Universität Stuttgart
Fakultät Informatik

Institut für Informatik
Breitwiesenstraße 20-22
D-70565 Stuttgart

Complexity Results for Confluence Problems

Markus Lohrey

Report Nr. 1999/05

March 4, 1999

Abstract

We study the complexity of the confluence problem for restricted kinds of semi–Thue systems, vector replacement systems and general trace rewriting systems. We prove that confluence for length–reducing semi–Thue systems is P–complete and that this complexity reduces to AC^1 in the monadic case (where all right–hand sides consist of at most one symbol). For length–reducing vector replacement systems we prove that the confluence problem is PSPACE–complete and that the complexity reduces to NP and P, respectively, for monadic vector replacement systems and special vector replacement systems (where all right–hand sides are empty), respectively. Finally we prove that for special trace rewriting systems, confluence can be decided in polynomial time and that the extended word problem for special trace rewriting systems is undecidable.

1 Introduction

Rewriting systems that operate on different kinds of objects have received a lot of attention in computer science and mathematics. For all kinds of rewriting systems, confluence and termination are two of the most interesting properties. Together they guarantee the existence of unique normalforms.

Two of the most intensively studied types of rewriting systems are rewriting systems on free monoids, which are better known as semi–Thue systems [BO93], and rewriting systems on free commutative monoid, which are better known as vector replacement systems or Petri nets. Both of these types of rewriting systems may be seen as special cases of trace rewriting systems [Die90b]. Trace rewriting systems operate on free partially commutative monoids, which are in computer science better known as trace monoids. Trace monoids were introduced by [Maz77] into computer science as a model of concurrent systems.

Several decidability and undecidability results are known for the confluence problem for the different types of rewriting systems mentioned above. Let us just mention a few of these results. It is known that for length–reducing semi–Thue systems confluence can be decided in polynomial time [BO81, KKMN85]. In contrast to this result there exists a trace monoid such that confluence is undecidable for length–reducing trace rewriting systems over this trace monoid [NO88]. In [Loh98] this result was even sharpened. It was shown that unless the underlying trace monoid is free or free commutative, confluence is undecidable for length–reducing trace rewriting systems. Concerning vector replacement systems it was shown in [VRL98] that confluence is decidable but EXSPACE–hard for the class of all vector replacement systems.

In this paper we continue the investigation of the confluence problem for restricted kinds of trace rewriting systems. In Section 3 we prove that the confluence problem for length–reducing semi–Thue systems is not only solvable in polynomial time but furthermore P–complete, which roughly means that it is inherently sequential. On the other hand we prove that for the more restricted class of monadic semi–Thue systems (where monadic means that all right–hand sides consist of at most one symbol) there exists an efficient parallel algorithm (more precisely an AC^1–algorithm) that decides confluence. Concerning vector replacement systems we prove in Section 4 that for the length–reducing case, confluence is PSPACE–complete and that this complexity reduces for the monadic case and the special case (where special means that all right–hand sides are empty), respectively, to NP and P, respectively. Finally in Section 5 we prove that confluence is decidable for special trace rewriting systems in polynomial time which solves a question from [Die90b]. We end this paper by showing that in contrast to semi–Thue systems the extended word problem [BO85] is undecidable even for special trace rewriting systems that contain only one rule.

2 Preliminaries

In this section we will introduce some notations that we will use in this paper. Given an alphabet Σ, Σ^* denotes the set of all finite words of elements of Σ. The empty word is denoted by 1. As usual $\Sigma^+ = \Sigma^* \backslash \{1\}$. The number of occurrences of $a \in \Sigma$ in the word s is denoted by $|s|_a$. The length of the word s is denoted by $|s|$. The set $\{s \in \Sigma^* \mid |s| = n\}$ is denoted by Σ^n. The set of all letters that occur in the word s is denoted by $\mathit{alph}(s)$. The word s written in reverse order is denoted by s^{rev}. A *context–free grammar* G, briefly CFG, is a tuple $G = (N, \Sigma, P, S)$ where N and Σ are disjoint alphabets of *non–terminal symbols* and *terminal symbols*, respectively, P is the set of *productions* which is a finite subset of $N \times (N \cup \Sigma)^*$, and $S \in N$ is the start symbol. A production (A, α) is usually written as $A \Rightarrow \alpha$. Given $A \in N$, we denote by $L(G, A)$ the set of all elements in Σ^* that can be derived by G, starting from A. Finally $L(G) = L(G, S)$. We say that G is 1–*free* if for every production $(A \Rightarrow \alpha) \in P$ it holds $\alpha \neq 1$. Let $a_1, a_2, \ldots, a_n$ be a fixed linear ordering of the alphabet Σ. A *commutative word* over Σ is a word of the form $a_1^{e_1} a_2^{e_2} \cdots a_n^{e_n}$, where $e_1, \ldots, e_n \in \mathbb{N}$. In particular $1 = a_1^0 \cdots a_n^0$ and $|a_1^{e_1} \cdots a_n^{e_n}| = e_1 + \cdots + e_n$. The set of all commutative words over Σ is denoted by $\Sigma^\oplus$. The concatenation of two commutative words $a_1^{d_1} \cdots a_n^{d_n}$ and $a_1^{e_1} \cdots a_n^{e_n}$ is $a_1^{d_1+e_1} \cdots a_n^{d_n+e_n}$. In this way $\Sigma^\oplus$ becomes isomorphic to the *free commutative monoid* $\mathbb{N}^{|\Sigma|}$. For a natural number $n \in \mathbb{N}$ let $\mathit{ld}(n)$ denote the logarithm of n to the base 2. Furthermore let $\mathit{bit}(n) = \lfloor \mathit{ld}(n) \rfloor + 1$ if $n > 0$ and $\mathit{bit}(0) = 1$, i.e., $\mathit{bit}(n)$ is the length of the binary representation of n. For $a_1^{e_1} \cdots a_n^{e_n} \in \Sigma^\oplus$ we define $\mathit{bit}(s) = \sum_{i=1}^{n} \mathit{bit}(e_i)$.

We assume that the reader is familiar with the basic notions of complexity theory, in particular with the complexity classes P, NP, and PSPACE, see for instance [Pap94]. In the following we introduce some notions concerning circuit complexity, see [BS90] for more details. All Boolean circuit that we consider in this paper are built from AND, OR, and NOT–gates. A Boolean circuit with n (linearly ordered) input nodes and exactly one output node accepts a language $L \subseteq \{a, b\}^n$ in the obvious way. By encoding an arbitrary alphabet with more than two symbols into the alphabet $\{a, b\}$, a Boolean circuit can also accept a language $L \subseteq \Sigma^n$, where Σ is an arbitrary alphabet. Let $\mathcal{F} = \{C_n \mid n \geq 0\}$ be a family of Boolean circuits, where C_n accepts a subset of Σ^n. We say that $\mathcal{F}$ is *uniform* if the function $n \mapsto C_n$ can be computed in deterministic logarithmic space. Note that this implies that there exists a polynomial $p(n)$ such that C_n contains at most $p(n)$ many gates. For $k \geq 0$, AC^k denotes the set of all languages L such that there exists a uniform family $\{C_n \mid n \geq 0\}$ of Boolean circuits such that the following holds: There exists a constant $c > 0$ such that the depth of the circuit C_n (i.e. the length of the longest path from an input node to the unique output node) is at most $c \cdot \mathit{ld}^k(n)$, the fan–in (i.e. the number of inputs) of each AND and OR–gate is unbounded (but since the number of gates is bounded by a polynomial the same can be assumed for the fan–in), and finally C_n accepts the language $\Sigma^n \cap L$. The class NC^k is defined in the same way, but only gates of fan–in at most two are allowed. Finally $\mathrm{NC} = \bigcup_{k \geq 0} \mathrm{NC}^k$. The problems in NC are viewed as those problems that can be efficiently parallelized. By allowing circuits with more than one output node it is possible to define classes of functions that correspond to AC^k, NC^k, and NC. We omit the obvious formal definition, and just say that a particular function can be calculated for instance in AC^k. In particular we use reduction functions between instances of computational problems that can be calculated in AC^0 and which are therefore also computable in deterministic logarithmic space by well known results, see e.g. [Pap94], Theorem 16.1. Since an AND–gate of fan–in m can be replaced by a tree of height $ld(m)$ which consists of AND–gates of fan–in two, and similarly for OR–gates, the following hierarchie is obvious.

$$\mathrm{NC}^0 \subseteq \mathrm{AC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{AC}^1 \subseteq \mathrm{NC}^2 \subseteq \cdots \subseteq \mathrm{NC} \subseteq P$$

For most of these inclusions it is unknown whether they are proper. In particular it is an open problem whether NC = P. It is generally believed that NC $\subsetneq$ P. If this is true then problems that are P–complete under NC–reductions cannot be contained in NC, i.e., are inherently sequential.

In the following we introduce some notions from trace theory, see [DR95] for more details. An *independence alphabet* is an undirected graph (Σ, I), where Σ is a finite alphabet and $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric relation, called an *independence relation*. Given an independence alphabet (Σ, I) we define the *trace monoid* $\mathbb{M}(\Sigma, I)$ as the quotient monoid $\Sigma^*/\equiv_I$, where $\equiv_I$ denotes the least equivalence relation that contains all pairs of the form $(sabt, sbat)$ for $(a, b) \in I$ and $s, t \in \Sigma^*$, which is a congruence on Σ^*. An element of $\mathbb{M}(\Sigma, I)$, i.e., an equivalence class of words, is called a *trace*. The trace that contains the word s is denoted by $[s]_I$. The neutral element of $\mathbb{M}(\Sigma, I)$ is the empty trace $[1]_I$ which will be also denoted by 1. Concatenation of traces is defined by $[s]_I[t]_I = [st]_I$. Since for all words $s, t \in \Sigma^*$, $s \equiv_I t$ implies $|s| = |t|$ and $alph(s) = alph(t)$, we can define $|[s]_I| = |s|$ and $alph([s]_I) = alph(s)$. The independence relation I can be lifted to $\mathbb{M}(\Sigma, I)$ by $u \; I \; v$ if $alph(u) \times alph(v) \subseteq I$. For the rest of this section let (Σ, I) be an independence alphabet and let $M = \mathbb{M}(\Sigma, I)$. If $I = (\Sigma \times \Sigma) \backslash Id_\Sigma$, where $Id_\Sigma = \{(a, a) \mid a \in \Sigma\}$, then M is isomorphic to the free commutative monoid $\mathbb{N}^{|\Sigma|} \simeq \Sigma^{\oplus}$. On the other hand if $I = \emptyset$ then M is isomorphic to the free monoid Σ^*. The following lemma is a generalization of Levi's lemma for traces [CP85], which states that for $u_1, u_2, v_1, v_2 \in M$ it holds $u_1 u_2 = v_1 v_2$ if and only if there exist $w_{i,j} \in M$ $(1 \leq i, j \leq 2)$ such that $u_i = w_{i,1} w_{i,2}$, $v_i = w_{1,i} w_{2,i}$ $(1 \leq i \leq 2)$ and $w_{1,2} \, I \, w_{2,1}$. The following lemma can be proved by induction on $n + m$ using Levi's lemma for the case $n = 2 = m$.

