

# SOME RESULTS ABOUT CONFLUENCE ON A GIVEN CONGRUENCE CLASS

Friedrich Otto

Fachbereich Informatik, Universitaet Kaiserslautern  
6750 Kaiserslautern, Fed. Rep. Germany

currently visiting at

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

## Abstract

It is undecidable in general whether or not a term-rewriting system is confluent on a given congruence class. This result is shown to hold even when the term-rewriting systems under consideration contain unary function symbols only, and all their rules are length-reducing. On the other hand, for certain subclasses of these systems confluence on a given congruence class is decidable.

## 1. Introduction

The **word or equivalence problem** for a term-rewriting system  $R$  on  $T = T(F;V)$  is the following fundamental decision problem :

INSTANCE : Two terms  $t_1, t_2 \in T$ .

QUESTION : Are  $t_1$  and  $t_2$  congruent modulo  $R$  ( $t_1 \leftrightarrow_R^* t_2$ )?

If  $R$  is a finite **canonical** (= Noetherian and confluent) system, then this problem can be solved effectively by reducing  $t_1$  and  $t_2$  to their normal forms  $t_1^0$  and  $t_2^0$ , respectively, and comparing these normal forms. In general it is undecidable whether or not  $R$  is Noetherian [7] and whether or not  $R$  is confluent [1]. However, if  $R$  is known to be Noetherian, then confluence is decidable by computing all critical pairs and by checking whether or not each of them can be resolved [8]. If  $R$  is not confluent, then there exist critical pairs that cannot be resolved. By making these pairs into new rules one can try to generate a canonical system  $R_0$  that is equivalent to  $R$ . This process is called **completion** [11].

If the word problem for  $R$  is undecidable, then the process of completion cannot generate a finite canonical system equivalent to  $R$ . However, even if the word problem is decidable, this process may fail. Actually, there are known examples of finite systems with decidable word problem for which no equivalent finite canonical systems exist [9,10].

However, in some situations it is not necessary to actually solve the word problem, rather it would suffice to solve a restricted version of it. Let  $t$  denote a fixed term. Then the **word problem for  $R$  restricted to  $[t]_R$**  is the following decision problem :

INSTANCE : A term  $t_1 \in T$ .

QUESTION : Is  $t_1$  congruent to  $t$  modulo  $R$  ?

For example, when the factor algebra  $T/\leftrightarrow_R^*$  is a group, then the word problem is equivalent to its restriction to the class of the identity  $[e]_R$ , but there are also other situations in which one is only interested in this restricted problem.

If the rewriting system  $R$  is Noetherian, and if it is confluent on  $[t]_R$ , then the process of rewriting induced by  $R$  solves the restricted word problem. Actually, **Dehn's algorithm** for the word problem, which applies to certain small cancellation groups, is of this form [6,13]. The investigation of Dehn's algorithm from the standpoint of rewriting systems has been initiated by Buecken in his Ph.D. dissertation [5]. Buecken and later LeChenadec [12] proved how certain restrictions upon the rules of  $R$  translate into a proof that  $R$  is confluent on  $[e]_R$ . Their proofs are technically rather involved, although they are tuned to the specific situation of group presentations at hand. Here we want to consider the problem of deciding whether or not a rewriting system is confluent on a given congruence class in a more general setting. However, we shall restrict our attention to term-rewriting systems containing unary function symbols only, which can be interpreted as string-rewriting systems on a finite alphabet  $\Sigma$  [7]. So we are interested in the following decision problem :

CONFLUENCE ON A GIVEN CONGRUENCE CLASS (CCC) :

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

QUESTION : Is  $R$  confluent on  $[w]_R$  ?

After establishing notation we shall derive a characterization theorem that gives necessary and sufficient conditions for a Noetherian rewriting system  $R$  to be confluent on a given congruence class  $[w]_R$  (Section 2). In Section 3 we shall outline a construction that shows that the problem CCC is undecidable in general, even if it is restricted to systems containing length-reducing rules only. Even if the system  $R$  or the word  $w$  is fixed, this problem remains undecidable. However, if we consider monadic systems only, then problem CCC can be solved in double exponential time (Section 4).

For many results presented here proofs are only sketched or even omitted. For complete details see Otto [19].

## 2. A Characterization Theorem

Let  $\Sigma$  be a finite **alphabet**. Then  $\Sigma^*$  denotes the **set of words** over  $\Sigma$ , where  $e$  denotes the **empty word**. As usual  $|w|$  stands for the **length** of the word  $w$ , and superscripts are used to abbreviate words.

