Complexity Results for Confluence Problems

Markus Lohrey

Universität Stuttgart, Institut für Informatik
Breitwiesenstr. 20–22, 70565 Stuttgart, Germany
e–mail: lohreyms@informatik.uni-stuttgart.de

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 NC^2 in the monadic case. For length–reducing vector replacement systems we prove that the confluence problem is PSPACE–complete and that the complexity reduces to NP and P for monadic systems and special systems, 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. Two of the most intensively studied types of rewriting systems are semi–Thue systems [BO93], which operate on free monoids, and vector replacement systems (or equivalently Petri–nets), which operate on free commutative monoids. Both of these types of rewriting systems may be seen as special cases of trace rewriting systems [Die90]. 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.

Confluence is a very desirable property for all kinds of rewriting systems since it implies that the order in which rewrite steps are performed is irrelevant. Several decidability and undecidability results are known for the confluence problem for the different types of rewriting systems mentioned above: For length–reducing semi–Thue systems confluence can be decided in polynomial time, see e.g. [BO93], Corollary 3.2.2. On the other hand 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 EXPSPACE–hard for the class of all vector replacement systems.

In this paper we will continue the investigation of the confluence problem for different kinds of rewriting systems. In Section 3 we will prove that confluence 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 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 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) 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 [Die90]. We end this paper by showing that in contrast to semi–Thue systems the extended word problem, see [BO85], is undecidable even for special trace rewriting systems that contain only one rule. Proofs that are omitted in this paper can be found in the long version [Loh99].

2 Preliminaries

In this section we introduce some notations that we will use in this paper. For an alphabet Σ, Σ^* denotes the set of all finite words of elements of Σ. The empty word is denoted by 1. The length of the word s is denoted by $|s|$. As usual $\Sigma^+ = \Sigma^* \backslash \{1\}$ and $\Sigma^n = \{s \in \Sigma^* \mid |s| = n\}$. The set of all letters that occur in the word s is denoted by $alph(s)$. For a natural number $n \in \mathbb{N}$ let $ld(n)$ denote the logarithm of n to the base 2. Let $bit(n) = \lfloor ld(n) \rfloor + 1$ if $n > 0$ and $bit(0) = 1$, i.e., $bit(n)$ is the length of the binary representation of n. For a vector $\underline{n} = (n_1, \ldots, n_k) \in \mathbb{N}^k$ let $bit(\underline{n}) = bit(n_1) + \cdots + bit(n_k)$. 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 e.g. [Pap94]. Let us just briefly mention the definition of the parallel complexity class NC^k where $k \geq 1$, see [GHR95] for more details. A language $L \subseteq \{a, b\}^*$ is in NC^k if for every $n \geq 1$ there exists a Boolean circuit with n linearly ordered inputs that (i) can be calculated from n in deterministic logarithmic space, (ii) contains $n^{O(1)}$ many gates of fan–in at most two, (iii) has depth $O(ld^k(n))$, and (iv) accepts the language $L \cap \{a, b\}^n$, where a (b) corresponds to the truth value true (false).

In the following we introduce some notions concerning trace theory, see [DR95] for more details. An *independence alphabet* (Σ, I) consists of a finite alphabet Σ and an irreflexive and symmetric relation $I \subseteq \Sigma \times \Sigma$, 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 empty trace $[1]_I$ 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 $alph([s]_I) = alph(s)$. We write $u \; I \; v$ if $alph(u) \times alph(v) \subseteq I$. For the rest of this section let (Σ, I) be an arbitrary independence alphabet and let $M = \mathbb{M}(\Sigma, I)$. If $I = (\Sigma \times \Sigma)\backslash \mathrm{Id}_\Sigma$, where $\mathrm{Id}_\Sigma = \{(a,a) \mid a \in \Sigma\}$, then M is isomorphic to the *free commutative monoid* $\mathbb{N}^{|\Sigma|}$ over $|\Sigma|$ many generators and we identify traces from M with $|\Sigma|$–dimensional vectors over $\mathbb{N}$. On the other hand if $I = \emptyset$ then M is isomorphic to the free monoid Σ^*. The following lemma is a simple generalization of the well known Levi's lemma for traces [CP85], see [Loh99] for a proof of this generalization.

