced again, neither by NORMALIZATION nor by CONTEXT_RESOLVING. If no letter is equivalent to e or to a different letter then $\xrightarrow{*}_R \subseteq \xrightarrow{*}_{R'}$.

Lemma 4.2. Let R be a finite monadic string-rewriting system on Σ such that the monoid $\mathfrak{M}_R$ is a group. If Procedure 4.1 terminates on input R, then it yields a finite monadic system R_i on Σ that is equivalent to R, weakly confluent and reduced.
Proof: On input R, Procedure 4.1 computes a sequence of finite monadic systems $R_0, R_1, R_2, \ldots$ which are the systems after NORMALIZATION, satisfying the following conditions for $j = 0,1,2,\ldots$
- R_j is equivalent to R
- $IRR(R_{j+1}) \subseteq IRR(R_j)$
- R_j is reduced.

This stems from the fact that in each step only a finite set of monadic rules, which are correct, is added. Procedure 4.1 terminates when $R_i' = \emptyset$, i.e. when no rule is added by the subroutine **CONTEXT_RESOLVING** applied to R_i. By Theorem 3.5 this happens iff R_i is weakly confluent. ∎

Thus, whenever Procedure 4.1 terminates, the system R_i constructed has indeed all the properties we want. It remains to show that this algorithm does terminate whenever a monadic system S exists that is finite, equivalent to R, and confluent on $[e]_R$. Because of the discussion at the end of Section 3 we may assume that S is in fact weakly confluent. Notice also that the existence of such a system does not depend on the fixed ordering on Σ, since a different ordering induces just a renaming. As a first step towards proving this fact, we analyse the situation when Procedure 4.1 does not terminate.

Lemma 4.3. Let R be a finite monadic string-rewriting system on Σ such that the monoid $\mathfrak{M}_R$ is a group. If Procedure 4.1 does not terminate on input R, then it enumerates an infinite monadic system R_∞ that is reduced, equivalent to R and weakly confluent.
Proof: Assume that Procedure 4.1 does not terminate on input R. Then it enumerates an infinite sequence $R_0, R_1, R_2, \ldots$ of finite monadic string-rewriting systems on Σ satisfying the following conditions for all $j \geq 0$:
- R_j is equivalent to R and reduced
- $IRR(R_{j+1}) \subset IRR(R_j)$
- $\langle e \rangle_{R_j} \subset \langle e \rangle_{R_{j+1}}$
- $\langle a \rangle_{R_j} \subseteq \langle a \rangle_{R_{j+1}}$ for $a \in \Sigma \cap IRR(R_{j+1})$.

The last two properties are easy to prove if $\xrightarrow{*}_{R_i} \subseteq \xrightarrow{*}_{R_{i+1}}$, which is the case if no letter becomes reducible to e or to a different letter. Otherwise a derivation $w \xrightarrow{*}_{R_j} a$ for $a \in (\Sigma \cup \{e\}) \cap IRR(R_{j+1})$ can be transformed into a derivation $w \xrightarrow{*}_{R_{j+1}} a$. For doing so we use the fact that R_j and R_{j+1} are reduced, so if (b,a) is a rule, this rule will be the only one containing the letter b.

Let $R_\infty := \{(\ell,b) \mid \exists j \geq 0\ \forall i \geq j: (\ell,b) \in R_i\}$, i.e. R_∞ is the set of persistent rules. Procedure 4.1 can be interpreted as enumerating this system. R_∞ is an infinite monadic system, since deleted rules are never introduced again.

Claim 1: R_∞ is equivalent to R.

Proof: By construction $\xleftrightarrow{*}_R = \xleftrightarrow{*}_{R_j} \supseteq \xleftrightarrow{*}_{R_\infty}$ for all $j \geq 0$. So if $(\ell,b) \in R$ then $\ell \xleftrightarrow{*}_{R_j} b$. If this is not a proof in R_∞, some rule used in the proof is deleted and so there is a strictly smaller proof in some later system. Since this can happen only finitely often, there must be a k such that $\ell \xleftrightarrow{*}_{R_k} b$ and all rules used in this proof are persistent, i.e., $\ell \xleftrightarrow{*}_{R_\infty} b$. ∎

Claim 2: R_∞ is reduced.

Proof: There are only finitely many rules of the form $(b,a) \in R_\infty$ with $b \in \Sigma$ and $a \in \Sigma \cup \{e\}$. Let $k \geq 0$ such that all these rules are in R_k. Since R_k is reduced, there is at most one rule for which the left-hand side is some fixed letter, and this letter does not appear in any other rule. So the right-hand sides of rules in R_∞ are irreducible. Assume that (ℓ_1,b_1) and $(x\ell_1y,b_2)$ are both in R_∞. Then there is an index $j \geq k$ such that both rules are in R_j. However, this contradicts the fact that R_j is reduced. ∎

Claim 3: R_∞ is weakly confluent.