A **rewriting system**  $R$  on  $\Sigma$  is a subset of  $\Sigma^* \times \Sigma^*$ , where  $dom(R) = \{l \in \Sigma^* \mid \exists r \in \Sigma^* : (l,r) \in R\}$  and  $range(R) = \{r \in \Sigma^* \mid \exists l \in \Sigma^* : (l,r) \in R\}$ .  $R$  induces the

**reduction relation**  $\rightarrow_R^*$  on  $\Sigma^*$ , which is the reflexive transitive closure of the **single-step reduction relation**  $\rightarrow_R$ , and the **congruence**  $\leftrightarrow_R^*$  on  $\Sigma^*$ , which is the equivalence relation generated by  $\rightarrow_R$ . For  $w \in \Sigma^*$ ,  $[w]_R = \{v \in \Sigma^* \mid w \leftrightarrow_R^* v\}$  is the **congruence class** of  $w$ ,  $\Delta_R^*(w) = \{v \in \Sigma^* \mid w \rightarrow_R^* v\}$  is the **set of descendants** of  $w$ , and  $\langle w \rangle_R = \{v \in \Sigma^* \mid v \rightarrow_R^* w\}$  is the **set of ancestors** of  $w$  modulo  $R$ .

In general, the process of reduction is non-deterministic in that several different rules of  $R$  may be applicable to a given word. However, this process becomes deterministic if we consider left-most reductions.

For each  $l \in \text{dom}(R)$ , let  $r(l)$  be a word such that  $(l, r(l)) \in R$ . Then  $R_1 := \{(l, l(r)) \mid l \in \text{dom}(R)\}$  is a subsystem of  $R$  for which no two different rules have identical left-hand sides. Now a reduction  $u \rightarrow_R v$  is called **left-most** (with respect to  $R_1$ ), if  $u = xly$ ,  $v = xry$ ,  $(l, r) \in R_1$ , and if  $u = x_1l_1y_1$ ,  $v_1 = x_1r_1y_1$ ,  $(l_1, r_1) \in R_1$ , then  $xl$  is a proper prefix of  $x_1l_1$ , or  $xl = x_1l_1$  and  $x$  is a proper prefix of  $x_1$ , or  $x = x_1$ ,  $l = l_1$ , and hence,  $v = v_1$ . We write  $u \rightarrow_{R,L} v$  if  $u \rightarrow_R v$  is left-most, and with  $\rightarrow_{R,L}^*$  we denote the reflexive transitive closure of  $\rightarrow_{R,L}$ .

Obviously, for each reducible word  $u \in \Sigma^*$ , there is a unique word  $v$  such that  $u \rightarrow_{R,L} v$ . Before we can state the characterization theorem we need one further notion.

Two (not necessarily distinct) rules  $(l_1, r_1), (l_2, r_2) \in R$  **overlap** if  $l_1 = xl_2y$  for some  $x, y \in \Sigma^*$  or  $l_1x = yl_2$  for some  $x, y \in \Sigma^*$ ,  $0 < |y| < |l_1|$ . The word  $l_1 = xl_2y$  or  $l_1x = yl_2$  can then be reduced by  $(l_1, r_1)$  and by  $(l_2, r_2)$ , thus giving the **critical pair**  $(r_1, xr_2y)$  or  $(r_1x, yr_2)$ , respectively. We say that a critical pair  $(u, v)$  can be **resolved** if  $\Delta_R^*(u) \cap \Delta_R^*(v) \neq \emptyset$ , otherwise it is called **unresolved**.

It is well-known that the rewriting system  $R$  is locally confluent if and only if all its critical pairs can be resolved, and if  $R$  is Noetherian, then this is also equivalent to  $R$  being confluent. Since the decision problem CCC is of interest only for systems that are non-confluent, the systems we shall be dealing with will usually have some unresolved critical pairs. For a rewriting system  $R$ , let  $UCP(R) = \{(u, v) \mid (u, v) \text{ is a critical pair of } R, \Delta_R^*(u) \cap \Delta_R^*(v) = \emptyset\}$  denote the **set of unresolved critical pairs**. Notice that for a given finite Noetherian rewriting system  $R$  this set can be computed effectively.

Finally, if  $R$  is Noetherian, then for two given words  $u, w \in \Sigma^*$ ,  $L_u(w)$  denotes the set  $L_u(w) = \{x\#y \mid x, y \in \text{IRR}(R), xuy \rightarrow_{R,L}^* w_1\}$ . Here  $w_1 \in \text{IRR}(R)$  such that  $w \rightarrow_{R,L}^* w_1$ , where  $\text{IRR}(R)$  denotes the **set of irreducible words**, and  $\# \notin \Sigma$  is a new letter. Since the process of left-most reduction is deterministic,  $w_1$  is uniquely determined by  $w$ . Now we can state the characterization theorem.