Lemma 1. *Let $u_1, u_2, u_3, v_1, v_2, v_3 \in M$. Then $u_1u_2u_3 = v_1v_2v_3$ iff there exist $w_{i,j} \in M$ $(1 \leq i, j \leq 3)$ such that (i) $u_i = w_{i,1}w_{i,2}w_{i,3}$ for $1 \leq i \leq 3$, (ii) $v_j = w_{1,j}w_{2,j}w_{3,j}$ for $1 \leq j \leq 3$, and (iii) $w_{i,j} \, I \, w_{k,l}$ if $i < k$ and $l < j$.*

v_3	$w_{1,3}$	$w_{2,3}$	$w_{3,3}$
v_2	$w_{1,2}$	$w_{2,2}$	$w_{3,2}$
v_1	$w_{1,1}$	$w_{2,1}$	$w_{3,1}$
	u_1	u_2	u_3

The diagram on the right visualizes the situation in the lemma. The i–th column represents u_i, the j–th row represents v_j, the intersection of the i–th column and the j–th row represents $w_{i,j}$, and $w_{i,j}$ and $w_{k,l}$ are independent if one of them is north–west of the other one.

A *trace rewriting system*, briefly TRS, over the trace monoid M is a finite subset of $M \times M$. In the rest of this section let $\mathcal{R}$ be a TRS over M. 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 more details on STSs. On the other hand if $I = (\Sigma \times \Sigma)\backslash \mathrm{Id}_\Sigma$, i.e., $M \simeq \mathbb{N}^{|\Sigma|}$, then $\mathcal{R}$ is also called a *vector replacement system*, briefly VRS, in the dimension $|\Sigma|$. Vector replacement systems are easily seen to be equivalent to Petri–nets. An element $(\ell, r) \in \mathcal{R}$ is also denoted by $\ell \to r$. The set $\{\ell \mid \exists r \in M : (\ell, r) \in \mathcal{R}\}$ of all left–hand sides of $\mathcal{R}$ is denoted by $\mathit{dom}(\mathcal{R})$. The set $\mathit{ran}(\mathcal{R})$ of all right–hand sides of $\mathcal{R}$ is defined analogously. Given $c = (\ell, r) \in \mathcal{R}$ and $s, t \in M$, we write $s \to_c t$ if $s = u\ell v$ and $t = urv$ for some $u, v \in M$. We write $s \to_\mathcal{R} t$ if $s \to_c t$ for some $c \in \mathcal{R}$. As usual, $\to_\mathcal{R}^+$ $(\to_\mathcal{R}^*)$ is the transitive (reflexive and transitive) closure of $\to_\mathcal{R}$ and $\leftrightarrow_\mathcal{R}^*$ is the reflexive, transitive, and symmetric closure of $\to_\mathcal{R}$. The pair $(u, v) \in M \times M$ is *confluent* (with respect to $\mathcal{R}$) if $u \to_\mathcal{R}^* w$ and $v \to_\mathcal{R}^* w$ for some $w \in M$. The TRS $\mathcal{R}$ 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 $u \to_\mathcal{R}^* v_2$ the pair (v_1, v_2) is confluent. The TRS $\mathcal{R}$ is *confluent* if $\mathcal{R}$ is confluent on all $u \in M$. The TRS $\mathcal{R}$ is *locally confluent* if for all $u, v_1, v_2 \in M$ with $u \to_\mathcal{R} v_1$ and $u \to_\mathcal{R} v_2$ the pair (v_1, v_2) is confluent. The TRS $\mathcal{R}$ is *terminating* if there does not exist an infinite chain $u_1 \to_\mathcal{R} u_2 \to_\mathcal{R} u_3 \to_\mathcal{R} \cdots$. If $\mathcal{R}$ is terminating then by Newman's lemma $\mathcal{R}$ is confluent iff $\mathcal{R}$ is locally confluent. A trace $u \in M$ 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 $u \in M$ that are irreducible with respect to $\mathcal{R}$ is denoted by $\mathrm{IRR}(\mathcal{R})$. The trace v is a *normalform* of u if $u \to_\mathcal{R}^* v$ and $v \in \mathrm{IRR}(\mathcal{R})$. The TRS $\mathcal{R}$ is *length–reducing* if $|\ell| > |r|$ for all $(\ell, r) \in \mathcal{R}$. Obviously, if $\mathcal{R}$ is length reducing then $\mathcal{R}$ is terminating. The TRS $\mathcal{R}$ is *monadic* if $\mathcal{R}$ is length–reducing and $\mathit{ran}(\mathcal{R}) \subseteq \{1\} \cup \Sigma$. The TRS $\mathcal{R}$ is *special* if $\mathit{ran}(\mathcal{R}) = \{1\}$ and $1 \notin \mathit{dom}(\mathcal{R})$. Let $\mathrm{COLR}(M)$ $(\mathrm{COMO}(M), \mathrm{COSP}(M))$ 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 $\mathcal{R} \in \mathcal{C}$ and $u, v \in M$, 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 $\mathcal{R}$. If $I \neq (\Sigma \times \Sigma) \backslash \mathrm{Id}_\Sigma$ then 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 in this case we define $\|\mathcal{R}\| = \sum\{(|\ell| + |r|) \mid (\ell, r) \in \mathcal{R}\}$. But if $I = (\Sigma \times \Sigma) \backslash \mathrm{Id}_\Sigma$, i.e., if $\mathcal{R}$ is a VRS we can code $\mathcal{R}$ more efficiently by using the binary notation. Therefore in this case we define $\|\mathcal{R}\| = \sum\{bit(\underline{\ell}) + bit(\underline{r}) \mid (\underline{\ell}, \underline{r}) \in \mathcal{R}\}$. In this paper we always assume that a TRS $\mathcal{R}$ is represented as a string of length $\Omega(\|\mathcal{R}\|)$.