Proof: Let $e \neq w \in \Delta^*_{R_\infty}(RI_{R_\infty}(a)\cdot a) \cap IRR(R_\infty)$. Then there is an index $k \geq 0$, such that $w \in \Delta^*_{R_k}(RI_{R_k}(a)\cdot a) \cap IRR(R_k)$. But then $w \notin IRR(R_{k+1})$, which contradicts our choice of w. Condition (iii) of Theorem 3.5 is verified in a similar way. Now let $(p,q) \in UCP(R_\infty)$. Then there are rules $(\ell_1,b_1),(\ell_2,b_2) \in R_\infty$ such that $\ell_1x = y\ell_2$ for some $x,y \in \Sigma^*$, $0 < |y| < |\ell_1|$, $p = b_1x$, $q = yb_2$ and p and q do not have a common descendant mod R_∞. Notice that these rules can only contain irreducible letters. Since R_∞ only contains the persistent rules, there is an index $j \geq 0$ such that $(\ell_1,b_1),(\ell_2,b_2) \in R_{j+i}$. Hence (p,q) is a critical pair for all R_{j+i}, for all $i \geq 0$. Assume further that R_j contains all $(b,a) \in R_\infty$ with $b \in \Sigma$. Then the pair cannot be resolved mod R_{j+i} for any $i \geq 0$, since any such resolution would involve only rules with R_∞-irreducible letters and so would lead to a resolution in R_∞, i.e. $(p,q) \in UCP(R_{j+i})$. Now let $x \in RI_{R_\infty}(p_1)$ for some $p_1 \in \Delta^*_{R_\infty}(p)$. Only a finite number of rules is involved in the corresponding reductions, and hence there is an index $k \geq j$ such that $p_1 \in \Delta^*_{R_k}(p)$ and $x \in RI_{R_k}(p_1)$. Hence $x \in RI_{R_{k+i}}(p_1)$ for all $i \geq 0$. By Theorem 3.5 we need to verify that $\Delta^*_{R_\infty}(qx) \cap IRR(R_\infty) = \{e\}$, i.e. that e is the only irreducible descendant of qx mod R_∞. Assume to the contrary that $qx \xrightarrow{*}_{R_\infty} y \in IRR(R_\infty)\backslash\{e\}$. As above we conclude that $qx \xrightarrow{*}_{R_\ell} y \in IRR(R_\ell)\backslash\{e\}$ for some $\ell \geq k$, and hence y becomes reducible in $R_{\ell+1}$, since $y \in Sp_1$ for this ℓ. This contradicts the fact that y was irreducible in R_∞ and therewith in all R_i. By symmetry, also the other condition holds and hence, R_∞ is indeed weakly confluent by Theorem 3.5. ∎

This completes the proof of Lemma 4.3. ∎

Thus, on input a finite monadic string-rewriting system R presenting a group, Procedure 4.1 always "computes" a monadic system R_∞ that is reduced, equivalent to R and weakly confluent. Procedure 4.1 terminates iff this system R_∞ is finite. Hence, it remains to characterize the condition under which this system R_∞ is indeed finite.

Theorem 4.4. Let R be a finite monadic string-rewriting system on Σ such that the monoid $\mathfrak{M}_R$ is a group. On input R, Procedure 4.1 terminates if and only if there exists a finite monadic system S on Σ that is equivalent to R and e-confluent.
Proof: We may assume that S is even weakly confluent with the fixed ordering on Σ used in Procedure 4.1 and reduced. If the procedure does not terminate on input R, then it enumerates an infinite monadic system R_∞ that is reduced, weakly confluent and equivalent to R. Let $S = \{(\ell_1,b_1),\ldots,(\ell_m,b_m)\}$. Since R_∞ is equivalent to R and therewith to S, and since R_∞ is weakly confluent, $\ell_i \xrightarrow{*}_{R_\infty} b_i$ for $i = 1,\ldots,m$. Hence, there is an index $k \geq 0$ such that $\ell_i \xrightarrow{*}_{R_k} b_i$ for $i = 1,\ldots,m$, i.e. $\xrightarrow{*}_S \subseteq \xrightarrow{*}_{R_k}$. However, since S is weakly confluent and $[a]_S = [a]_{R_k}$, $a \in \Sigma \cup \{b\}$, R_k is already weakly confluent. Because of Theorem 3.5 this yields that $R'_k = \emptyset$, i.e. on input R Procedure 4.1 terminates after computing R_k. ∎

It can easily be verified, that the system R_∞ is uniquely determined by R and the ordering on Σ, i.e. if S and T are two reduced monadic systems on Σ that are both equivalent to R and that are both weakly confluent, then S and T are in fact identical. This coincides with more general situations for systems that are confluent everywhere (see e.g. [8]).
We close this section by presenting an example to illustrate the way Procedure 4.1 works.