Lemma 2.1. Let $u_1, \ldots, u_m, v_1, \ldots, v_n \in M$. Then it holds

$$u_1 u_2 \ldots u_m = v_1 v_2 \ldots v_n$$

if and only if there exist $w_{i,j} \in M$ $(1 \leq i \leq m,\ 1 \leq j \leq n)$ such that

- $u_i = w_{i,1} w_{i,2} \ldots w_{i,n}$ for every $1 \leq i \leq m$,
- $v_j = w_{1,j} w_{2,j} \ldots w_{m,j}$ for every $1 \leq j \leq n$, and
- $w_{i,j} \, I \, w_{k,l}$ if $1 \leq i < k \leq m$ and $1 \leq l < j \leq n$.

The situation in the lemma can be visualized by the diagram below, where $n = m = 5$. The i–th column corresponds to u_i, the j–th row corresponds to v_j and the intersection of the i–th column and the j–th row represents $w_{i,j}$. Furthermore $w_{i,j}$ and $w_{k,l}$ are independent if one of them is north–west of the other one.

v_5	$w_{1,5}$	$w_{2,5}$	$w_{3,5}$	$w_{4,5}$	$w_{5,5}$
v_4	$w_{1,4}$	$w_{2,4}$	$w_{3,4}$	$w_{4,4}$	$w_{5,4}$
v_3	$w_{1,3}$	$w_{2,3}$	$w_{3,3}$	$w_{4,3}$	$w_{5,3}$
v_2	$w_{1,2}$	$w_{2,2}$	$w_{3,2}$	$w_{4,2}$	$w_{5,2}$
v_1	$w_{1,1}$	$w_{2,1}$	$w_{3,1}$	$w_{4,1}$	$w_{5,1}$
	u_1	u_2	u_3	u_4	u_5

Proof. We use induction on $m + n$. The case $m = 1$ or $n = 1$ is trivial. Thus let $m > 1$ and $n > 1$. Levi's lemma applied to the identity $(u_1 \cdots u_{m-1})u_m = (v_1 \cdots v_{n-1})v_n$ gives four traces x, u, v and $w_{m,n}$ such that

$$u_1 u_2 \cdots u_{m-1} = xv, \quad v_1 v_2 \cdots v_{n-1} = xu, \quad u_m = uw_{m,n}, \quad v_n = vw_{m,n}, \quad u\, I\, v.$$

Next we apply the induction hypothesis to the identity $u_1 u_2 \cdots u_{m-1} = xv$. We obtain traces $y_1, y_2, \ldots, y_{m-1}$ and $w_{1,n}, w_{2,n}, \ldots w_{m-1,n}$ such that

$$\begin{aligned} &x = y_1 y_2 \cdots y_{m-1}, && v = w_{1,n} w_{2,n} \cdots w_{m-1,n}, \\ &u_i = y_i w_{i,n} \ (1 \le i \le m-1), && y_k\, I\, w_{i,n} \text{ if } 1 \le i < k \le m-1. \end{aligned}$$

Similarly, by the induction hypothesis applied to the identity $v_1 v_2 \cdots v_{n-1} = xu$ there exist traces $z_1, z_2, \ldots, z_{n-1}$ and $w_{m,1}, w_{m,2}, \ldots w_{m,n-1}$ such that

$$\begin{aligned} &x = z_1 z_2 \cdots z_{n-1}, && u = w_{m,1} w_{m,2} \cdots w_{m,n-1}, \\ &v_j = z_j w_{m,j} \ (1 \le j \le n-1), && z_i\, I\, w_{m,j} \text{ if } 1 \le j < i \le n-1. \end{aligned}$$

Thus $y_1 y_2 \cdots y_{m-1} = x = z_1 z_2 \cdots z_{n-1}$. The induction hypothesis applied to this identity gives traces $w_{i,j}$ $(1 \le i \le m-1,\, 1 \le j \le n-1)$ such that

- $y_i = w_{i,1} w_{i,2} \ldots w_{i,n-1}$ for every $1 \le i \le m-1$,
- $z_j = w_{1,j} w_{2,j} \ldots w_{m-1,j}$ for every $1 \le j \le n-1$, and
- $w_{i,j}\, I\, w_{k,l}$ if $1 \le i < k \le m-1$ and $1 \le l < j \le n-1$.

Altogether we now obtain the following:

- $u_i = y_i w_{i,n} = w_{i,1} w_{i,2} \ldots w_{i,n-1} w_{i,n}$ for every $1 \le i \le m-1$.
- $u_m = uw_{m,n} = w_{m,1} w_{m,2} \cdots w_{m,n-1} w_{m,n}$
- $v_j = z_j w_{m,j} = w_{1,j} w_{2,j} \ldots w_{m-1,j} w_{m,j}$ for every $1 \le j \le n-1$.
- $v_n = vw_{m,n} = w_{1,n} w_{2,n} \cdots w_{m-1,n}, w_{m,n}$

Finally we have to verify that $w_{i,j}\, I\, w_{k,l}$ if $1 \leq i < k \leq m$ and $1 \leq l < j \leq n$. For the case $k < m$ and $j < n$ this was already stated above.

- $1 \leq i < k \leq m-1$ and $1 \leq l < n$: Since $y_k\, I\, w_{i,n}$ and $w_{k,l}$ is a factor of y_k it holds $w_{i,n}\, I\, w_{k,l}$.
- $1 \leq i < m$ and $1 \leq l < j < n$: It holds $z_j\, I\, w_{m,l}$. Since $w_{i,j}$ is a factor of z_j it holds $w_{i,j}\, I\, w_{m,l}$.
- $1 \leq i < m$ and $1 \leq l < n$: Then $w_{i,n}$ is a factor of v and $w_{m,l}$ is a factor of u. Since $u\, I\, v$ we have $w_{i,n}\, I\, w_{m,l}$.

Now we have covered all possibilities. $\square$

A *trace rewriting system*, briefly TRS, over the trace monoid M is a non–empty finite subset of $M \times M$. In the rest of this section let $\mathcal{R}$ be an arbitrary TRS over the trace monoid $M = \mathbb{M}(\Sigma, I)$. If $I = \emptyset$, i.e., $M \simeq \Sigma^*$, then $\mathcal{R}$ is also called a *semi–Thue system*, briefly STS, over Σ^* (see [BO93] for a detailed introduction into the theory of semi–Thue systems). On the other hand if $I = (\Sigma \times \Sigma) \backslash Id_\Sigma$, i.e., $M \simeq \mathbb{N}^{|\Sigma|}$, then $\mathcal{R}$ is also called a *vector replacement system*, briefly VRS, over $\Sigma^\oplus$ (or a VRS in the dimension $|\Sigma|$). An element $(l, r) \in \mathcal{R}$ is also denoted by $l \to r$. The set $\{l \mid \exists r \in M : (l, r) \in \mathcal{R}\}$ of all left–hand sides of $\mathcal{R}$ is denoted by $dom(\mathcal{R})$. The set $\{r \mid \exists l \in M : (l, r) \in \mathcal{R}\}$ of all right–hand sides of $\mathcal{R}$ is denoted by $ran(\mathcal{R})$. Given $c = (l, r) \in \mathcal{R}$ and $s, t \in M$, we write $s \to_c t$ if $s = ulv$ and $t = urv$ for some $u, v \in M$. We write $s \to_\mathcal{R} t$ if there exists a $c \in \mathcal{R}$ with $s \to_c t$. The transitive (reflexive and transitive) closure of $\to_\mathcal{R}$ is denoted by $\to_\mathcal{R}^+$ ($\to_\mathcal{R}^*$). The transitive, reflexive and symmetric closure of $\to_\mathcal{R}$ is denoted by $\leftrightarrow_\mathcal{R}^*$. It is a congruence relation on M. We say that $\mathcal{R}$ is *terminating* if there does not exist an infinite chain $u_1 \to_\mathcal{R} u_2 \to_\mathcal{R} u_3 \to_\mathcal{R} \cdots$ in M. We say that a pair $(u, v) \in M \times M$ is *confluent* (with respect to $\mathcal{R}$) if there exists a $w \in M$ such that $u \to_\mathcal{R}^* w$ and $v \to_\mathcal{R}^* w$. We say that $\mathcal{R}$ is *confluent on* the trace $u \in M$ if for all $v_1, v_2 \in M$ with $u \to_\mathcal{R}^* v_1$ and $u \to_\mathcal{R}^* v_2$ there exists a $w \in M$ with $v_1 \to_\mathcal{R}^* w$ and $v_2 \to_\mathcal{R}^* w$. We say that $\mathcal{R}$ is *confluent* if $\mathcal{R}$ is confluent on all $u \in M$. We say that $\mathcal{R}$ is *locally confluent* if for all $u, v_1, v_2 \in M$ with $u \to_\mathcal{R} v_1$ and $u \to_\mathcal{R} v_2$ there exists a $w \in M$ with $v_1 \to_\mathcal{R}^* w$ and $v_2 \to_\mathcal{R}^* w$. If $\mathcal{R}$ is terminating then by Newman's lemma [New43] $\mathcal{R}$ is confluent if and only if $\mathcal{R}$ is locally confluent. A trace u is *irreducible* (with respect to $\mathcal{R}$) if there does not exist a $v \in M$ with $u \to_\mathcal{R} v$. The set of all traces in M that are irreducible with respect to $\mathcal{R}$ is denoted by $IRR(\mathcal{R})$. The trace v is a *normalform* of u if $u \to_\mathcal{R}^* v$ and $v \in IRR(\mathcal{R})$. We say that $\mathcal{R}$ is *length–reducing* if $|l| > |r|$ for every $(l, r) \in \mathcal{R}$. Obviously, if $\mathcal{R}$ is length reducing then $\mathcal{R}$ is also terminating. We say that $\mathcal{R}$ is *monadic* if $\mathcal{R}$ is length–reducing and $ran(\mathcal{R}) \subseteq \{1\} \cup \Sigma$. We say that $\mathcal{R}$ is *special* if $ran(\mathcal{R}) = \{1\}$ and $1 \notin dom(\mathcal{R})$. Let COLR($M$) (COMO($M$), COSP($M$)) denote the set of all confluent TRSs over M that are length–reducing (monadic, special). The *uniform word problem* for a class $\mathcal{C}$ of TRSs over M is the following decision problem: Given a $\mathcal{R} \in \mathcal{C}$ and two traces $u, v \in M$, does $u \leftrightarrow_\mathcal{R}^* v$ hold?

Since we will investigate the complexity of algorithms that take a TRS as input, we have to define the length $\|\mathcal{R}\|$ of the TRS $\mathcal{R}$. First assume that $I \neq (\Sigma \times \Sigma)\backslash Id_\Sigma$. In this case in general the best possible coding of a rule from $\mathcal{R}$ is to simply write down words over Σ that represent the left- and right–hand side of the rule. Thus we define $\|\mathcal{R}\| = \sum\{|l|+|r| \mid (l,r) \in \mathcal{R}\}$. But if $I = (\Sigma \times \Sigma)\backslash Id_\Sigma$, i.e., if $\mathcal{R}$ is a VRS over $\Sigma^\oplus$ we can code $\mathcal{R}$ more efficiently by using the binary notation. Therefore in this case we define $\|\mathcal{R}\| = \sum\{\mathit{bit}(l) + \mathit{bit}(r) \mid (l,r) \in \mathcal{R}\}$. In this paper we always assume that a TRS $\mathcal{R}$ is represented as a string of length $\Omega(\|\mathcal{R}\|)$ (since the different rules of $\mathcal{R}$ must be separated by special markers, we use the Ω–notation).