**Theorem 1.** Let  $R$  be a finite Noetherian string-rewriting system on  $\Sigma$ , and let  $w \in \Sigma^*$ . Then the following two statements are equivalent :

- (i) The system  $R$  is confluent on  $[w]_R$ .
- (ii)  $\forall (u, v) \in UCP(R) : L_u(w) = L_v(w)$ .

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

(i)  $\Rightarrow$  (ii) : If  $R$  is confluent on  $[w]_R$ , then for all  $z \in \Sigma^*$ , if  $z \leftrightarrow_R^* w$ , then  $z \rightarrow_{R,L}^* w$ . Let  $(u, v) \in UCP(R)$ . Then  $u \leftrightarrow_R^* v$ . If  $x\#y \in L_u(w)$ , then  $x, y \in IRR(R)$  and  $xuy \rightarrow_{R,L}^* w$ . Hence,  $w \leftrightarrow_R^* xuy \leftrightarrow_R^* xvy$  implying  $xvy \rightarrow_{R,L}^* w$ , i.e.,  $x\#y \in L_v(w)$ . Thus,  $L_u(w) \subseteq L_v(w)$ , and by symmetry this gives  $L_u(w) = L_v(w)$ .

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

**Claim 1.** There exists a word  $z \in \Sigma^*$  such that  $\Delta_R^*(z) \cap IRR(R) \not\supseteq \{w\}$ .

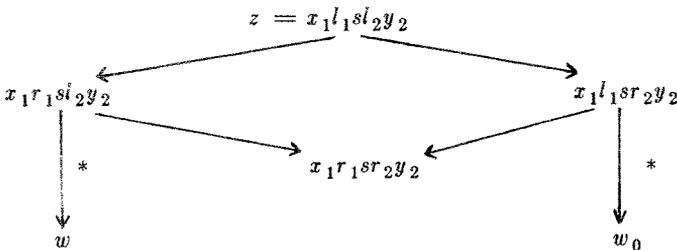
**Proof.** Since  $R$  is not confluent on  $[w]_R$ , there exists a word  $v \in IRR(R)$  such that  $v \neq w$ , but  $v \leftrightarrow_R^* w$ . Hence, there exist an integer  $m > 0$  and words  $w_0, w_1, \dots, w_m \in \Sigma^*$  such that  $w = w_0 \leftrightarrow_R w_1 \leftrightarrow_R \dots \leftrightarrow_R w_m = v$ . Since  $w, v \in IRR(R)$ , we have  $m \geq 2$ ,  $w_1 \rightarrow_R w_0$  and  $w_{m-1} \rightarrow_R w_m$ .

Let  $k := \max\{i \mid w_i \rightarrow_R^* w\}$ . If  $k = m-1$ , then  $\Delta_R^*(w_k) \cap IRR(R) \not\supseteq \{w\}$ . If  $k < m-1$ , then  $w_k \rightarrow_R^* w$ , and  $w_k \rightarrow_R w_{k+1}$ , whereas  $w_{k+1} \not\rightarrow_R^* w$ . Since  $\Delta_R^*(w_{k+1}) \cap IRR(R) \neq \emptyset$ , we thus have  $\Delta_R^*(w_k) \cap IRR(R) \supseteq \{w\} \cup (\Delta_R^*(w_{k+1}) \cap IRR(R)) \not\supseteq \{w\}$ . This proves Claim 1.  $\square$

Since  $R$  is Noetherian, the ordering  $>$  on  $\Sigma^*$  defined by  $u > v$  if and only if  $u \rightarrow_R^+ v$  is well-founded. Let  $z$  be a minimal word with respect to this ordering such that  $\Delta_R^*(z) \cap IRR(R) \not\supseteq \{w\}$ . Then for each word  $z_1$  such that  $z > z_1$ , either  $\Delta_R^*(z_1) \cap IRR(R) = \{w\}$  or  $w \notin \Delta_R^*(z_1)$ .