3 Semi–Thue systems

For terminating STSs confluence is decidable [BO81]. This classical result is based on the *critical pairs* of a STS. Let $\mathcal{R}$ be a STS over Σ. The set of *critical pairs* $\mathrm{CP}(\mathcal{R})$ contains exactly all pairs of the form (i) (sr_1t, r_2) where $(\ell_1, r_1), (s\ell_1t, r_2) \in \mathcal{R}$ and (ii) (r_1u, sr_2) where $(st, r_1), (tu, r_2) \in \mathcal{R}$ and $t \neq 1$. Note that $\mathrm{CP}(\mathcal{R})$ is finite. It is well known that $\mathcal{R}$ is locally confluent iff all critical pairs are confluent [NB72], which can be decided in the terminating case. For length–reducing STSs, confluence can be even decided in polynomial time [BO81,KKMN85]. In this section we prove that $\mathrm{COLR}(\{a,b\}^*)$ is moreover P–complete. Under reasonable assumptions from complexity theory this roughly means that the problem $\mathrm{COLR}(\{a,b\}^*)$ is inherently sequential.

Theorem 1. $\mathrm{COLR}(\{a,b\}^*)$ is P–complete.

Proof. The following problem is known to be P-complete [GHR95]: Given a deterministic Turing–machine $\mathcal{M}$, an input w for $\mathcal{M}$, and a word $t \in \{\#\}^*$, does $\mathcal{M}$ halt on w after $\leq |t|$ steps? Let $\mathcal{M} = (Q, \Sigma, \Box, \delta, q_0, q_f)$ be a deterministic Turing–machine, $w \in (\Sigma \backslash \{\Box\})^*$ be an input for $\mathcal{M}$, and $m \geq 0$. Here Q is the set of states, Σ is the tape alphabet ($Q \cap \Sigma = \emptyset$), $\Box \in \Sigma$ 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 final state. $\mathcal{M}$ halts iff it reaches the final state q_f. Let $\Sigma' = \{a' \mid a \in \Sigma\}$ be a disjoint copy of Σ with $\Sigma' \cap Q = \emptyset$. Let $\triangleright$ (left–end marker), $\triangleleft$ (right–end marker), A, and B be additional symbols and let $n = 3(m+1) + |w| + 2$. We define the length–reducing STS $\mathcal{R}$ over $\Gamma = Q \cup \Sigma \cup \Sigma' \cup \{\triangleright, \triangleleft, A, B\}$ by the following rules, where $a, b, c \in \Sigma$, $p, q \in Q$, $q \neq q_f$:

(1a) $q_f x \to q_f$ for all $x \in \Gamma$	(2a) $A^n B \to \triangleright q_0^{3(m+1)} w \triangleleft$
(1b) $x q_f \to q_f$ for all $x \in \Gamma$	(2b) $AB \to q_f$

(3a) $\alpha q^{3i} \triangleleft \to \alpha b' p^{3(i-1)} \triangleleft$	if $\delta(q, \Box) = (p, b, R)$, $1 \leq i \leq m+1$, $\alpha \in \Sigma' \cup \{\triangleright\}$
(3b) $\alpha q^{3i} a \to \alpha b' p^{3(i-1)}$	if $\delta(q, a) = (p, b, R)$, $1 \leq i \leq m+1$, $\alpha \in \Sigma' \cup \{\triangleright\}$
(3c) $a' q^{3i} \triangleleft \to p^{3(i-1)} ab \triangleleft$	if $\delta(q, \Box) = (p, b, L)$, $1 \leq i \leq m+1$
(3d) $\triangleright q^{3i} \triangleleft \to \triangleright p^{3(i-1)} \Box b \triangleleft$	if $\delta(q, \Box) = (p, b, L)$, $1 \leq i \leq m+1$
(3e) $c' q^{3i} a \to p^{3(i-1)} cb$	if $\delta(q, a) = (p, b, L)$, $1 \leq i \leq m+1$
(3f) $\triangleright q^{3i} a \to \triangleright p^{3(i-1)} \Box b$	if $\delta(q, a) = (p, b, L)$, $1 \leq i \leq m+1$