3 Semi–Thue systems

For terminating STSs confluence is known to be decidable [BO81]. This classical result is based on the so called *critical pairs* of a STS. Let $\mathcal{R}$ be a STS over Σ^*. The set of *critical pairs* $CP(\mathcal{R})$ is the set

$$\begin{aligned} CP(\mathcal{R}) = &\{(sr_1t, r_2) \mid (l_1, r_1), (sl_1t, r_2) \in \mathcal{R}\} \cup \\ &\{(r_1u, sr_2) \mid (st, r_1), (tu, r_2) \in \mathcal{R}, t \neq 1\}. \end{aligned}$$

Note that $CP(\mathcal{R})$ is finite. It is well known that $\mathcal{R}$ is locally confluent if and only if all critical pairs are confluent [NB72]. Since the last property can be decided effectively for the class of terminating STSs, confluence is decidable for this class. For length–reducing STSs, confluence can be even decided in polynomial time [BO81]. To the knowledge of the author, the best known algorithm for deciding COLR(Σ^*) is the $O(\|\mathcal{R}\|^3)$ algorithm from [KKMN85]. In this section we prove that COLR(Σ^*) is moreover P–complete if $|\Sigma| \geq 2$. Thus, COLR(Σ^*) seems to be inherently sequential. But this situation changes for the monadic case. At the end of this section we show that COMO(Σ^*) is contained in AC^1.

In order to prove that COLR(Σ^*) is P–complete if $|\Sigma| \geq 2$ we will first prove that the confluence problem is P–complete for the class of all STSs (without restriction on the cardinality of the underlying finite alphabet). Afterwards we will use the following lemma.

Lemma 3.1. Let $k > 2$ and $\Sigma = \{a_1, \ldots, a_k\}$. Let $\mathcal{R}$ be a length–reducing STS over Σ. Let the injective morphism $\phi : \Sigma^* \to \{a, b\}^*$ be defined by $\phi(a_i) = aba^{i+1}b^{k-i+2}$ for $i \in \{1, \ldots, k\}$ and let $\phi(\mathcal{R}) = \{(\phi(l), \phi(r)) \mid (l,r) \in \mathcal{R}\}$. Then

(1) $\phi(\mathcal{R})$ is length–reducing and can be calculated from $\mathcal{R}$ in AC^0.

(2) If $\phi(s) \to_{\phi(\mathcal{R})} u$ then $u = \phi(t)$ and $s \to_{\mathcal{R}} t$ for some $t \in \Sigma$.

(3) $\phi(s) \to_{\phi(\mathcal{R})} \phi(t)$ if and only if $s \to_{\mathcal{R}} t$.

(4) $\mathcal{R}$ is confluent if and only if $\phi(\mathcal{R})$ is confluent.

Proof. The first statement of the lemma is obvious (note that $|\phi(a_i)| = k+5$ for all $i \in \{1, \ldots, k\}$). The second statement follows from the following statement, where s and t are non–empty words:

$$\text{If } \phi(s) = u_1\phi(t)u_2 \text{ then } u_1 = \phi(v_1) \text{ and } u_2 = \phi(v_2) \text{ for some } v_1, v_2 \in \Sigma^*. \quad (1)$$

Note that this statement does not hold for $t = 1$. Since we need (1) only for the case that $t \in dom(\mathcal{R})$ and since $\mathcal{R}$ is length–reducing, the restriction $t \neq 1$ does not matter. The if–direction from the third statement is obvious and the only if–direction follows from the second statement of the lemma and the injectivity of ϕ. The if–direction of the fourth statement follows from the second statement of the lemma as follows. Let $\phi(\mathcal{R})$ be confluent and let $s, t, u \in \Sigma^*$ such that $s \to_{\mathcal{R}} t$ and $s \to_{\mathcal{R}} u$. Thus $\phi(s) \to_{\phi(\mathcal{R})} \phi(t)$ and $\phi(s) \to_{\phi(\mathcal{R})} \phi(u)$. Confluence of $\phi(\mathcal{R})$ implies $\phi(t) \to^*_{\phi(\mathcal{R})} w$ and $\phi(u) \to^*_{\phi(\mathcal{R})} w$ for some $w \in \{a,b\}^*$. An inductive generalization of the second statement of the lemma implies $w = \phi(v)$ and $t \to^*_{\mathcal{R}} v$, $u \to^*_{\mathcal{R}} v$ for some $v \in \Sigma^*$. Thus $\mathcal{R}$ is confluent. Finally, the only if–direction of the fourth statement of the lemma follows from (1) and the following statement:

$$\text{If } \phi(s) = uv,\ \phi(t) = vw \text{ then } \exists x, y, z \in \Sigma^*\colon\ u = \phi(x),\ v = \phi(y),\ w = \phi(z).$$

Together with (1), this statement implies that every overlapping of two left–hand sides of $\phi(\mathcal{R})$ results from an overlapping of two left–hand sides of $\mathcal{R}$. In particular if $(s,t) \in \{a,b\}^*$ is a critical pair of $\phi(\mathcal{R})$ then $s = \phi(u)$, $t = \phi(v)$ for some $u, v \in \Sigma^*$ and (u,v) is a critical pair of $\mathcal{R}$. Thus if $\mathcal{R}$ is confluent then (u,v) is confluent and thus also (s,t) is confluent with respect to $\phi(\mathcal{R})$. □

Theorem 3.2. COLR(Σ^*) is P–complete under AC^0–reductions for every finite alphabet Σ with $|\Sigma| \geq 2$.

Before we prove Theorem 3.2 we will first investigate the uniform word problem for the class of confluent and length–reducing STSs. It is known that for a confluent and length–reducing STS the word problem is decidable in polynomial time [Boo82].

Theorem 3.3. The uniform word problem for the class of confluent and length–reducing STS over $\{a,b\}^*$ is P–complete under AC^0–reductions.

Proof. Our starting point for the proof of the theorem is the *Generic Machine Simulation Problem*, briefly GMSP, see e.g. [GHR95]. GMSP is the following decision problem.
INPUT: A deterministic Turing machine $\mathcal{M}$, an input word w for $\mathcal{M}$, and a word $t \in \{\#\}^*$, where # is a new symbol which does not occur in the description of $\mathcal{M}$.
QUESTION: Does M terminate on input w after $\leq |t|$ steps ?

It is known that GMSP is P–complete under AC^0–reductions [GHR95]. In GMSP the Turing machine $\mathcal{M}$ is represented by its transition table. We will

(1a)	$q_f x \to q_f$	for all $x \in \Gamma$
(1b)	$x q_f \to q_f$	for all $x \in \Gamma$
(2a)	$\alpha q^{3i} \triangleleft \to \alpha b_l p^{3(i-1)} \triangleleft$	if $\delta(q, \Box) = (p, b, R)$, $1 \leq i \leq m+1$, $\alpha \in \Sigma_l \cup \{\triangleright\}$
(2b)	$\alpha q^{3i} a_r \to \alpha b_l p^{3(i-1)}$	if $\delta(q, a) = (p, b, R)$, $1 \leq i \leq m+1$, $\alpha \in \Sigma_l \cup \{\triangleright\}$
(2c)	$a_l q^{3i} \triangleleft \to p^{3(i-1)} a_r b_r \triangleleft$	if $\delta(q, \Box) = (p, b, L)$, $1 \leq i \leq m+1$
(2d)	$\triangleright q^{3i} \triangleleft \to \triangleright p^{3(i-1)} \Box_r b_r \triangleleft$	if $\delta(q, \Box) = (p, b, L)$, $1 \leq i \leq m+1$
(2e)	$c_l q^{3i} a_r \to p^{3(i-1)} c_r b_r$	if $\delta(q, a) = (p, b, L)$, $1 \leq i \leq m+1$
(2f)	$\triangleright q^{3i} a_r \to \triangleright p^{3(i-1)} \Box_r b_r$	if $\delta(q, a) = (p, b, L)$, $1 \leq i \leq m+1$

Figure 1: The STS $\mathcal{P}(\mathcal{M}, m)$, where $a, b, c \in \Sigma$, $q \in Q \backslash \{q_f\}$, and $p \in Q$.

reduce GMSP to the uniform word problem for confluent and length–reducing STSs over $\{a, b\}^*$. Thus let $(\mathcal{M}, w, \#^m)$ be an instance of GMSP. Here $\mathcal{M} = (Q, \Sigma, \Box, \delta, q_0, q_f)$ is a deterministic Turing machine, where Q is the finite set of states, Σ is the tape alphabet, $\Box \in \Sigma$ is the blank symbol, $\delta : Q \backslash \{q_f\} \times \Sigma \to Q \times \Sigma \times \{L, R\}$ is the total transition function, $q_0 \in Q$ is the initial state, and $q_f \in Q$ is the unique final state. The word $w \in (\Sigma \backslash \{\Box\})^*$ is an input for $\mathcal{M}$. Note that $\mathcal{M}$ terminates if and only if it reaches the final state q_f. Let $\Sigma_l = \{a_l \mid a \in \Sigma\}$ and $\Sigma_r = \{a_r \mid a \in \Sigma\}$ be two disjoint copies of Σ with $\Sigma_l \cap Q = \emptyset = \Sigma_r \cap Q$. The word w_r results from w by replacing every $a \in \Sigma$ by a_r. Let $\triangleright$ (left–end marker) and $\triangleleft$ (right–end marker) be additional symbols and let $\Gamma = Q \cup \Sigma_l \cup \Sigma_r \cup \{\triangleright, \triangleleft\}$. We define the STS $\mathcal{P}(\mathcal{M}, m)$ over Γ^* by the rules of Figure 1. The rules (1a) and (1b) make q_f absorbing. The rules (2a) to (2f) simulate the machine $\mathcal{M}$. Note that the state symbol is represented $3i$ times on the left–hand side and $3(i-1)$ times on the right–hand side. This makes $\mathcal{P}(\mathcal{M}, m)$ length–reducing. It is also easy to see that $\mathcal{P}(\mathcal{M}, m)$ can be computed from $\mathcal{M}$ and $\#^m$ in AC^0 (for this it is necessary that m is given in the unary representation $\#^m$ since $\|\mathcal{P}(\mathcal{M}, m)\|$ increases exponentially with $\mathit{bit}(m)$).
Claim: $\mathcal{P}(\mathcal{M}, m)$ is length–reducing and confluent. Furthermore $\mathcal{M}$ terminates on input w after $\leq m$ steps if and only if $\triangleright q_0^{3(m+1)} w_r \triangleleft \to^*_{\mathcal{P}(\mathcal{M},m)} q_f$.