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

**Proof.** Since  $\Delta_R^*(z) \cap IRR(R) \not\supseteq \{w\}$ ,  $z$  has factorizations of the form given above. Assume that the distinguished occurrences of  $l_1$  and  $l_2$  in  $z$  do not overlap, i.e.,  $z = x_1 l_1 s l_2 y_2$  or  $z = x_2 l_2 s l_1 y_1$ . Without loss of generality we may assume the former. Then we have the following situation :



Since  $z \rightarrow_R x_1 r_1 s l_2 y_2$ ,  $x_1 r_1 s l_2 y_2 < z$ . Hence,  $w \in \Delta_R^*(x_1 r_1 s l_2 y_2)$  implies that  $\Delta_R^*(x_1 r_1 s l_2 y_2) \cap IRR(R) = \{w\}$  due to the minimality of  $z$ . Hence,  $x_1 r_1 s r_2 y_2 \rightarrow_R^* w$ , which in turn gives that  $\Delta_R^*(x_1 l_1 s r_2 y_2) \cap IRR(R) \not\supseteq \{w\}$ , a contradiction. Thus, the distinguished

occurrences of  $l_1$  and  $l_2$  overlap. If the critical pair resulting from this overlap resolves, then we get the same contradiction as before. Hence, the resulting critical pair does not resolve.  $\square$

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

$$\begin{aligned} z &= x_1 l_1 y_1 \rightarrow_{R,L} x_1 r_1 y_1 \rightarrow_{R,L}^* z_1 \in \text{IRR}(R), \text{ and} \\ z &= x_2 l_2 y_2 \rightarrow_R x_2 r_2 y_2 \rightarrow_R^* z_2 \in \text{IRR}(R), \end{aligned}$$

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 unresolved critical pair  $(u, v) \in \text{UCP}(R)$ , i.e.,

$$\begin{aligned} z &= xsy \rightarrow_{R,L} xuy \rightarrow_{R,L}^* z_1 \in \text{IRR}(R), \text{ and} \\ z &= xsy \rightarrow_R xvy \rightarrow_R^* z_2 \in \text{IRR}(R), \end{aligned}$$

where  $z_1 \neq z_2$  and  $w \in \{z_1, z_2\}$ . Note that by Claim 2,  $x, y \in \text{IRR}(R)$ .

- (i) Assume that  $z_1 = w$ . Then  $x\#y \in L_u(w)$ . However,  $w \notin \Delta_R^*(xvy)$  implying that  $x\#y \notin L_v(w)$ .
- (ii) Assume that  $z_1 \neq w$ , i.e.,  $z_2 = w$ . Since  $xvy < z$ , this means that  $\Delta_R^*(xvy) \cap \text{IRR}(R) = \{w\}$ , and hence  $xvy \rightarrow_{R,L}^* z_2 = w$ , i.e.,  $x\#y \in L_v(w)$ . However,  $w \notin \Delta_R^*(xuy)$  implying that  $x\#y \notin L_u(w)$ .

In any case  $L_u(w) \neq L_v(w)$ , thus completing the proof of Theorem 1.  $\square$

### 3. Undecidability Results

We want to show that confluence on a given congruence class is undecidable even if only finite length-reducing systems are being considered. Here a rewriting system  $R$  is called **length-reducing** if  $|l| > |r|$  for each rule  $(l, r) \in R$ . Actually, we shall derive the following result.

**Theorem 2.** There exists a finite length-reducing rewriting system  $R$  on  $\Sigma$  such that the following problem is undecidable:

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

QUESTION: Is  $R$  confluent on  $[w]_R$ ?

The proof of this theorem is based on a construction taken from O'Dunlaing's PhD dissertation ([16], see also [17], Theorem 4.1.1) that has also been used in [14,15,18] for proving various undecidability results. From a single-tape Turing machine  $M = (\Sigma, Q, q_0, \delta)$  accepting a language  $L \subseteq \Sigma^*$  this construction yields a finite length-reducing rewriting system  $R(M)$  on alphabet  $\Gamma \supseteq \Sigma$  such that  $R(M)$  is confluent, and two regular subsets  $\text{CONFIG}, \text{HALTING} \subseteq \Gamma^*$ . The elements of the set  $\text{CONFIG}$  are descriptions of possible configurations, and the elements of the set  $\text{HALTING}$  are descriptions of possible halting

configurations of the Turing machine  $M$ , while the reduction sequences induced by  $R(M)$  correspond to reversed computations of  $M$ . The important property obtained is the following :

For all  $x \in \Sigma^*$ ,  $M$  halts on input  $x$  if and only if there is a word  $w \in HALTING$  such that  $w \rightarrow_{R(M)}^* \$s_0x\&$ . Here  $s_0$  is a special symbol corresponding to the initial state  $q_0$  of  $M$ , and  $\$$  and  $\&$  are special symbols from  $\Gamma-\Sigma$  used as markers.