The rules (1a) and (1b) make q_f absorbing. The rules (3a) to (3f) 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 in order to make these rules length–reducing. Furthermore since $\mathcal{M}$ is deterministic, these rules do not generate critical pairs. The rule (2a) generates 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 (3a) to (3f). Note that $\mathcal{R}$ can be computed from $\mathcal{M}$, w, and $\#^m$ in deterministic logarithmic space. For this it is necessary that m is given in the unary representation $\#^m$ since $\|\mathcal{R}\|$ increases exponentially with $\mathit{bit}(m)$. We claim that $\mathcal{R}$ is confluent iff $\mathcal{M}$ halts on w after $\leq m$ steps.

If $\mathcal{M}$ does not halt on w after $\leq m$ steps then by simulating $m+1$ steps of $\mathcal{M}$ we obtain $A^nB \rightarrow_{(2a)} \triangleright q_0^{3(m+1)} w \triangleleft \rightarrow_{\mathcal{R}}^{m+1} \triangleright u'v \triangleleft \in \mathrm{IRR}(\mathcal{R})$ for some $u, v \in \Sigma^*$. Since also $A^nB \rightarrow_{(2b)} A^{n-1}q_f \rightarrow_{(1b)}^{n-1} q_f \in \mathrm{IRR}(\mathcal{R})$, $\mathcal{R}$ is not confluent. If $\mathcal{M}$ halts on w after $\leq m$ steps then $A^nB \rightarrow_{(2a)} \triangleright q_0^{3(m+1)} w \triangleleft \rightarrow_{\mathcal{R}}^{*} \triangleright u' q_f^{3j} v \triangleleft \rightarrow_{\mathcal{R}}^{*} q_f$ for some $j \geq 1$, $u, v \in \Sigma^*$. Hence the critical pair $(A^{n-1}q_f, \triangleright q_0^{3(m+1)} w \triangleleft)$ is confluent. In all other critical pairs one of the rules (1a) or (1b) is involved. Since q_f is absorbing these critical pairs are also confluent.

Now assume that $\Gamma = \{a_1, \ldots, a_k\}$ and let $\mathcal{P} = \{(\phi(\ell), \phi(r)) \mid (\ell, r) \in \mathcal{R}\}$, where the morphism $\phi : \Gamma^* \rightarrow \{a, b\}^*$ is defined by $\phi(a_i) = aba^{i+1}b^{k-i+2}$. Then (i) $\mathcal{P}$ is length–reducing and can be calculated from $\mathcal{R}$ in deterministic logarithmic space and (ii) $\mathcal{P}$ is confluent iff $\mathcal{R}$ is confluent, see [Loh99] for a proof of this fact. This proves the theorem. □

Using essentially the construction from the previous proof, the following result for the uniform word problem for the class of confluent and length–reducing STSs, which is known be in P by [Boo82], can be proven, see [Loh99].

Theorem 2. The uniform word problem for the class of confluent and length–reducing STSs over the alphabet $\{a, b\}$ is P–complete.

In contrast to the problem COLR($\{a, b\}^*$), which seems to be inherently sequential by Theorem 1, for the more restricted problem COMO(Σ^*) there exists an efficient parallel algorithm, see [Loh99] for the proof of the following theorem.

Theorem 3. COMO(Σ^*) is in NC^2 for every finite alphabet Σ.

This theorem follows from two facts: (i) The uniform word problem for ϵ-free context free grammars can solved in NC^2 [GHR95], p 176, and (ii) the problem whether a given pair of words is confluent with respect to a monadic STS can be reduced to the word problem for an ϵ-free context free grammar [BJW82].

4 Vector replacement systems

In [VRL98] it was shown that confluence is decidable but EXPSPACE–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 in the dimension n. The set $\mathrm{CP}(\mathcal{R})$ of critical pairs of $\mathcal{R}$ contains exactly all pairs $((s_1,\ldots,s_n),(t_1,\ldots,t_n))$ such that there exist rules $(k_1,\ldots,k_n)\to(p_1,\ldots,p_n)$ and $(\ell_1,\ldots,\ell_n)\to(r_1,\ldots,r_n)$ in $\mathcal{R}$ and for all $i\in\{1,\ldots,n\}$ it holds $s_i = \mathit{max}(k_i,\ell_i)-k_i+p_i$ and $t_i=\mathit{max}(k_i,\ell_i)-\ell_i+r_i$. Then $\mathcal{R}$ is locally confluent iff 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 vector $\underline{n}$ with respect to a length–reducing VRS may involve a number of steps that is exponential in $\mathit{bit}(\underline{n})$. Therefore the calculation of normalforms for the finitely many vectors that occur in the finitely many critical pairs does not lead to a polynomial time algorithm (as it is the case for STSs).