Confluence of $\mathcal{P}(\mathcal{M}, m)$ is obvious, since only the rules (1a) and (1b) generate critical pairs. Since q_f is absorbing, these critical pairs are confluent. Now assume that $\mathcal{M}$ does not terminate on input w after $\leq m$ steps. By simulating $m+1$ steps of $\mathcal{M}$ we obtain $\triangleright q_0^{3(m+1)} w_r \triangleleft \to^{m+1}_{\mathcal{P}(\mathcal{M},m)} \triangleright u v \triangleleft \in \mathit{IRR}(\mathcal{P}(\mathcal{M}, m))$ for some $u \in \Sigma_l^*$, $v \in \Sigma_r^*$. But then $\triangleright q_0^{3(m+1)} w_r \triangleleft \to^*_{\mathcal{P}(\mathcal{M},m)} q_f$ cannot hold since also $q_f \in \mathit{IRR}(\mathcal{P}(\mathcal{M}, m))$ and $\mathcal{P}(\mathcal{M}, m)$ is confluent. Now assume that $\mathcal{M}$ terminates on input w after $\leq m$ steps. Then for some $j \geq 1$, $u \in \Sigma_l^*$, and $v \in \Sigma_r^*$ it holds $\triangleright q_0^{3(m+1)} w_r \triangleleft \to^*_{\mathcal{P}(\mathcal{M},m)} \triangleright u q_f^{3j} v \triangleleft$. By applying the rules (1a) and (1b), the word $\triangleright u q_f^{3j} v \triangleleft$ can be reduced to q_f. Thus, the claim is proved.

Now consider the STS $\phi(\mathcal{P}(\mathcal{M}, m))$ over the alphabet $\{a, b\}$, where ϕ is the coding function from Lemma 3.1. Then $\phi(\mathcal{P}(\mathcal{M}, m))$ is also confluent and length–reducing and can be calculated from $\mathcal{P}(\mathcal{M}, m)$ (and hence from $\mathcal{M}$ and

$\#^m$) in AC^0. Furthermore $\phi(\triangleright q_0^{3(m+1)} w_r \triangleleft) \leftrightarrow^*_{\phi(\mathcal{P}(\mathcal{M},m))} \phi(q_f)$ if and only if $\phi(\triangleright q_0^{3(m+1)} w_r \triangleleft) \rightarrow^*_{\phi(\mathcal{P}(\mathcal{M},m))} \phi(q_f)$ (since $\phi(\mathcal{P}(\mathcal{M},m))$ is confluent and $\phi(q_f)$ irreducible with respect to $\phi(\mathcal{P}(\mathcal{M},m))$) if and only if $\triangleright q_0^{3(m+1)} w_r \triangleleft \rightarrow^*_{\mathcal{P}(\mathcal{M},m)} q_f$ if and only if $\mathcal{M}$ terminates on w after $\leq m$ steps. This proves the theorem. $\square$

Proof of Theorem 3.2. As mentioned above, $\mathrm{COLR}(\Sigma^*)$ belongs to P. In order to prove the P–hardness of $\mathrm{COLR}(\Sigma^*)$ for $|\Sigma| \geq 2$ it suffices to prove the P–hardness of $\mathrm{COLR}(\{a,b\}^*)$.

Let $(\mathcal{M}, w, \#^m)$ be an instance of GMSP and let $n = 3(m+1)+|w|+2$. Let Γ be the alphabet from the previous proof and let $\mathcal{P}(\mathcal{M},m)$ be the length–reducing and confluent STS from the previous proof. Let A and B be symbols which are not in Γ and define the length–reducing STS $\mathcal{R}(\mathcal{M},w,m)$ over $(\Gamma \cup \{A,B\})^*$ by

$$\mathcal{R}(\mathcal{M},w,m) = \mathcal{P}(\mathcal{M},m) \cup \{A^n B \rightarrow \triangleright q_0^{3(m+1)} w_r \triangleleft,\ AB \rightarrow q_f\}.$$

With the rule $A^n B \rightarrow \triangleright q_0^{3(m+1)} w_r \triangleleft$ we generate an initial configuration for $\mathcal{M}$. Since the initial state q_0 is represented $3(m+1)$ times in the initial configuration, at most $m+1$ steps of $\mathcal{M}$ will be simulated with the rules (2a) to (2f). Of course also $\mathcal{R}(\mathcal{M},w,m)$ can be computed from $\mathcal{M}$, w, and $\#^m$ in AC^0.

By the claim form the proof of Theorem 3.3 the machine $\mathcal{M}$ terminates on input w after $\leq m$ steps if and only if $\triangleright q_0^{3(m+1)} w_r \triangleleft \rightarrow^*_{\mathcal{P}(\mathcal{M},m)} q_f$ which clearly holds if and only if $\triangleright q_0^{3(m+1)} w_r \triangleleft \rightarrow^*_{\mathcal{R}(\mathcal{M},w,m)} q_f$.

Claim : $\mathcal{R}(\mathcal{M},w,m)$ is confluent if and only if $\triangleright q_0^{3(m+1)} w_r \triangleleft \rightarrow^*_{\mathcal{R}(\mathcal{M},w,m)} q_f$.

First assume that $\mathcal{R}(\mathcal{M},w,m)$ is confluent. Since

$$A^n B \rightarrow_{\mathcal{R}(\mathcal{M},w,m)} A^{n-1} q_f \rightarrow^{n-1}_{(1b)} q_f \quad \text{and} \quad A^n B \rightarrow_{\mathcal{R}(\mathcal{M},w,m)} \triangleright q_0^{3(m+1)} w_r \triangleleft$$

and since $q_f \in \mathit{IRR}(\mathcal{R}(\mathcal{M},w,m))$ it must hold $\triangleright q_0^{3(m+1)} w_r \triangleleft \rightarrow^*_{\mathcal{R}(\mathcal{M},w,m)} q_f$. Now assume that $\triangleright q_0^{3(m+1)} w_r \triangleleft \rightarrow^*_{\mathcal{R}(\mathcal{M},w,m)} q_f$ holds. Then the critical pair $(A^{n-1} q_f, \triangleright q_0^{3(m+1)} w_r \triangleleft)$ is confluent. In all other critical pairs one of the rules (1a) or (1b) must be involved. Since q_f is absorbing these critical pairs are also confluent. This proves the claim.

Now consider the STS $\phi(\mathcal{R}(\mathcal{M},w,m))$, where ϕ is the coding function from Lemma 3.1. Then $\phi(\mathcal{R}(\mathcal{M},w,m))$ is confluent if and only if $\mathcal{R}(\mathcal{M},w,m)$ is confluent if and only if $\triangleright q_0^{3(m+1)} w_r \triangleleft \rightarrow^*_{\mathcal{R}(\mathcal{M},w,m)} q_f$ if and only if $\mathcal{M}$ terminates on input w after $\leq m$ steps. $\square$

In the rest of this section we will show that Theorem 3.2 does not hold any longer for monadic STSs unless NC = P. More precisely, we prove that $\mathrm{COMO}(\Sigma^*)$ is contained in AC^1. To the knowledge of the author this result was never stated explicitly, but it easily follows from known results. We start with the *uniform word problem for* 1*–free CFGs* which is the following problem:

INPUT: A 1–free CFG G over a terminal alphabet Σ and a word $s \in \Sigma^*$.

QUESTION: Is $s \in L(G)$?

The following result was implicitly proven in [Ruz80], see also [GHR95], pp. 176.

Lemma 3.4. The uniform word problem for 1–free CFGs is in AC^1.

In [BJW82] it was shown that the question whether a pair (s,t) of words is confluent with respect to a monadic STS can be reduced to the word problem for CFGs. We present the construction from [BJW82] for completeness and in order to convince the reader that it can be carried out in AC^0. Let $\mathcal{R}$ be a monadic STS over Σ^*. With $\mathcal{R}$ we associate a 1–free CFG $G_{\mathcal{R}}$ in the following way. Let $\Sigma_l = \{a_l \mid a \in \Sigma\}$ and $\Sigma_r = \{a_r \mid a \in \Sigma\}$ be two disjoint copys of Σ and let $\#$ be an additional symbol. For a word $s \in \Sigma^*$, s_l and s_r are defined in the obvious way. Then $G_{\mathcal{R}} = (\Sigma_l \cup \Sigma_r \cup \{S, S_l, S_r\}, \Sigma \cup \{\#\}, P, S)$, where P contains all productions of the form

$$\begin{array}{ll} S \Rightarrow a_l S a_r \mid SS_r \mid S_l S \mid \# \ \text{ for } a \in \Sigma, & a_l \Rightarrow a,\ a_r \Rightarrow a \text{ for } a \in \Sigma \\ S_l \Rightarrow s_l,\ S_r \Rightarrow s_r^{rev} \text{ for } (s,1) \in \mathcal{R}, & a_l \Rightarrow s_l,\ a_r \Rightarrow s_r^{rev} \text{ for } a \in \Sigma,\ (s,a) \in \mathcal{R}, \\ x \Rightarrow S_l x \mid x S_l \ \text{ for } x \in \Sigma_l \cup \{S_l\}, & x \Rightarrow S_r x \mid x S_r \ \text{ for } x \in \Sigma_r \cup \{S_r\}. \end{array}$$

Obviously, $G_{\mathcal{R}}$ can be constructed from $\mathcal{R}$ by an AC^0–circuit. Furthermore $L(G_{\mathcal{R}}, S_l) = \{s \in \Sigma^+ \mid s \to_{\mathcal{R}}^* 1\}$, $L(G_{\mathcal{R}}, a_l) = \{s \in \Sigma^* \mid s \to_{\mathcal{R}}^* a\}$ and $L(G_{\mathcal{R}}, S_r) = \{s \in \Sigma^+ \mid s^{rev} \to_{\mathcal{R}}^* 1\}$, $L(G_{\mathcal{R}}, a_r) = \{s \in \Sigma^* \mid s^{rev} \to_{\mathcal{R}}^* a\}$. Thus

$$L(G_{\mathcal{R}}) = \{s\#t^{rev} \mid s \to_{\mathcal{R}}^* u \text{ and } t \to_{\mathcal{R}}^* u \text{ for some } u \in \Sigma^*\}.$$

Now the following theorem is easy to prove.

Theorem 3.5. COMO(Σ^*) is in AC^1.

Proof. Let $\mathcal{R}$ be a monadic STS over Σ^*. First we construct an AC^0–circuit that calculates from $\mathcal{R}$ the set of all critical pairs. This is possible, since in parallel we can test for each pair of rules $l_1 \to r_1$ and $l_2 \to r_2$ and for each factorization $l_1 = st$ with $t \neq 1$ whether l_2 is a prefix of t or t is a prefix of l_2 (using unbounded fan–in this is clearly possible in constant depth). If this is the case, we obtain a critical pair. In a second step we have to test in parallel whether each critical pair $(s,t) \in CP(\mathcal{R})$ is confluent. For this we construct in AC^0 from $\mathcal{R}$ the 1–free CFG $G_{\mathcal{R}}$ and test whether $s\#t^{rev} \in L(G_{\mathcal{R}})$ which can be done in AC^1 by Lemma 3.4. $\square$

4 Vector replacement systems

In [VRL98] it was shown that confluence is decidable but EXSPACE–hard for the class of all vector replacement systems. Based on critical pairs, more feasible upper bounds can be obtained for the length–reducing case. Similarly to STSs, also VRSs yield finite sets of critical pairs [BL81]. Let $\mathcal{R}$ be a VRS over $\Sigma^\oplus$. The set $CP(\mathcal{R})$ of critical pairs of $\mathcal{R}$ contains exactly all pairs $(s,t) \in \Sigma^\oplus \times \Sigma^\oplus$ such that there exist rules $(k,p),(l,r) \in \mathcal{R}$ such that for all $a \in \Sigma$ it holds $|s|_a = max(|k|_a, |l|_a) - |k|_a + |p|_a$ and $|t|_a = max(|k|_a, |l|_a) - |l|_a + |r|_a$. Then $\mathcal{R}$

is locally confluent if and only if all critical pairs are confluent. Note that there are at most $|\mathcal{R}| \cdot (|\mathcal{R}| - 1)$ many critical pairs that are not trivially confluent. For the length–reducing case, testing all critical pairs for confluence leads to a straight–forward PSPACE–algorithm for deciding confluence. In this section we will prove that confluence is moreover PSPACE–complete for the class of all VRSs (without restriction on the dimension), i.e., $\bigcup_{k>0} \mathrm{COLR}(\mathbb{N}^k)$ is PSPACE–complete. Note that the calculation of a normalform of a $s \in \Sigma^{\oplus}$ with respect to a length–reducing VRS may involve a number of steps that is exponential in $bit(s)$. Therefore the calculation of normalforms for the finitely many critical pairs does not lead to a polynomial time algorithm (as it is the case for STSs).