Thus,  $x \in L$  if and only if  $\langle \$s_0x\& \rangle_{R(M)} \cap HALTING \neq \emptyset$ . Now additional rules can be introduced that are applicable only to descriptions of halting configurations, i.e., to elements of  $HALTING$ . These rules ensure that each element  $w \in HALTING$  has more than one irreducible descendant. Since these rules do not apply to descriptions of non-halting configurations, and since  $\langle \$s_0x\& \rangle_{R(M)} \subseteq CONFIG$  for all  $x \in \Sigma^*$ , this gives the following :

$R(M)$  is confluent on  $[\$s_0x\&]_{R(M)}$  if and only if  $\langle \$s_0x\& \rangle_{R(M)} \cap HALTING = \emptyset$  if and only if  $x \notin L$ .

If  $M$  is a Turing machine accepting a non-recursive language, then the above considerations show that the resulting rewriting system  $R(M)$  meets the requirements of Theorem 2. Thus, the decision problem CCC is undecidable in general, even when it is restricted to a single length-reducing rewriting system.

Instead of dealing with a single system, we can restrict the word  $w$  to always be the empty word, therewith considering the following decision problem:

CONFLUENCE ON CLASS OF THE EMPTY WORD (CCEW) :

INSTANCE : A finite Noetherian rewriting system  $R$  on  $\Sigma$ .

QUESTION : Is  $R$  confluent on  $[e]_R$ ?

For proving this problem to be undecidable, consider a non-recursive language  $L \subseteq \Sigma^*$ . Let  $M$  be a single-tape Turing machine accepting  $L$ , and let  $R(M)$  denote the finite length-reducing system constructed from  $M$  as before. Now for  $x \in \Sigma^*$ , let  $R(x)$  denote the system  $R(M) \cup \{(\$s_0x\&, e)\}$ . Then for each  $x \in \Sigma^*$ ,  $R(x)$  is a finite length-reducing system that can easily be obtained from  $x$ . Further, since the rule  $\$s_0x\& \rightarrow e$  does not introduce any new critical pairs,  $R(x)$  is confluent on  $[e]_{R(x)}$  if and only if  $R(M)$  is confluent on  $[\$s_0x\&]_{R(M)}$ , which is undecidable in general due to the choice of  $R(M)$ . This shows the following result.

**Theorem 3.** The decision problem CCEW remains undecidable even when it is restricted to finite length-reducing rewriting systems.

In particular, this implies that the arguments Buecken and LeChenadec [5,12] give for showing that certain rewriting systems associated with small cancellation groups are confluent on  $[e]$  cannot be generalized to yield an algorithm for this decision problem.

#### 4. Decidability Results

If the rewriting system  $R$  is finite and length-reducing, then for each pair of words  $u, w \in IRR(R)$ , the set  $L_u(w) = \{x\#y \mid x, y \in IRR(R), xuy \xrightarrow{R, L}^* w\}$  is a context-sensitive language. Thus, it is not surprising that it is undecidable in general whether or not  $L_u(w) = L_v(w)$  holds for all  $(u, v) \in UCP(R)$ . If, however, the system  $R$  is finite and monadic, then each set of the form  $L_u(w)$  is a context-free language. Here, a rewriting system  $R$  is called **monadic** if  $R$  is length-reducing and  $range(R) \subseteq \Sigma \cup \{e\}$ . Since the equivalence problem for context-free languages is also undecidable, this observation does not seem to help much. However, as we shall see  $L_u(w)$  is a very restricted kind of context-free language in this situation.

Let  $u, w \in IRR(R)$ , and let  $x\#y \in L_u(w)$ . Then  $x, y \in IRR(R)$ , and  $xuy \xrightarrow{R, L}^* w$ . This reduction sequence consists of two parts : first,  $xu \xrightarrow{R, L}^* x_k \in IRR(R)$ , and then  $x_k y \xrightarrow{R, L}^* w$ . Since  $x$  and  $u$ , respectively  $x_k$  and  $y$ , are irreducible, each step of the above left-most reduction sequences takes place at the border between  $x$  and  $u$ , respectively  $x_k$  and  $y$ . Formally, this fact can be expressed as follows.

**Lemma 1.** Let  $R$  be a finite monadic rewriting system on  $\Sigma$ , let  $x, y \in IRR(R)$ , and let  $w, w_1 \in \Sigma^*$  such that  $xy \xrightarrow{R}^* w \xrightarrow{R} w_1$ . Then there exist words  $z, t \in \Sigma^*$  and a rule  $(l, r) \in R$  such that the following conditions are satisfied :

- (i)  $w = zt$ ,
- (ii)  $w_1 = zrt$ ,
- (iii)  $z$  is a proper prefix of  $x$ , and
- (iv)  $t$  is a proper suffix of  $y$ .

In particular,  $z$  and  $t$  are irreducible.