Theorem 4. $\bigcup_{k>0}\mathrm{COLR}(\mathbb{N}^k)$ is PSPACE–complete.

Proof. The following problem is known to be PSPACE–complete [Kar72]: Given a *deterministic linear bounded automaton* (briefly dlba) $\mathcal{M}$ and an input w for $\mathcal{M}$, does $\mathcal{M}$ accept w? Let us fix a dlba $\mathcal{M}=(Q,\Sigma,\rhd,\lhd,\delta,q_0,q_f)$ and an input $w\in(\Sigma\backslash\{\rhd,\lhd\})^*$ for $\mathcal{M}$, where Q is the finite set of states, Σ is the tape alphabet, $\rhd\in\Sigma$ is the left–end marker, $\lhd\in\Sigma$ 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. $\mathcal{M}$ accepts an input iff it finally reaches the final state q_f. The transition function must be defined such that (i) the read–write head never moves to the left (right) of $\rhd$ ($\lhd$) and (ii) does not overwrite $\rhd$ ($\lhd$) by a symbol different from $\rhd$ ($\lhd$) and (iii) does not overwrite a tape symbol $a\in\Sigma\backslash\{\rhd,\lhd\}$ by $\rhd$ or $\lhd$. We identify each tape cell of $\mathcal{M}$ with a number from $\{0,\ldots,|w|+1\}$, where cell 0 always contains $\rhd$ and cell $|w|+1$ always contains $\lhd$. We assume that $\mathcal{M}$ starts with the read–write head scanning cell 0 and that the read–write head is always in cell 0 if the final state q_f is reached.

We will construct a VRS $\mathcal{R}$ such that $\mathcal{R}$ is confluent iff $\mathcal{M}$ accepts the input w, which proves the theorem. Our construction is based on the simulation of a dlba by a Petri–net from [JLL77]. Let the alphabet Γ be

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

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 we assume that $\mathcal{M}$ terminates iff 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. Let $w=a_1a_2\cdots a_{|w|}$, $m=|Q|\cdot|\Sigma|^{|w|}\cdot(|w|+2)$, and $n=m+|w|+4$. The $|\Gamma|$–dimensional VRS $\mathcal{R}$

consists of the following rules, where $p, q \in Q$, $0 \leq i, j \leq |w| + 1$, and $a, b \in \Sigma$ (we use commutative words over Γ instead of vectors from $\mathbb{N}^{|\Gamma|}$ for the definition of $\mathcal{R}$ in order to improve readability):

(1a)	$\$(i,q)(i,a) \to (i+1,p)(i,b)$	if $\delta(q,a) = (p,b,R)$, $q \neq q_f$, $i \leq |w|$
(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)	$A^n \to \$^m(0,q_0)(0,\triangleright)(1,a_1)\cdots(|w|,a_{|w|})(|w|+1,\triangleleft)$	
(4b)	$A^2 \to (0,q_f)$	

The rules (1a) and (1b) simulate $\mathcal{M}$ where the additional \$ on the left–hand side makes these rules length–reducing. Rule (2) makes $(0, q_f)$ absorbing. Critical pairs that result from the rules (1a) and (1b) can be resolved with the rules (3a) and (3b). 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 $\mathcal{R}$ consumes a \$, we have to make enough \$ available for the initial configuration. Since there are at most $m = |Q| \cdot |\Sigma|^{|w|} \cdot (|w| + 2)$ different configurations for $\mathcal{M}$, the dlba $\mathcal{M}$ either terminates after $\leq m$ steps or loops forever. Thus m many \$ suffice. Note that in the binary representation of $\mathcal{R}(\mathcal{M}, w)$ the m many \$ are represented by $O(ld(m)) = O(ld(|Q|) + |w| \cdot ld(|\Sigma|) + ld(|w|+2))$ many bits, which is polynomial in $|w|$ and the length of the description of $\mathcal{M}$. The same holds for the number $n = m + |w| + 4$ in the left–hand side of rule (4a), which is chosen such that (4a) is length–reducing. The proof that $\mathcal{R}$ is confluent iff $\mathcal{M}$ accepts w is similar to the proof of Theorem 1, see [Loh99] for the details. □

Whether confluence is also PSPACE–complete for VRSs in a sufficiently large but fixed dimension is left as an open question. Similarly to the semi–Thue case, also for VRSs the complexity of the confluence problem decreases for the monadic and special case, see [Loh99]:

Theorem 5. $\bigcup_{k>0} \mathrm{COMO}(\mathbb{N}^k)$ is in NP and $\bigcup_{k>0} \mathrm{COSP}(\mathbb{N}^k)$ is in P.