Theorem 4.1. $\bigcup\limits_{k>0} \mathrm{COLR}(\mathbb{N}^k)$ is PSPACE–complete.

Proof. The following problem is known to be PSPACE–complete [Kar72]:
INPUT: A *deterministic linear bounded automaton* $\mathcal{M}$ and an input w for $\mathcal{M}$.
QUESTION: Does $\mathcal{M}$ accept w ?
Let us fix a deterministic linear bounded automaton $\mathcal{M} = (Q, \Sigma, \triangleright, \triangleleft, \delta, q_0, q_f)$ and an input $w \in (\Sigma \backslash \{\triangleright, \triangleleft\})^*$ for $\mathcal{M}$, where Q is the finite set of states, Σ is the tape alphabet, $\triangleright \in \Sigma$ is the left–end marker, $\triangleleft \in \Sigma$ is the right–end marker, $\delta : (Q \backslash \{q_f\}) \times \Sigma \to Q \times \Sigma \times \{L, R\}$ is the transition function, $q_0 \in Q$ is the initial state, and q_f is the unique final state. The transition function must be defined such that

- the read–write head never moves to the left (right) of $\triangleright$ ($\triangleleft$) and
- does not overwrite $\triangleright$ ($\triangleleft$) by a symbol different from $\triangleright$ ($\triangleleft$) and
- does not overwrite a tape symbol $a \in \Sigma \backslash \{\triangleright, \triangleleft\}$ by $\triangleright$ or $\triangleleft$.

We identify each tape cell of $\mathcal{M}$ with a number from $\{0, \ldots, |w| + 1\}$, where cell 0 always contains the left–end marker $\triangleright$ and cell $|w| + 1$ always contains the right–end marker $\triangleleft$. We assume that $\mathcal{M}$ starts with the read–write head scanning cell 0. Note that $\mathcal{M}$ accepts w if and only if it terminates on w if and only if it reaches the final state q_f. Furthermore we may assume that the read–write head is always in cell 0 if the final state q_f is reached.

We will construct a VRS $\mathcal{R}(\mathcal{M}, w)$ such that $\mathcal{R}(\mathcal{M}, w)$ is confluent if and only if $\mathcal{M}$ accepts the input w, which proves the theorem. Our construction is based on the simulation of a linear bounded automaton by a Petri net from [JLL77]. Let us define the alphabet Γ by

$$\Gamma = (\{0, \ldots, |w| + 1\} \times Q) \cup (\{0, \ldots, |w| + 1\} \times \Sigma) \cup \{A, \$\}.$$

Note that we consider pairs $(i, q) \in \{0, \ldots, |w| + 1\} \times Q$ and pairs $(i, a) \in \{0, \ldots, |w| + 1\} \times \Sigma$ as single symbols. The symbol (i, q) means that $\mathcal{M}$ is in the state q and the read–write head is scanning cell i, whereas the symbol (i, a) means that cell i contains the tape symbol a. Since cell 0 always contains $\triangleright$ and cell $|w| + 1$ always contains $\triangleleft$, the symbols $(0, \triangleright)$ and $(|w| + 1, \triangleleft)$ will be always

(1a)	$\$(i,q)(i,a) \to (i+1,p)(i,b)$	if $\delta(q,a) = (p,b,R)$, $q \neq q_f$, $i \leq \lvert w \rvert$
(1b)	$\$(i,q)(i,a) \to (i-1,p)(i,b)$	if $\delta(q,a) = (p,b,L)$, $q \neq q_f$, $i \geq 1$
(2)	$(0,q_f)x \to (0,q_f)$	if $x \in \Gamma$
(3a)	$(i,a)(i,b) \to (0,q_f)$	
(3b)	$(i,p)(j,q) \to (0,q_f)$	
(4a)	$A^n \to \$^m(0,q_0)(0,\triangleright)(1,a_1)\cdots(\lvert w\rvert, a_{\lvert w \rvert})(\lvert w \rvert+1,\triangleleft)$	
(4b)	$A^2 \to (0,q_f)$	

Figure 2: The VRS $\mathcal{R}(\mathcal{M}, w)$, where $p, q \in Q$, $0 \leq i, j \leq |w| + 1$, and $a, b \in \Sigma$.

present. Moreover since we assumed that $\mathcal{M}$ terminates if and only if it reaches the final state q_f and the read–write head is in cell 0, the presence of the symbol $(0, q_f)$ indicates that $\mathcal{M}$ has terminated. The VRS $\mathcal{R}(\mathcal{M}, w)$ over $\Gamma^{\oplus}$ is shown in Figure 2, where we assume that $w = a_1 a_2 \cdots a_{|w|}$ and $m = |Q| \cdot |\Gamma|^{|w|} \cdot (|w|+2)$ and $n = m + |w| + 4$ (we did not fix a linear ordering on Γ, which is necessary for $\Gamma^{\oplus}$, but this can be done in an arbitrary way).

The rules (1a) and (1b) simulate $\mathcal{M}$ where the additional \$ on the left–hand side is necessary in order to make these rules length–reducing. Rule (2) makes $(0, q_f)$ absorbing. With the rules (3a) and (3b) it is possible to resolve critical pairs that result from the rules (1a) and (1b). In particular it is easy to see that the VRS that consists of the rules (1a), (1b), (2), (3a) and (3b) is confluent. With the rules (4a) and (4b) we intentionally create a critical pair. The first rule (4a) produces the encoding of the initial configuration of $\mathcal{M}$. Since each simulation step of $\mathcal{R}(\mathcal{M}, w)$ consumes a \$, we have to make enough \$'s available for the initial configuration. Since there are at most $m = |Q| \cdot |\Gamma|^{|w|} \cdot (|w| + 2)$ different configurations for $\mathcal{M}$, the automaton $\mathcal{M}$ either terminates after $\leq m$ steps or loops forever. Thus m many \$'s suffice. Note that in the binary representation of $\mathcal{R}(\mathcal{M}, w)$ the m many \$'s are represented by $O(ld(m)) = O(ld(|Q|) + |w| \cdot ld(|\Gamma|) + ld(|w| + 2))$ many bits, which is polynomial in $|w|$ and the length of the description of $\mathcal{M}$. The same holds also for the number $n = m + |w| + 4$ in the left–hand side of rule (4a) which is chosen such that (4a) is length–reducing. Finally rule (4b) writes the absorbing symbol $(0, q_f)$.

The proof that $\mathcal{R}(\mathcal{M}, w)$ is confluent if and only if $\mathcal{M}$ accepts w is similar to the proof of Theorem 3.2. First note that $A^n \to_{(4b)} A^{n-2}(0, q_f) \to_{(2)}^{n-2} (0, q_f) \in IRR(\mathcal{R}(\mathcal{M}, w))$ and $A^n \to_{(4a)} \$^m(0, q_0)(0, \triangleright)(1, a_1) \cdots (|w|, a_{|w|})(|w|+1, \triangleleft)$. If $\mathcal{M}$ does not accept w, i.e., if $\mathcal{M}$ does not terminate on input w, then by simulating m steps of $\mathcal{M}$ we obtain

$$\begin{aligned}\$^m(0,q_0)(0,\triangleright)(1,a_1)\cdots(|w|,a_{|w|})(|w|+1,\triangleleft) &\to^m_{\mathcal{R}(\mathcal{M},w)} \\ &(i,q)(0,\triangleright)(1,b_1)\cdots(|w|,b_{|w|})(|w|+1,\triangleleft) \in IRR(\mathcal{R}(\mathcal{M},w))\end{aligned}$$

for some $i \in \{0, \ldots, |w| + 1\}$, $q \in Q\backslash\{q_f\}$, and $b_j \in \Sigma\backslash\{\triangleright, \triangleleft\}$ $(1 \leq j \leq |w|)$. Thus $\mathcal{R}(\mathcal{M}, w)$ is not confluent. On the other hand if $\mathcal{M}$ accepts w, then $\mathcal{M}$

reaches after $k \leq m$ steps the final state q_f and terminates in cell 0. Hence

$$\$^m(0,q_0)(0,\triangleright)(1,a_1)\cdots(|w|,a_{|w|})(|w|+1,\triangleleft) \to^k_{\mathcal{R}(\mathcal{M},w)}$$
$$\$^{m-k}(0,q_f)(0,\triangleright)(1,b_1)\cdots(|w|,b_{|w|})(|w|+1,\triangleleft) \to^+_{(2)} (0,q_f).$$

Since all other critical pairs of $\mathcal{R}(\mathcal{M},w)$ are confluent, $\mathcal{R}(\mathcal{M},w)$ is confluent. □

One might ask whether confluence is also PSPACE–complete for VRSs in a sufficiently large but fixed dimension. The usual technique of coding several symbols into two symbols that we applied for STSs does not work for VRSs. On the other hand there exists a simulation of a 3–counter machine with exponentially bounded counters (for which the acceptance problem is PSPACE–complete) by a VRS in the dimension 6 [Huy85]. But in this simulation unwanted critical pairs arise and it does not seem to be obvious whether these critical pairs can be resolved by adding a polynomial number of additional rules (like the rules (3a) and (3b) in the previous proof).

The following theorem follows easily from the previous proof by using the same arguments that we applied for the proof of Theorem 3.3.

Theorem 4.2. The uniform word problem for length–reducing and confluent VRSs is PSPACE–complete.

Similarly to the semi–Thue case, also for VRSs the complexity of the confluence problem decreases for the monadic case. This is the content of the next theorem.

Theorem 4.3. $\bigcup\limits_{k>0} \mathrm{COMO}(\mathbb{N}^k)$ is in NP.

Proof. The theorem follows easily from the results of [Huy83, Esp97], which state that the *reachability problem for communication–free Petri nets* is in NP (in fact it is NP–complete). In terms of VRSs, a VRS $\mathcal{R}$ is communication–free, if for each left–hand side $l \in dom(\mathcal{R})$ it holds $|l| = 1$. Using this result, we can decide $\bigcup_{k>0} \mathrm{COMO}(\mathbb{N}^k)$ in NP as follows. Given a monadic VRS $\mathcal{R}$ over $\Sigma^\oplus$ we first construct the set of all critical pairs, as defined at the beginning of this section. Let $(s,t) \in \Sigma^\oplus \times \Sigma^\oplus$ be one of these critical pairs. We now guess a $u \in \Sigma^\oplus$ with $|u| \leq |s|$ and $|u| \leq |t|$. It suffices to check in NP whether $s \to^*_{\mathcal{R}} u$ and $t \to^*_{\mathcal{R}} u$. In order to verify whether $s \to^*_{\mathcal{R}} u$ we proceed as follows. Let $\mathcal{P}$ be the following communication–free VRS over $(\Sigma \cup \{\$\})^\oplus$, where $\$ \notin \Sigma$ is a new symbol:

$$\mathcal{P} = \{(r,l) \mid (l,r) \in \mathcal{R}, |r| = 1\} \cup \{(\$,l) \mid (l,1) \in \mathcal{R}\}$$

Then $s \to^*_{\mathcal{R}} u$ if and only if $u\$^i \to^*_{\mathcal{P}} s$ for some $0 \leq i \leq |s| - |u|$. Thus we simply have to guess such a number i (note that $bit(i)$ is bounded by a polynomial in $bit(s)$) and check whether $u\$^i \to^*_{\mathcal{P}} s$, which is possible in NP by the results mentioned above. This concludes the proof. □

Whether $\bigcup_{k>0} \mathrm{COMO}(\mathbb{N}^k)$ is also NP–complete is left as an open question. If we further restrict the input to be a special VRS, then we can even give a simple deterministic polynomial time algorithm that decides confluence.