Example 4.5: Let $\Sigma = \{a,b,c\}$ and $R = \{(ab,e),(ba,e),(c^2,e),(cac,b)\}$. $\mathfrak{M}_R$ is a group, R is reduced, $a^{-1} = b$, $b^{-1} = a$, $c^{-1} = c$ and $UCP(R) = \{(ac,cb),(bc,ca)\}$.
The procedure first computes the sets RI(u) for $u \in \Sigma \cup \{ac,cb,bc,ca\}$:
$RI(a) = \{b\}$, $RI(b) = \{a\}$, $RI(c) = \{c,aca\}$
$RI(ac) = \{cb,ac,acab\}$, $RI(cb) = \{ac,aaca\}$
$RI(bc) = \{ca,acaa\}$, $RI(ca) = \{ca,bc,baca\}$.
Now the check $\Delta_R^*(RI(u)\cdot u) \cap IRR(R) = \{e\}$ for $u \in \Sigma$ is done. In the present case this is true, so no rules are introduced by this test. Since aca is irreducible in RI(c), we get the monadic rule (aca,c) as a candidate. Finally, from the test $\Delta_R^*(p\cdot RI(q))$, respectively $\Delta_R^*(q\cdot RI(p))$, for the two unresolvable critical pairs in R we get as rules (acaaca,e),(cbcb,e),(caacaa,e) and (bcbc,e). From these last four rules, two are deleted by the NORMALIZATION and so

$$R_1 = R \cup \{(aca,c),(cbcb,e),(bcbc,e)\}.$$

In the next call of CONTEXT_RESOLVING the inverses stay as they were, but new critical pairs are added, e.g. (bcb,c) and (cbc,a). In fact these rules will be added, since bcb and cbc are irreducible right-inverses of c, respectively b. After the second step and NORMALIZATION we get the system

$$R_2 = \{(ab,e),(ba,e),(c^2,e),(cac,b),(aca,c),(cbc,a),(bcb,c)\}$$

with unresolvable critical pairs $\{(ac,cb),(bc,ca),(cbb,aac),(caa,bbc)\}$.
For this monadic system $RI(a) \cap IRR(R_2) = \{b\}$, $RI(b) \cap IRR(R_2) = \{a\}$ and $RI(c) \cap IRR(R_2) = \{c\}$, so no proper monadic rules are added.

Finally because of
$RI(ac) \cap IRR(R_2) = RI(cb) \cap IRR(R_2) = \{ac,cb\}$, $RI(bc) \cap IRR(R_2) = RI(ca) \cap IRR(R_2) = \{bc,ca\}$,
$RI(cbb) \cap IRR(R_2) = RI(aac) \cap IRR(R_2) = \{cbb,aac,acb\}$ and
$RI(caa) \cap IRR(R_2) = RI(bbc) \cap IRR(R_2) = \{bbc,caa,bca\}$
no further rule is added. The procedure terminates with the weakly confluent system R_2.

5. Concluding Remarks

We have developed a specialized completion procedure for monadic string-rewriting systems presenting groups, based on a polynomial test for confluence on the congruence class of the identity for such systems. The main purpose for such a procedure is to find equivalent presentations which are syntactically restricted and hence provide much more structural and algorithmical information than general presentations. The completion procedure itself can be seen as a kind of unfailing completion, where the role of non-orientable equations is taken by the unresolvable critical pairs, and ground confluence is replaced by confluence on [e]. Thus the divergency of usual completion procedures may be avoided in some cases. A generalization to other classes of systems, e.g. length-reducing ones, seems to be quite hard, since no known decidable criteria for confluence on a single congruence class are known for other classes.
In the subroutine CONTEXT_RESOLVING Procedure 4.1 adds special rules when $ax \xrightarrow{*}_R e$ but $xa \not\xrightarrow{*}_R e$. In this way one tries to make $\langle e \rangle$ closed under cyclic permutations. Here is a possible improvement of the procedure: Whenever an irreducible word $w \in [e]$ is found, add special or monadic rules which guarantee that w and **all** cyclic permutations of it reduce to e. In fact this idea is similar to the one used in the completion procedures of ([4],[10]) and is based on the notion of **symmetrized group presentations** [11].
If we start with a special string-rewriting system R such that the left-hand sides form a symmetrized set (every element is cyclically reduced and the set is closed under cyclic permutations and taking inverses), then $\langle e \rangle_R$ is closed under cyclic permutations. LeChenadec ([10]) presents a process he calls the group symmetrization algorithm that on input a finite symmetrized group presentation $\langle \Sigma;L \rangle$ satisfying certain small cancellation conditions generates the finite length-reducing system S used in Dehn's algorithm to solve the word problem for such groups: Rules of the form $w \rightarrow e$ are split as $w = uv \rightarrow e$, where u is maximal with $u > v^{-1}$, and the rule $u \rightarrow v^{-1}$ is generated. We are doing in fact the same if v^{-1} is a letter.