5 Special trace rewriting systems

In [NO88] a trace monoid M is presented such that COLR(M) is undecidable. This result was sharpened in [Loh98], where it was shown that COLR(M) is decidable iff M is free or free commutative. These results imply that in general TRSs do not have finitely many critical pairs (in contrast to STSs and VRSs). Furthermore these results motivate the question whether there exist restricted but non–trivial classes of (length–reducing) TRSs for which confluence is decidable. In particular, in [Die90], p 154, it was asked whether confluence is decidable for special TRSs. We answer this question positively in this section.

Theorem 6. COSP(M) is in P for every trace monoid M.

Proof. For special VRSs the statement of the theorem is contained in Theorem 5. Thus let $M = \mathbb{M}(\Sigma, I)$ be a trace monoid where $I \neq (\Sigma \times \Sigma)\backslash \mathrm{Id}_\Sigma$ and let $\mathcal{R}$ be a special TRS over M. Let NF be an algorithm that computes an arbitrary normalform $\mathrm{NF}(u, \mathcal{R})$ of a given input trace u with respect to $\mathcal{R}$. Consider the following algorithm that we call SPECIAL:

```
Input: A special TRS R over M(Σ, I)
    forall (p1 s q1, p2 s q2) ∈ dom(R) × dom(R) with p1 I p2, q1 I q2 do
        nf1 := NF(p1 q1, R); nf2 := NF(p2 q2, R);
        if nf1 ≠ nf2 then return " R not confluent" (∗)
        else
            u := nf1 (= nf2);
            forall a ∈ Σ with a I p2 s q1 or a I p1 s q2 do
                nf1 := NF(au, R); nf2 := NF(ua, R);
                if nf1 ≠ nf2 then return "R not confluent" (∗∗)
            endfor
    endfor
endfor
return "R confluent" (∗∗∗)
```

First we prove that $\mathcal{R}$ is not confluent if SPECIAL outputs "$\mathcal{R}$ not confluent". If SPECIAL executes line $(*)$ then there exist $p_1sq_1, p_2sq_2 \in dom(\mathcal{R})$ with $p_1 \, I \, p_2$, and $q_1 \, I \, q_2$. Furthermore there exists a normalform u_i of p_iq_i ($i \in \{1, 2\}$) with $u_1 \neq u_2$. But then $\mathcal{R}$ is indeed not confluent since $p_2p_1sq_1q_2 \to_\mathcal{R} p_2q_2 \to^*_\mathcal{R} u_2$ and $p_2p_1sq_1q_2 = p_1p_2sq_2q_1 \to_\mathcal{R} p_1q_1 \to^*_\mathcal{R} u_1$. Now assume that SPECIAL executes line $(**)$. Then $u_1 = u_2 = u$ but there exists an $a \in \Sigma$ such that either $a \, I \, p_2sq_1$ or $a \, I \, p_1sq_2$ 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 $a \, I \, p_2sq_1$. Then $p_2p_1sq_1aq_2 \to_\mathcal{R} p_2aq_2 = ap_2q_2 \to^*_\mathcal{R} au \to^*_\mathcal{R} v_1$ and $p_2p_1sq_1aq_2 = p_1p_2sq_1aq_2 = p_1ap_2sq_2q_1 \to_\mathcal{R} p_1aq_1 = p_1q_1a \to^*_\mathcal{R} ua \to^*_\mathcal{R} v_2$. 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 SPECIAL outputs "$\mathcal{R}$ confluent" in line $(***)$. By induction on the length of traces it suffices to prove for all $t \in M$ that $\mathcal{R}$ is confluent on t if $\mathcal{R}$ is confluent on all t' with $|t'| < |t|$. Thus, let $t \in M$ and assume that $\mathcal{R}$ is confluent on all t' with $|t'| < |t|$. We have to prove that all pairs (t_1, t_2) with $t \to^i_\mathcal{R} t_1$ and $t \to^j_\mathcal{R} t_2$ for some $i, j \geq 0$ are confluent. The case $i = 0$ or $j = 0$ is trivial. Assume for a moment that we have already considered all cases with $i = 1 = j$. Then we can apply the arguments from the 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 $s_i \to^*_\mathcal{R} s$ ($i \in \{1, 2\}$) for some $s \in M$. 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 $u \in M$. 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 $v \in M$ and the pair (t_1, t_2) is confluent. Thus, it suffices to consider arbitrary factorizations $t = u_1\ell_1v_1 = u_2\ell_2v_2$, where $\ell_1, \ell_2 \in dom(\mathcal{R})$, and to prove that the pair (u_1v_1, u_2v_2) is confluent. Lemma 1 applied to the identity $u_1\ell_1v_1 = u_2\ell_2v_2$ gives nine traces p_i, q_i, w_i, y_i ($i \in \{1,2\}$) and s such that (see also the diagram below)