Theorem 4.4. $\bigcup\limits_{k>0} \mathrm{COSP}(\mathbb{N}^k)$ is in P.

Proof. Let $\mathcal{R}$ be a special VRS over $\Sigma^{\oplus}$, where $\Sigma = \{a_1, \dots, a_n\}$. In order to check whether $\mathcal{R}$ is confluent it suffices to calculate for each critical pair $(s,t) \in \Sigma^{\oplus} \times \Sigma^{\oplus}$ normalforms of s and t and to check whether these normalforms are equal. Thus we have to prove that a normalform of a $s \in \Sigma^{\oplus}$ can be calculated in P. Let $l_1, \dots, l_m$ be the left–hand sides of $\mathcal{R}$. Let $s_0 = s$ and for $1 \leq i \leq m$ let s_i be a normalform of s_{i-1} with respect to the special one–rule VRS $\{l_i \to 1\}$. Then it is easy to see that s_m is a normalform of s. Thus we have to show that the normalform of $a_1^{d_1} \cdots a_n^{d_n} \in \Sigma^{\oplus}$ with respect to a rule $a_1^{e_1} \cdots a_n^{e_n} \to 1$ can be calculated in P. But this is easy. For each $1 \leq i \leq n$ we have to calculate the integer–quotient $q_i = \lfloor d_i / e_i \rfloor$ and take the minimum $q = min\{q_1, \dots, q_n\}$ of these quotients. For $1 \leq i \leq n$ let $f_i = d_i - q \cdot e_i$. Then the normalform is $a_1^{f_1} \cdots a_n^{f_n}$. $\square$

Also the problem whether $\bigcup_{k>0} \mathrm{COSP}(\mathbb{N}^k)$ is P–complete must be left as an open question.

5 Special trace rewriting systems

In contrast to the decidability results of the last two sections, there exists a trace monoid M such that $\mathrm{COLR}(M)$ is undecidable [NO88]. This result was sharpened in [Loh98], where it was shown that $\mathrm{COLR}(M)$ is decidable if and only if M is a free monoid or a free commutative monoid. In particular this implies that in general TRSs cannot have finitely many critical pairs (in contrast to STSs and VRSs). Furthermore these undecidability results lead to the question whether there exist restricted but non trivial classes of (length–reducing) TRSs for which confluence is decidable. In particular, in [Die90b] it was asked whether confluence is decidable for special TRSs and monadic TRSs, respectively. For the special case we answer this question positively in this section. In fact we will prove that confluence is decidable for a broader class of TRSs that satisfy the following condition, which we call condition (A): A TRS $\mathcal{R}$ over $\mathbb{M}(\Sigma, I)$ satisfies condition (A) if

(1) for all $(l_1, r_1), (l_2, r_2) \in \mathcal{R}$ and all factorizations $l_1 = p_1 q_1$, $l_2 = p_2 q_2$ with $p_i \neq 1 \neq q_i$ for $i \in \{1,2\}$, $p_1 \, I \, p_2$, and $q_1 \, I \, q_2$ it holds: There exist factorizations $r_1 = s_1 t_1$, $r_2 = s_2 t_2$ such that $a \, I \, p_i$ implies $a \, I \, s_i$ and $a \, I \, q_i$ implies $a \, I \, t_i$ for all $a \in \Sigma$, $i \in \{1,2\}$.

(2) for all $(l_1, r_1), (l_2, r_2) \in \mathcal{R}$ and all factorizations $l_1 = p_1 s q_1$, $l_2 = p_2 s q_2$ with $s \neq 1$, $p_1 \, I \, p_2$, and $q_1 \, I \, q_2$ it holds: $a \, I \, s$ implies $a r_1 = r_1 a$ and $a r_2 = r_2 a$ for all $a \in \Sigma$.

```
Input: A length–reducing TRS R over M(Σ, I) that satisfies condition (A)
  begin
    forall ((l_1, r_1), (l_2, r_2))) ∈ R × R do
      forall factorizations l_1 = p_1 s q_1, l_2 = p_2 s q_2 with
       s ≠ 1, p_1 I p_1, q_1 I q_2 do
        nf_1 := NF(p_1 r_2 q_1, R); nf_2 := NF(p_2 r_1 q_2, R);
        if nf_1 ≠ nf_2 then
          (*) return "R not confluent"
        else
          u := nf_1(= nf_2);
          forall a ∈ Σ with a I p_2 s q_1 or a I p_1 s q_2 do
            nf_1 := NF(au, R); nf_2 := NF(ua, R);
            if nf_1 ≠ nf_2 then
              (**) return "R not confluent"
          endfor
      endfor
    endfor
    (***) return "R confluent"
  end
```

Figure 3: The algorithm CONFL

Note that condition (A2) implies that for every rule $(l, r) \in \mathcal{R}$ if $a\, I\, l$ then $ar = ra$.

Theorem 5.1. The following problem is decidable for every trace monoid M:
INPUT: A length-reducing TRS $\mathcal{R}$ over M that satisfies condition (A).
QUESTION: Is $\mathcal{R}$ confluent?

Proof. Let $M = \mathbb{M}(\Sigma, I)$ be a trace monoid and let $\mathcal{R}$ be a length–reducing TRS over M that satisfies condition (A). Let *NF* be an algorithm that computes an arbitrary normalform $NF(u, \mathcal{R})$ of a given input trace u with respect to $\mathcal{R}$. Consider the algorithm CONFL in Figure 3. We claim that CONFL outputs "$\mathcal{R}$ confluent" if and only if $\mathcal{R}$ is confluent. First we prove that $\mathcal{R}$ is not confluent if CONFL outputs "$\mathcal{R}$ not confluent". If CONFL executes line $(*)$ then there exist rules $l_1 \to r_1$ and $l_2 \to r_2$ in $\mathcal{R}$ and factorizations $l_1 = p_1 s q_1$, $l_2 = p_2 s q_2$ such that $s \neq 1$, $p_1\, I\, p_2$, and $q_1\, I\, q_2$. Furthermore there exists a normalform u_1 of $p_1 r_2 q_1$ and a normalform u_2 of $p_2 r_1 q_2$ such that $u_1 \neq u_2$. But then $\mathcal{R}$ is indeed not confluent since $p_2 p_1 s q_1 q_2 \to_{\mathcal{R}} p_2 r_1 q_2 \to^*_{\mathcal{R}} u_2$ and $p_2 p_1 s q_1 q_2 = p_1 p_2 s q_2 q_1 \to_{\mathcal{R}} p_1 r_2 q_1 \to^*_{\mathcal{R}} u_1$. Now assume that CONFL executes line $(**)$. Then it holds $u_1 = u_2 = u$ but there exists an $a \in \Sigma$ such that either $a\, I\, p_2 s q_1$ or $a\, I\, p_1 s q_2$ and there exist a normalform v_1 of au and a normalform v_2 of ua such that $v_1 \neq v_2$. Assume that $a\, I\, p_2 s q_1$. Since $\mathcal{R}$ satisfies condition

(A2) it follows $ar_1 = r_1a$ and $ar_2 = r_2a$. Hence

$$p_2p_1sq_1aq_2 \to_{\mathcal{R}} p_2r_1aq_2 = ap_2r_1q_2 \to^*_{\mathcal{R}} au \to^*_{\mathcal{R}} v_1 \quad \text{and}$$
$$p_2p_1sq_1aq_2 = p_1p_2sq_1aq_2 = p_1ap_2sq_2q_1 \to_{\mathcal{R}} p_1ar_2q_1 = p_1r_2q_1a \to^*_{\mathcal{R}} ua \to^*_{\mathcal{R}} v_2.$$

Thus, again $\mathcal{R}$ is not confluent. The case that $a\,I\,p_1sq_2$ can be dealt similarly by considering the trace $p_2ap_1sq_1q_2$ instead of $p_2p_1sq_1aq_2$.

Now assume that CONFL outputs "$\mathcal{R}$ confluent" in line $(***)$. We have to show that $\mathcal{R}$ is confluent. By induction on the length of traces it suffices to prove the following implication:

If $\mathcal{R}$ is confluent on all traces t' with $|t'| < |t|$ then $\mathcal{R}$ is confluent on t.

Thus, let $t \in M$ and assume that $\mathcal{R}$ is confluent on all traces t' with $|t'| < |t|$. We have to prove that all pairs (t_1, t_2) such that $t \to^i_{\mathcal{R}} t_1$ and $t \to^j_{\mathcal{R}} t_2$ for some $i, j \geq 0$ are confluent. Of course the case $i = 0$ or $j = 0$ is trivial. Let us assume for a moment that we have already considered all cases with $i = 1 = j$. Then we can apply the same arguments as in the (standard) proof of Newman's lemma: $t \to_{\mathcal{R}} s_1 \to^*_{\mathcal{R}} t_1$ and $t \to_{\mathcal{R}} s_2 \to^*_{\mathcal{R}} t_2$ imply that there exists a trace s with $s_i \to^*_{\mathcal{R}} s$ $(i \in \{1,2\})$. Since $|s_1| < |t|$ and $s_1 \to^*_{\mathcal{R}} t_1$, $s_1 \to^*_{\mathcal{R}} s$ it holds $t_1 \to^*_{\mathcal{R}} u$ and $s \to^*_{\mathcal{R}} u$ for some trace u. Since also $|s_2| < |t|$ and $s_2 \to^*_{\mathcal{R}} t_2$, $s_2 \to^*_{\mathcal{R}} s \to^*_{\mathcal{R}} u$ it holds $t_2 \to^*_{\mathcal{R}} v$ and $u \to^*_{\mathcal{R}} v$, i.e., $t_1 \to^*_{\mathcal{R}} u \to^*_{\mathcal{R}} v$ for some trace v.