Thus, a reduction sequence  $xy = w_0 \xrightarrow{R} w_1 \xrightarrow{R} \dots \xrightarrow{R} w_k$ , where  $x, y \in IRR(R)$  and  $R$  is monadic, can be written as  $xy = z_0 l_0 t_0 \xrightarrow{R} z_0 r_0 t_0 = z_1 l_1 t_1 \xrightarrow{R} z_1 r_1 t_1 = z_2 l_2 t_2 \xrightarrow{R} \dots \xrightarrow{R} z_{k-1} r_{k-1} t_{k-1} = w_k$ , where  $(l_i, r_i) \in R$ ,  $z_i$  is a prefix of  $z_{i-1}$ , and  $t_i$  is a suffix of  $t_{i-1}$ . Hence, a pushdown automaton  $A(u, w)$  for recognizing the language  $L_u(w)$  may work as follows.

First, we choose a subsystem  $R_1$  of  $R$  such that  $dom(R_1) = dom(R)$ , and no two different rules of  $R_1$  have identical left-hand sides. In the following we shall only be dealing with this subsystem. Since  $R$  is finite,  $dom(R) = dom(R_1)$  is finite, and so the set  $IRRR(R)\{\#\}IRRR(R)$  is a regular language. Given  $R$  one can easily construct a deterministic finite state acceptor (dfs)  $B$  accepting this language. Running the pushdown automaton  $A(u, w)$  and the dfs  $B$  in parallel, we can determine whether or not the input for  $A(u, w)$  is of the form  $x\#y$  with  $x, y \in IRR(R)$ . So let  $x\#y$ ,  $x, y \in IRR(R)$ , be the given input for  $A(u, w)$ .

Let  $\mu = \max\{|l| \mid l \in \text{dom}(R) \cup \{w\}\}$ , and let  $\lambda = \mu + |u|$ . As **input alphabet** and as **stack alphabet** we take  $\Sigma_0 := \Sigma \cup \{\#\}$ , where  $\#$  also serves as the **start symbol** marking the bottom of the pushdown store.  $A(u, w)$  can store two words  $w_1, w_2 \in \Sigma^*$  of length  $|w_1| \leq \lambda$ ,  $|w_2| \leq \mu$  in its finite control. Thus, at each time the actual configuration of  $A(u, w)$  can be described by a 5-tuple  $(q, w_1, w_2, w_3, w_4)$ , where

- $q$  is from a finite set  $Q$  of **proper states**,
- $w_1 \in \Sigma^*$  such that  $\#w_1$  is the contents of the pushdown store,
- $w_2 \in \Sigma^*$ ,  $|w_2| \leq \lambda$ , is the word stored in the left part of the finite control,
- $w_3 \in \Sigma^*$ ,  $|w_3| \leq \mu$ , is the word stored in the right part of the finite control, and
- $w_4 \in \Sigma_0^*$  is the remaining part of the input, the initial letter of  $w_4$  being the actual input symbol.

The **initial configuration** on input  $x\#y$  is described by  $(q_0, e, u, e, x\#y)$ .  $A(u, w)$  has three stages : READ, REDUCE LEFT, REDUCE RIGHT, which are executed one after another in the given order.

**READ** : The initial part  $x$  of the input is read letter by letter and copied onto the pushdown store, giving the configuration  $(q_1, x, u, e, \#y)$ .

**REDUCE LEFT** : The word  $xu$  is reduced to some irreducible word  $x_1$  by computing a left-most reduction  $xu = w_0 \rightarrow_{R,L} w_1 \rightarrow_{R,L} \dots \rightarrow_{R,L} w_k = x_1$  using the rules of the subsystem  $R_1$ . At step  $i$  of this reduction sequence ( $i=0,1,\dots,k-1$ ) the suffix of length  $\lambda$  of the word  $w_i$  is stored in the left part of the finite control, if  $|w_i| \geq \lambda$ , otherwise, all of  $w_i$  is stored there. By the remarks on reduction sequences in finite monadic systems stated before each reduction step is actually performed on the word stored in the left part of the finite control. Thus, an upper portion 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. Thus, the configuration obtained is of the form  $(q_2, w_1, w_2, e, \#y)$ , where  $w_1w_2 = x_1$ , and  $|w_2| < \lambda$  implies  $w_1 = e$ .