- $\ell_1 = p_1sq_1,\ \ell_2 = p_2sq_2$,
- $u_1 = y_1p_2w_2,\ u_2 = y_1p_1w_1,\ v_1 = w_1q_2y_2,\ v_2 = w_2q_1y_2$,
- $t = y_1p_1w_1p_2sq_2w_2q_1y_2 = y_1p_2w_2p_1sq_1w_1q_2y_2$,
- $p_1\,I\,p_2,\ q_1\,I\,q_2,\ w_1\,I\,w_2,\ w_1\,I\,p_2sq_1,\ w_2\,I\,p_1sq_2$.

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 show that the pair $(y_1p_1w_1w_2q_1y_2,\ y_1p_2w_2w_1q_2y_2)$ is confluent. If $y_1 \neq 1$ or $y_2 \neq 2$ then for $t' = p_2w_2p_1sq_1w_1q_2$ it holds $|t'| < t$ and $t' \rightarrow_{\mathcal{R}} p_2w_2w_1q_2$, $t' = p_1w_1p_2sq_2w_2q_1 \rightarrow_{\mathcal{R}} p_1w_1w_2q_1$. Thus the pair $(p_1w_1w_2q_1,\ p_2w_2w_1q_2)$ is confluent which therefore also holds for the pair $(y_1p_1w_1w_2q_1y_2,\ y_1p_2w_2w_1q_2y_2)$. Thus it suffices to show that the pair $(p_1w_1w_2q_1,\ p_2w_2w_1q_2) = (w_2p_1q_1w_1, w_1p_2q_2w_2)$ is confluent. We have one of the situations that are considered in the outer **forall**–loop of SPECIAL. Since we assume that SPECIAL outputs "$\mathcal{R}$ confluent" we know that $p_iq_i \rightarrow^*_{\mathcal{R}} u$ ($i \in \{1,2\}$) for some $u \in M$. Hence $w_2p_1q_1w_1 \rightarrow^*_{\mathcal{R}} w_2uw_1$, $w_1p_2q_2w_2 \rightarrow^*_{\mathcal{R}} w_1uw_2$ 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 $w_1\,I\,p_2sq_1$ it follows $a\,I\,p_2sq_1$. Thus, $a \in \Sigma$ is one of the symbols that are considered in the inner **forall**–loop of SPECIAL. It follows $au \rightarrow^*_{\mathcal{R}} v$ and $ua \rightarrow^*_{\mathcal{R}} v$ for some $v \in M$. Thus

$$wuaw_2 \rightarrow^*_{\mathcal{R}} wvw_2 \quad \text{and} \quad w_1uw_2 = wauw_2 \rightarrow^*_{\mathcal{R}} wvw_2. \tag{1}$$

Next let us consider $t' = p_1wp_2sq_2w_2q_1 = p_1w\ell_2w_2q_1$ (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 $t' = p_2w_2p_1sq_1wq_2 = p_2w_2\ell_1wq_2$. Thus $t' \rightarrow_{\mathcal{R}} p_1ww_2q_1 = w_2p_1q_1w \rightarrow^*_{\mathcal{R}} w_2uw$, $t' \rightarrow_{\mathcal{R}} p_2w_2wq_2 = wp_2q_2w_2 \rightarrow^*_{\mathcal{R}} wuw_2$. Hence $w_2uw \rightarrow^*_{\mathcal{R}} x$, $wuw_2 \rightarrow^*_{\mathcal{R}} x$ for some $x \in M$ and

$$w_2uw_1 = w_2uwa \rightarrow^*_{\mathcal{R}} xa \quad \text{and} \quad wuaw_2 = wuw_2a \rightarrow^*_{\mathcal{R}} xa. \tag{2}$$