Thus, it suffices to consider arbitrary factorizations $t = u_1l_1v_1 = u_2l_2v_2$ where $(l_1, r_1), (l_2, r_2) \in \mathcal{R}$. We have to prove that the pair $(u_1r_1v_1, u_2r_2v_2)$ is confluent. Lemma 2.1 applied to the identity $u_1l_1v_1 = u_2l_2v_2$ gives nine traces y_i, p_i, q_i $(i \in \{1,2\})$ and s such that (see also the diagram below)

- $l_1 = p_1sq_1,\ \ l_2 = p_2sq_2,$
- $u_1 = y_1p_2w_2,\ \ u_2 = y_1p_1w_1,$
- $v_1 = w_1q_2y_2,\ \ v_2 = w_2q_1y_2,$
- $t = y_1p_1w_1p_2sq_2w_2q_1y_2 = y_1p_2w_2p_1sq_1w_1q_2y_2,$
- $p_1\,I\,p_2,\ \ q_1\,I\,q_2,\ \ w_1\,I\,p_2sw_2q_1,\ \ w_2\,I\,p_1w_1sq_2.$

v_2	w_2	q_1	y_2
l_2	p_2	s	q_2
u_2	y_1	p_1	w_1
	u_1	l_1	v_1

We have to show that the pair $(y_1p_1w_1r_2w_2q_1y_2,\ y_1p_2w_2r_1w_1q_2y_2)$ is confluent. First assume that either $y_1 \neq 1$ or $y_2 \neq 1$. For the trace $t' = p_1w_1p_2sq_2w_2q_1 = p_2w_2p_1sq_1w_1q_2$ it holds $|t'| < |t|$ and

$$t' = p_1w_1l_2w_2q_1 \to_{\mathcal{R}} p_1w_1r_2w_2q_1, \qquad t' = p_2w_2l_1w_1q_2 \to_{\mathcal{R}} p_2w_2r_1w_1q_2.$$

Thus the pair $(p_1w_1r_2w_2q_1, p_2w_2r_1w_1q_2)$ is confluent. But then the same also holds for the pair $(y_1p_1w_1r_2w_2q_1y_2, y_1p_2w_2r_1w_1q_2y_2)$. Thus we may assume that $y_1 = y_2 = 1$ and we have to consider the pair $(p_1w_1r_2w_2q_1, p_2w_2r_1w_1q_2)$.

Next assume that $s = 1$, i.e., $l_1 = p_1q_1$ and $l_2 = p_2q_2$. First assume that also $p_1 = 1$, i.e., $l_1 = q_1$. We have to show that the pair $(w_1r_2w_2q_1, p_2w_2r_1w_1q_2)$ is confluent. Since $w_1r_2w_2q_1 = w_1r_2w_2l_1 \rightarrow_{\mathcal{R}} w_1r_2w_2r_1$ it suffices to prove that also $p_2w_2r_1w_1q_2 \rightarrow_{\mathcal{R}} w_1r_2w_2r_1$. This can be deduced as follows: Since $q_1 \, I \, w_1q_2$, i.e., $l_1 \, I \, w_1q_2$, condition (A2) (more precisely the remark after condition (A2)) for $\mathcal{R}$ implies $w_1r_1 = r_1w_1$ and $q_2r_1 = r_1q_2$. Thus

$$\begin{aligned} p_2w_2r_1w_1q_2 = & \quad p_2w_2w_1q_2r_1 && \text{(since } r_1w_1q_2 = w_1q_2r_1\text{)} \\ = & \quad w_1p_2q_2w_2r_1 && \text{(since } w_1 \, I \, w_2\text{, } w_1 \, I \, p_2\text{, and } w_2 \, I \, q_2\text{)} \\ \rightarrow_{\mathcal{R}} & \; w_1r_2w_2r_1 \end{aligned}$$

Thus, we may assume that $p_1 \neq 1$. Similarly we may assume that also $q_1 \neq 1$, $p_2 \neq 1$, and $q_2 \neq 1$. But then condition (A1) implies that there exist factorizations $r_1 = s_1t_1$, $r_2 = s_2t_2$ such that $a \, I \, p_i$ implies $a \, I \, s_i$ and $a \, I \, q_i$ implies $a \, I \, t_i$ for all $a \in \Sigma$. In particular it holds

$$p_2 \, I \, s_1, \; w_2 \, I \, s_1, \; p_1 \, I \, s_2, \; w_1 \, I \, s_2, \; q_2 \, I \, t_1, \; w_1 \, I \, t_1, \; q_1 \, I \, t_2, \; w_2 \, I \, t_2.$$

Furthermore $p_2 \, I \, s_1$ implies $s_2 \, I \, s_1$ and $q_2 \, I \, t_1$ implies $t_2 \, I \, t_1$. Thus, we obtain

$$\begin{aligned} p_2w_2r_1w_1q_2 = p_2w_2s_1t_1w_1q_2 = s_1w_1p_2q_2w_2t_1 \rightarrow_{\mathcal{R}} s_1w_1s_2t_2w_2t_1 = s_2w_2s_1t_1w_1t_2, \\ p_1w_1r_2w_2q_1 = p_1w_1s_2t_2w_2q_1 = s_2w_2p_1q_1w_1t_2 \rightarrow_{\mathcal{R}} s_2w_2s_1t_1w_1t_2. \end{aligned}$$

In the rest of the proof we assume that $s \neq 1$. But then we have one of the situations that are considered in the two outermost **forall**–loops of CONFL. Since we assume that CONFL outputs "$\mathcal{R}$ confluent" we know that there exists a trace u such that $p_1r_2q_1 \rightarrow^*_{\mathcal{R}} u$ and $p_2r_1q_2 \rightarrow^*_{\mathcal{R}} u$. Furthermore since $\mathcal{R}$ satisfies condition (A2) and $s \, I \, w_1w_2$ it holds $r_iw_j = w_jr_i$ for $i, j \in \{1, 2\}$. Hence $p_1w_1r_2w_2q_1 = w_2p_1r_2q_1w_1 \rightarrow^*_{\mathcal{R}} w_2uw_1$ and $p_2w_2r_1w_1q_2 = w_1p_2r_1q_2w_2 \rightarrow^*_{\mathcal{R}} w_1uw_2$ and it suffices to prove that the pair (w_2uw_1, w_1uw_2) is confluent. The case $w_1 = 1 = w_2$ is trivial. Thus assume w.l.o.g. $w_1 = wa$, where $a \in \Sigma$. Since $w_1 \, I \, p_2sw_2q_1$ it follows $a \, I \, p_2sq_1$. Thus $a \in \Sigma$ is one of the symbols that are considered in the innermost **forall**–loop of CONFL. It follows that there exists a trace v such that $au \rightarrow^*_{\mathcal{R}} v$ and $ua \rightarrow^*_{\mathcal{R}} v$. Thus

$$wuaw_2 \rightarrow^*_{\mathcal{R}} wvw_2 \quad \text{and} \quad w_1uw_2 = wauw_2 \rightarrow^*_{\mathcal{R}} wvw_2. \tag{2}$$

Next let us consider the trace $t' = p_1wp_2sq_2w_2q_1 = p_1wl_2w_2q_1$ (t' results from t by replacing the factor $w_1 = wa$ by w). It holds $|t'| < |t|$ and since w satisfies the same independencies as w_1 it holds $t' = p_2w_2p_1sq_1wq_2 = p_2w_2l_1wq_2$. Thus we obtain (note that $wr_1 = r_1w$ and $wr_2 = r_2w$)

$$\begin{aligned} t' &\rightarrow_{\mathcal{R}} p_1wr_2w_2q_1 = w_2p_1r_2q_1w \rightarrow^*_{\mathcal{R}} w_2uw \quad \text{and} \\ t' &\rightarrow_{\mathcal{R}} p_2w_2r_1wq_2 = wp_2r_1q_2w_2 \rightarrow^*_{\mathcal{R}} wuw_2. \end{aligned}$$

Hence there exists a trace x such that $w_2uw \rightarrow^*_{\mathcal{R}} x$ and $wuw_2 \rightarrow^*_{\mathcal{R}} x$. It follows

$$w_2uw_1 = w_2uwa \rightarrow^*_{\mathcal{R}} xa \quad \text{and} \quad wuaw_2 = wuw_2a \rightarrow^*_{\mathcal{R}} xa. \tag{3}$$

Finally since $wuaw_2 \to_{\mathcal{R}}^* wvw_2$ by (2) and $wuaw_2 \to_{\mathcal{R}}^* xa$ by (3) and $|wuaw_2| = |w_1uw_2| \leq |w_1p_1r_2q_1w_2| < |p_1w_1p_2sq_2w_2q_1| = |t|$ (where the strict inequality follows from $|r_2| < |p_2sq_2|$) there exists a trace z such that $wvw_2 \to_{\mathcal{R}}^* z$ and $xa \to_{\mathcal{R}}^* z$. Now we obtain $w_1uw_2 \to_{\mathcal{R}}^* wvw_2 \to_{\mathcal{R}}^* z$ from (2) and $w_2uw_1 \to_{\mathcal{R}}^* xa \to_{\mathcal{R}}^* z$ from (3). Thus the pair (w_1uw_2, w_2uw_1) is confluent and the correctness of CONFL is proved. □

Theorem 5.2. COSP(M) is in P for every trace monoid M.

Proof. For special VRSs we already proved the statement of the theorem. On the other hand if $I \neq (\Sigma \times \Sigma)\backslash Id_\Sigma$ then CONFL runs in polynomial time. This follows from the following two facts: (i) For a fixed independence alphabet (Σ, I), the number of different factorizations $l = psq$ of a trace l is bounded by a polynomial in the length of $|l|$. This follows from the fact that the number of prefixes of a trace t is bounded by a polynomial in $|t|$ [BMS89]. (ii) A normalform of a trace t with respect to a length–reducing TRS $\mathcal{R}$, which is not a VRS, can be calculated in time bounded by a polynomial in $|t|$ and $\|\mathcal{R}\|$, see [Die90c, Die90a, BD95, Ber95, BD96] for the normalform problem [1]. □

One might further ask, whether confluence can be decided for arbitrary monadic TRSs. We leave this as an open question.

We close this section with a simple problem that is decidable for monadic STSs but in general undecidable for special TRSs. A trace $u \in \mathbb{M}(\Sigma, I)$ is said to be *connected* if there does not exist a factorization $u = vw$ such that $v \neq 1 \neq w$ and $v\,I\,w$. A set $L \subseteq \mathbb{M}(\Sigma, I)$ is connected if every trace in L is connected. A set $L \subseteq \mathbb{M}(\Sigma, I)$ is *recognizable* if the set $\{s \in \Sigma^* \mid \exists u \in L : u = [s]_I\}$ of all words that represent a trace in L is recognizable. This is just one of several different possibilities of defining recognizable trace languages, see e.g. chapter 6 of [DR95]. A fundamental result of Ochmański [Och85], states that the class of all recognizable trace languages in $\mathbb{M}(\Sigma, I)$ is the smallest class $\mathcal{C}$ that contains all singleton subsets of $\mathbb{M}(\Sigma, I)$ and that is closed under (i) union, (ii) concatenation of two languages (where the concatenation of L_1 and L_2 is $L_1L_2 = \{u_1u_2 \mid u_1 \in L_1, u_2 \in L_2\}$) and (iii) the star–operator restricted to connected languages, i.e., if $L \in \mathcal{C}$ is connected then also $L^* = \{u_1u_2\cdots u_n \mid n \geq 0, u_1, u_2, \ldots, u_n \in L\} \in \mathcal{C}$. Given a TRS $\mathcal{R}$ over M and a set $L \subseteq M$ we denote by $\Delta_{\mathcal{R}}^*(L) = \{v \in M \mid \exists u \in L : u \to_{\mathcal{R}}^* v\}$ the set of all *descendants* of L with respect to $\mathcal{R}$. It is known that if $L \subseteq \Sigma^*$ is a recognizable word language and $\mathcal{R}$ is a monadic STS then also $\Delta_{\mathcal{R}}^*(L)$ is recognizable [BO85]. But already for special TRSs that contain only one rule this fact does not hold in general, as the following example shows.