**REDUCE RIGHT** : The input letter  $\#$  is deleted. Then the word  $x_1y$  is reduced to some irreducible word  $x_2$  by computing a left-most reduction  $x_1y = v_0 \rightarrow_{R,L} v_1 \rightarrow_{R,L} \dots \rightarrow_{R,L} v_l = x_2$ . This reduction sequence is computed as follows.  $A(u, w)$  reads the input  $y$  letter by letter. Assume that  $(q, w_1, w_2, y_1, ay_2)$  is the actual configuration at a certain time, where  $|w_2| \leq \lambda$ ,  $|w_2| < \lambda$  implying  $w_1 = e$ ,  $|y_1| \leq \mu$ ,  $w_1w_2y_1 \in \text{IRR}(R)$ , and  $a \in \Sigma$ .

**Case 1.**  $|y_1| < \mu$ , and  $w_2y_1a \in \text{IRR}(R)$ . Then the letter  $a$  is appended to the word  $y_1$ , i.e., we obtain the configuration  $(q, w_1, w_2, y_1a, y_2)$ .

**Case 2.**  $|y_1| = \mu$ , and  $w_2y_1a \in \text{IRR}(R)$ . Then  $A(u, w)$  enters its failure state  $q_f$ , since in this situation  $x_2 = w_1w_2y_1ay_2 \in \text{IRR}(R)$ , but  $x_2 \neq w$ .

**Case 3.**  $w_2y_1a$  is reducible. Then  $w_2y_1a = zl$  for some word  $z \in \Sigma^*$ ,  $|z| \leq |w_2|$ , and some rule  $(l, r) \in R_1$  such that  $w_2y_1a = zl \rightarrow_{R,L} zr$ .  $A(u, w)$  performs this reduction step thus yielding the configuration  $(q, w_1, z, r, y_2)$ . Now the left part of the finite control is refilled

by reading letters from the pushdown store, i.e., we obtain the configuration  $(q, w_3, w_4z, r, y_2)$ , where  $w_1 = w_3w_4$ ,  $|w_4z| \leq \lambda$ , and  $|w_4z| < \lambda$  implies that  $w_3 = e$ .

If  $w_4zr$  is reducible, then another left-most reduction step is applied. This process is repeated until a configuration  $(q, w_5, w_6, r_1, y_2)$  is reached such that  $|w_6| \leq \lambda$ ,  $|w_6| < \lambda$  implies  $w_5 = e$ ,  $r_1 \in \text{range}(R_1)$ , and  $w_6r_1 \in \text{IRR}(R)$ .

$A(u, w)$  accepts if and only if after reading all of its input  $A(u, w)$  is in a configuration of the form  $(q, e, w_1, w_2, e)$ , where  $w_1w_2 = w$ .

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  $w$ . While  $A(u, w)$  is executing stage READ, the length of the contents of the pushdown store is strictly increasing; later on it is non-increasing. Thus,  $A(u, w)$  is a **deterministic one-turn pushdown automaton** (d1pda). As can be checked easily by using the properties of reduction sequences in monadic rewriting systems pointed out earlier,  $A(u, w)$  accepts the language  $L_u(w)$ . Thus,  $L_u(w)$  is a **deterministic one-turn context-free language**.

Let  $R$  be a finite monadic rewriting system on  $\Sigma$ , and let  $w \in \Sigma^*$ . From  $R$  we can effectively compute a finite subsystem  $R_1$  such that  $\text{dom}(R) = \text{dom}(R_1)$  and no two different rules of  $R_1$  have the same left-hand side, the finite set  $UCP(R)$  of unresolved critical pairs of  $R$ , and an irreducible descendant  $w_1$  of  $w$  such that  $w \rightarrow_{R,L}^* w_1$ . For each pair  $(u, v) \in UCP(R)$ , d1pdas  $A(u, w_1)$  and  $A(v, w_1)$  can be determined effectively such that  $A(u, w_1)$  recognizes the language  $L_u(w)$ , while  $A(v, w_1)$  recognizes the language  $L_v(w)$ . Since the equivalence problem for deterministic finite-turn pdas is decidable [20], this gives the following result.

**Theorem 4.** The problem CCC is decidable when it is restricted to finite monadic rewriting systems.

From  $R$  the set  $UCP(R)$  can be constructed in polynomial time, as can the subsystem  $R_1$ . Given  $w$  the word  $w_1 \in \text{IRR}(R)$  such that  $w \rightarrow_{R,L}^* w_1$  can also be determined in polynomial time [3]. However, for constructing the d1pda  $A(u, w_1)$  from  $u$ ,  $w_1$ , and  $R_1$ , in general exponential time will be needed, since  $A(u, w_1)$  may be of exponential size. It takes double exponential time to decide the equivalence of two deterministic finite-turn pdas [2]. However, due to their specific form the equivalence of  $A(u, w_1)$  and  $A(v, w_1)$  is decidable in exponential time (cf. the proof of Theorem 5.2 of [2]). Thus, our solution for the decision problem CCC restricted to monadic systems takes double exponential time.