There are examples where the sets RI(a) or $RI(p_i)$ are indeed infinite. One interesting question is whether the confluence criterion can be specialized to have finite test sets and not just regular ones. This is indeed the case for special systems [19]. The same confluence criterion holds, if one restricts the elements of RI to be irreducible, but these sets still may be infinite in the monadic case.
The examples presented here are fairly simple ones. The reason for this is due to the number and size of the sets $RI_{R_i}(u)$ involved. An implementation of the procedure is currently under way and we hope to gain further insights into how the procedure behaves in practice.

References

[1] J.M. Autebert, L. Boasson, G. Senizergues; Groups and NTS languages; J. Comput. System Sci. 35 (1987), 243-267.

[2] R.V. Book; Decidable sentences of Church-Rosser congruences; Theoretical Computer Science 23 (1983), 301-312.

[3] R.V. Book; Thue systems as rewriting systems; J. Symbolic Computation 3 (1987), 39-68.

[4] H. Bücken; Reduction systems and small cancellation theory; in: Proceedings 4th Workshop on Automated Deduction (1979), 53-59.

[5] R.H. Gilman; Presentations of groups and monoids; J. of Algebra 57 (1979), 544-554.

[6] J.E. Hopcroft, J.D. Ullman; Introduction to Automata Theory, Languages and Computation (Addison-Wesley, Reading, MA, 1979).

[7] M. Jantzen, Confluent String-Rewriting (Springer, Berlin, 1988).

[8] D. Kapur, P. Narendran; The Knuth-Bendix completion procedure and Thue systems; SIAM J. on Computing 14 (1985), 1052-1072.

[9] D. Knuth, P. Bendix; Simple word problems in universal algebras; in: J. Leech (ed.), Computational Problems in Abstract Algebra (Pergamon, New York, 1970), 263-297.

[10] Ph. LeChenadec; Canonical Forms in Finitely Presented Algebras (Pitman: London, Wiley: New York, Toronto, 1986).

[11] R.C. Lyndon, P.E. Schupp; Combinatorial Group Theory (Springer, Berlin, 1977).

[12] K. Madlener, F. Otto; Using string-rewriting for solving the word problem for finitely presented groups; Information Processing Letters 24 (1987), 281-284.

[13] K. Madlener, F. Otto; About the descriptive power of certain classes of finite string-rewriting systems; Theoretical Computer Science 67 (1989), 143-172.

[14] K. Madlener, F. Otto; Decidable sentences for context-free groups, Preprint No., Universität Kaiserslautern, FB Informatik, 1990.

[15] D.E. Muller, P.E. Schupp; Groups the theory of ends, and context-free languages; J. Comput. Systems Ci. 26 (1983), 295-310.

[16] P. Narendran, C. O'Dunlaing, F. Otto; It is undecidable whether a finite special string-rewriting system presents a group: Discrete Math., to appear.

[17] F. Otto; On deciding whether a monoid is a free monoid or is a group; Acta Informatica 23 (1986), 99-110.

[18] F. Otto; On deciding the confluence of a finite string-rewriting system on a given congruence class; J. Comp. Sci. Sciences 35 (1987), 285-310.

[19] F. Otto; The problem of deciding confluence on a given congruence class is tractable for finite special string-rewriting systems; Preprint No. 4/90, FB Math., GhK, Kassel, West Germany, 1990.

[20] F. Otto; Completing a finite special string-rewriting system presenting a group on the congruence class of the empty word; Preprint No. 8/90, FB Math., GhK, Kassel, West Germany, 1990.

[21] C.C. Squier; Word problems and a homological finiteness condition for monoids, J. Pure Appl. Algebra 49 (1987), 201-217.

[22] L. Zhang; The word problem and undecidability results for finitely presented special monoids; submitted for publication.