Example 5.3. Let $\Sigma = \{a, b, c\}$ and $I = \{(a, c), (c, a)\}$. Since the trace $u = [abc]_I$ is connected, the language $\{u\}^*$ is recognizable. Let $\mathcal{R}$ be the special TRS $\{b \to 1\}$. Assume that $\Delta_{\mathcal{R}}^*(\{u\}^*)$ is recognizable. Since recognizable trace languages are closed under intersection, also the language $\Delta_{\mathcal{R}}^*(\{u\}^*) \cap \{a, c\}^* =$

[1] The algorithms in [Die90c, Die90a, BD95, Ber95, BD96] are all non uniform, i.e., the TRSs is fixed. But it is easy to see that they run also in the uniform case in polynomial time.

$\{[a^n c^n]_I \mid n \geq 0\}$ would be recognizable. But this is not the case, see e.g. [DR95], pp 172.

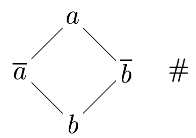
For a class $\mathcal{C}$ of TRSs over a trace monoid M we define the *extended word problem* for $\mathcal{C}$ and M as follows [BO85]:
INPUT: A TRS $\mathcal{R} \in \mathcal{C}$ and two recognizable languages L_1 and L_2 of M.
QUESTION: Do there exist $u_1 \in L_1$ and $u_2 \in L_2$ such that $u_1 \leftrightarrow^*_{\mathcal{R}} u_2$?

A simple consequence of the above mentioned closure of recognizable word languages under the operator $\Delta^*_{\mathcal{R}}$ for a monadic STS $\mathcal{R}$ is that the extended word problem for monadic and confluent STSs is decidable [BO85]. In contrast to this, the following theorem holds.

Theorem 5.4. There exists a trace monoid $M = \mathbb{M}(\Sigma, I)$, a special TRS $\mathcal{R}$ over M of the form $\mathcal{R} = \{a \to 1\}$, where $a \in \Sigma$, and a recognizable language $L_1 \subseteq M$ such that the following problem is undecidable.
INPUT: A recognizable language $L_2 \subseteq M$.
QUESTION: Do there exist $u_1 \in L_1$ and $u_2 \in L_2$ such that $u_1 \leftrightarrow^*_{\mathcal{R}} u_2$?

Proof. It is well–known that the Post Correspondence Problem, briefly PCP, is undecidable over a two–element alphabet. Furthermore it can be required that every solution of the PCP has to start with a distinguished pair. Thus the following problem is undecidable, which is called the modified PCP over $\{a, b\}$:
INPUT: A set $\{(s_1, t_1), \ldots, (s_n, t_n)\}$ of pairs with $s_i, t_i \in \{a, b\}^*$ for $1 \leq i \leq n$.
QUESTION: Does there exist a $i_1 i_2 \cdots i_k \in \{1, \ldots, n\}^*$ with $s_1 s_{i_1} s_{i_2} \cdots s_{i_k} = t_1 t_{i_1} t_{i_2} \cdots t_{i_k}$?

Let $P = \{(s_1, t_1), \ldots, (s_n, t_n)\}$ be an instance of the modified PCP over $\{a, b\}$. Let $\{\overline{a}, \overline{b}\}$ be a copy of $\{a, b\}$ and let $\#$ be an additional symbol. Let $\Sigma = \{a, b, \overline{a}, \overline{b}, \#\}$ and define an independence relation I on Σ by the following graph:



Thus the symbols in $\{a, b\}$ and $\{\overline{a}, \overline{b}\}$ pairwise commute whereas $\#$ is dependent from all symbols. For a word $s \in \{a, b\}^*$ the word $\overline{s}$ is defined in the obvious way. Let $L_1 = \{[a\#\overline{a}]_I, [b\#\overline{b}]_I\}^*$ and $L_2 = \{[s_1\#\overline{t}_1]_I\}\{[s_i\#\overline{t}_i]_I \mid 1 \leq i \leq n\}^*$. By Ochmański's theorem both L_1 and L_2 are recognizable subsets of $\mathbb{M}(\Sigma, I)$. Let $\mathcal{R} = \{\# \to 1\}$. Thus $\mathcal{R}$ is special and confluent. Now the PCP P has a solution if and only if

$$\begin{aligned}
&\exists u_1 \in L_1, u_2 \in L_2, v \in \mathit{IRR}(\mathcal{R}) : u_1 \to^*_{\mathcal{R}} v,\ u_2 \to^*_{\mathcal{R}} v \quad \text{if and only if}\\
&\exists u_1 \in L_1, u_2 \in L_2, v \in \mathbb{M}(\Sigma, I) : u_1 \to^*_{\mathcal{R}} v,\ u_2 \to^*_{\mathcal{R}} v \quad \text{if and only if}\\
&\exists u_1 \in L_1, u_2 \in L_2 : u_1 \leftrightarrow^*_{\mathcal{R}} u_2
\end{aligned}$$

where the last equivalence holds since $\mathcal{R}$ is confluent. $\square$

It might be an interesting problem to characterize those trace monoids for which the extended word problem for confluent and monadic (or special) TRSs is decidable.

6 Conclusion

In this paper we have investigated the complexity of the confluence problem for restricted kinds of semi–Thue systems, vector replacement systems and general trace rewriting systems. We would like to close this paper with a list several questions that remain unsolved.

- What is the complexity of the confluence problem for length–reducing vector replacement system if the dimension is fixed? Note that in Theorem 4.1, the dimension is not fixed.
- Are the upper bounds given in Theorem 4.3, Theorem 4.4 and Theorem 5.2 sharp, i.e., are the decision problems, considered in these theorems, also hard for the given complexity classes?
- Is confluence decidable for general monadic trace rewriting systems. This question was already asked in [Die90b]?

References

[BD95] M. Bertol and V. Diekert. On efficient reduction-algorithms for some trace rewriting systems. In H. Common and J.-P. Jouannaud, editors, *Term Rewriting.*, number 909 in Lecture Notes in Computer Science, pages 114–126, Berlin-Heidelberg-New York, 1995. Springer.

[BD96] M. Bertol and V. Diekert. Trace rewriting: Computing normal forms in time $\mathcal{O}(n \log n)$. In C. Puech and R. Reischuk, editors, *Proceedings of the13th Annual Symposium on Theoretical Aspects of Computer Science 1996*, number 1046 in Lecture Notes in Computer Science, pages 269–280, Berlin-Heidelberg-New York, 1996. Springer.

[Ber95] M. Bertol. Efficient rewriting in cograph trace monoids. In H. Reichel, editor, *Proceedings of the10th Fundamentals of Computation Theory (FCT '95), Dresden (Germany) 1995*, number 965 in Lecture Notes in Computer Science, pages 146–155, Berlin-Heidelberg-New York, 1995. Springer.

[BJW82] R.V. Book, M. Jantzen, and C. Wrathall. Monadic thue systems. *Theoretical Computer Science*, 19:231–251, 1982.

[BL81] A. M. Ballantyne and D. S. Lankford. New decision algorithms for finitely presented commutative semigroups. *Comput. and Maths. with Appls.*, 7:159–165, 1981.

[BMS89] A. Bertoni, G. Mauri, and N. Sabadini. Membership problems for regular and context free trace languages. *Information and Computation*, 82:135–150, 1989.

[BO81] R.V. Book and C.P. O'Dunlaing. Testing for the church-rosser property (note). *Theoretical Computer Science*, 16:223–229, 1981.

[BO85] R.V. Book and F. Otto. Cancellation rules and extended word problems. *Information Processing Letters*, 20:5–11, 1985.

[BO93] R.V. Book and F. Otto. *String–Rewriting Systems*. Springer–Verlag, 1993.

[Boo82] R.V. Book. Confluent and other types of Thue systems. *Journal of the ACM*, 29(1):171–182, January 1982.

[BS90] R.P. Boppana and M. Sipser. The complexity of finite functions. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science (Volume A: Algorithms and Complexity)*. Elsevier and MIT Press, 1990.

[CP85] R. Cori and D. Perrin. Automates et commutations partielles. *R.A.I.R.O. — Informatique Théorique et Applications*, 19:21–32, 1985.

[Die90a] V. Diekert. Combinatorial rewriting on traces. In C. Choffrut et al., editors, *Proceedings of the7th Annual Symposium on Theoretical Aspects of Computer Science (STACS'90), Rouen (France) 1990*, number 415 in Lecture Notes in Computer Science, pages 138–151, Berlin-Heidelberg-New York, 1990. Springer.

[Die90b] V. Diekert. *Combinatorics on Traces*. Number 454 in Lecture Notes in Computer Science. Springer, Berlin-Heidelberg-New York, 1990.

[Die90c] V. Diekert. Word problems over traces which are solvable in linear time. *Theoretical Computer Science*, 74:3–18, 1990.

[DR95] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.

[Esp97] J. Esparza. Petri nets, commutative context–free grammars, and basic parallel processes. *Fundamenta Informatica*, 30:23–41, 1997.

[GHR95] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, 1995.

[Huy83] D. T. Huynh. Commutative grammars: The complexity of uniform word problems. *Information and Control*, 57:21–39, 1983.

[Huy85] D. T. Huynh. Complexity of the word problem for commutative semigroups of fixed dimension. *Acta Informatica*, 22:421–432, 1985.

[JLL77] N. D. Jones, L. H. Landweber, and Y. E. Lien. Complexity of some problems in petri nets. *Theoretical Computer Science*, 4:277–299, 1977.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

[KKMN85] D. Kapur, M. S. Krishnamoorthy, R. McNaughton, and P. Narendran. An $O(|T|^3)$ algorithm for testing the Church-Rosser property of Thue systems. *Theoretical Computer Science*, 35(1):109–114, January 1985.

[Loh98] M. Lohrey. On the confluence of trace rewriting systems. In V. Arvind and R. Ramanujam, editors, *Foundations of Software Technology and Theoretical Computer Science*, volume 1530 of *Lect. Notes Comput. Sci.*, pages 319–330, 1998.

[Maz77] A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, Aarhus, 1977.

[NB72] M. Nivat and M. Benois. Congruences parfaites et quasi–parfaites. *Seminaire Dubreil*, 25(7–01–09), 1971–1972.

[New43] M. H. A. Newman. On theories with a combinatorial definition of "equivalence". *Annals Mathematics*, 43:223–243, 1943.

[NO88] P. Narendran and F. Otto. Preperfectness is undecidable for Thue systems containing only length-reducing rules and a single commutation rule. *Information Processing Letters*, 29:125–130, 1988.

[Och85] E. Ochmański. Regular behaviour of concurrent systems. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 27:56–67, October 1985.

[Pap94] C.H. Papadimitriou. *Computational Complexity.* Addison Wesley, 1994.

[Ruz80] W. L. Ruzzo. Tree–size bounded alternation. *Journal of Computer and System Sciences*, 21:218–235, 1980.

[VRL98] R. M. Verma, M. Rusinowitch, and D. Lugiez. Algorithms and reductions for rewriting problems. In *Proceedings 9th Conference on Rewriting Techniques and Applications, Tsukuba (Japan)*, volume 1379 of *Lecture Notes in Computer Science*, pages 166–180. Springer-Verlag, 1998.