For versions of the problem CCC that are even further restricted algorithms with lower time bounds have been obtained. If the rewriting systems considered are finite and **special**, i.e., length-reducing with  $\text{range}(R) = \{e\}$ , then the problem CCEW is decidable in polynomial time, due to the fact that for this particular problem a simplified version of the characterization theorem exists. When restricted to a fixed finite special rewriting system the problem CCC becomes decidable in exponential time. Finally, we would like to mention the following

interesting result.

**Theorem 5.** Let  $R = \{(l, e)\}$  be a special one-rule rewriting system on  $\Sigma$ . Then the following statements are equivalent :

- (i)  $R$  is confluent.
- (ii)  $R$  is confluent on  $[w]_R$  for some word  $w \in \Sigma^*$ .
- (iii) The root  $\rho(l)$  of the word  $l$  has no overlap.

The equivalence of statements (i) and (iii) is a result of Book [4]. The new aspect is that if a special one-rule system is confluent on any congruence class, then it already is a confluent system.

So far we have only been dealing with certain restricted classes of string-rewriting systems, i.e., term-rewriting systems containing unary function symbols only. For future research it might be promising to look for other types of term-rewriting systems for which the problem CCC becomes decidable.

**Acknowledgement.** The author wants to thank Ron Book for bringing this problem to his attention, and for many fruitful discussions regarding the various results presented in this paper.

### References

1. G. Bauer, F. Otto; Finite complete rewriting systems and the complexity of the word problem; *Acta Informatica* 21 (1984), 521-540.
2. C. Beeri; An improvement on Valiant's decision procedure for equivalence of deterministic finite turn pushdown machines; *Theoret. Comput. Sci.* 3 (1976), 305-320.
3. R.V. Book; Confluent and other types of Thue systems; *Journal ACM* 29 (1982), 171-182.
4. R.V. Book; A note on special Thue systems with a single defining relation; *Math. Systems Theory* 16 (1983), 57-60.
5. H. Buecken; Anwendung von Reduktionssystemen auf das Wortproblem in der Gruppentheorie; dissertation, Aachen, 1979.
6. B. Domanski, M. Anshel; The complexity of Dehn's algorithm for word problems in groups; *Journal Algorithms* 6 (1985), 543-549.
7. G. Huet, D.S. Lankford; On the uniform halting problem for term rewriting systems ; *Lab. Rep. No.283, INRIA, Le Chesnay, France, 1978.*
8. G. Huet; Confluent reductions : Abstract properties and applications to term rewriting systems; *Journal ACM* 27 (1980), 797-821.
9. M. Jantzen; A note on a special one-rule semi-Thue system; *Inform. Proc. Letters* 21 (1985), 135-140.
10. D. Kapur, P. Narendran; A finite Thue system with decidable word problem and without equivalent finite canonical system; *Theoret. Comput. Sci.* 35 (1985), 337-344.
11. D. Knuth, P. Bendix; Simple word problems in universal algebras; in: J.Leech (ed.), *Computational Problems in Abstract Algebra*, Pergamon Press, 1970, 263-297.
12. P. LeChenadec; Canonical Forms in Finitely Presented Algebras, Pitman Publ. Limited, John Wiley & Sons, New York, 1986.
13. R.E. Lyndon, P.E. Schupp; *Combinatorial Group Theory*, Springer, 1977.
14. P. Narendran, C. O'Dunlaing; Cancellativity in finitely presented semigroups; *Journal Symbolic Computation*, to appear.
15. P. Narendran, C. O'Dunlaing, H. Rolletschek; Complexity of certain decision problems about

- congruential languages; *Journal Comp. System Sciences*, 30 (1985), 343-358.
16. C. O'Dunlaing; Finite and infinite regular Thue systems, PhD dissertation, Dept. of Math., University of California, Santa Barbara, 1981.
  17. C. O'Dunlaing; Infinite regular Thue systems; *Theoret. Comput. Sci.* 25 (1983), 171-192.
  18. F. Otto; Some undecidability results for non-monadic Church-Rosser Thue systems; *Theoret. Comput. Sci.* 33 (1984), 261-278.
  19. F. Otto; On deciding the confluence of a finite string-rewriting system on a given congruence class; Technical Report 159/86, Fachbereich Informatik, Universitaet Kaiserslautern, 1986; also submitted for publication.
  20. L.G. Valiant; The equivalence problem for deterministic finite-turn pushdown automata; *Inf. and Control* 25 (1974), 123-133.