Finally since $wuaw_2 \rightarrow^*_{\mathcal{R}} wvw_2$ by (1) and $wuaw_2 \rightarrow^*_{\mathcal{R}} xa$ by (2) and $|wuaw_2| = |w_1uw_2| \leq |w_1p_1q_1w_2| < |p_1w_1p_2sq_2w_2q_1| = |t|$ (where the strict inequality follows from $\ell_2 = p_2sq_2 \neq 1$) it holds $wvw_2 \rightarrow^*_{\mathcal{R}} z$ and $xa \rightarrow^*_{\mathcal{R}} z$ for some $z \in M$. But then $w_1uw_2 \rightarrow^*_{\mathcal{R}} wvw_2 \rightarrow^*_{\mathcal{R}} z$ by (1) and $w_2uw_1 \rightarrow^*_{\mathcal{R}} xa \rightarrow^*_{\mathcal{R}} z$ by (2). Thus the pair (w_1uw_2, w_2uw_1) is confluent and the correctness of SPECIAL is proved.

Finally we have to show that SPECIAL runs in polynomial time. This follows from the following two facts: (i) For a fixed independence alphabet (Σ, I), the number of different factorizations $\ell = psq$ of a trace ℓ is bounded by a polynomial in $|\ell|$. 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 $\|\mathcal{R}\|$ [Die90] (in the algorithms in [Die90] the TRS $\mathcal{R}$ is not part of the input, but it is easy to see that they run also in the uniform case, where the TRS is part of the input, in polynomial time). $\square$

We should mention that we proved a slight generalization of Theorem 6 in the long version [Loh99] of this paper. One might ask, whether confluence can be decided also 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 $u = vw$ with $v \neq 1 \neq w$ and $v\, I\, w$. A set $L \subseteq \mathbb{M}(\Sigma, I)$ is connected if every $u \in L$ 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 a regular word language. This is just one of several 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 subsets of $\mathbb{M}(\Sigma, I)$ is the smallest class $\mathcal{C}$ that contains all finite subsets of $\mathbb{M}(\Sigma, I)$ and that is closed under (i) union, (ii) concatenation of two sets (where the concatenation of L_1 and L_2 is $L_1 L_2 = \{u_1 u_2 \mid u_1 \in L_1, u_2 \in L_2\}$) and (iii) the star–operator restricted to connected sets, i.e., if L belongs to $\mathcal{C}$ and is connected then also $L^* = \{u_1 u_2 \cdots u_n \mid n \geq 0, u_1, u_2, \ldots, u_n \in L\}$ belongs to $\mathcal{C}$. It is known that for two recognizable word languages $L_1, L_2 \subseteq \Sigma^*$ and a confluent and monadic STS $\mathcal{R}$ it can be decided whether there exist $u_1 \in L_1$, $u_2 \in L_2$ with $u_1 \leftrightarrow^*_{\mathcal{R}} u_2$ [BO85]. This decision problem is known as the *extended word problem*. For trace monoids the situation is quite different as the following theorem shows.

Theorem 7. 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: Given a recognizable language $L_2 \subseteq M$, 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 the alphabet $\{a, b\}$. Let $P = \{(s_1, t_1), \ldots, (s_n, t_n)\}$ be an instance of the PCP, where $s_i, t_i \in \{a, b\}^*$. Let $\{\overline{a}, \overline{b}\}$ be a copy of $\{a, b\}$ and let $\# \notin \{a, b, \overline{a}, \overline{b}\}$. Let $\Sigma = \{a, b, \overline{a}, \overline{b}, \#\}$ and define an independence relation I on Σ by $I = \{a, b\} \times \{\overline{a}, \overline{b}\} \cup \{\overline{a}, \overline{b}\} \times \{a, b\}$. Note that $\#$ is depentant from every symbol. 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 $L_2 = \{[s_i\#\overline{t}_i]_I \mid 1 \leq i \leq n\}^+$. By Ochmański's theorem L_1 and L_2 are recognizable. Let $\mathcal{R} = \{\# \to 1\}$ which is confluent. Now it is easy to see that the PCP P has a solution iff there exist $u_1 \in L_1, u_2 \in L_2$, and $v \in \mathrm{IRR}(\mathcal{R})$ with $u_1 \to^*_{\mathcal{R}} v$, $u_2 \to^*_{\mathcal{R}} v$. Since $\mathcal{R}$ is confluent, the last property holds iff $u_1 \leftrightarrow^*_{\mathcal{R}} u_2$. □

Acknowledgments I would like to thank Volker Diekert, Anca Muscholl, and Friedrich Otto for valuable comments.

References

[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, 1993.

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

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

[Die90] V. Diekert. *Combinatorics on Traces.* Number 454 in Lecture Notes in Computer Science. Springer, Berlin-Heidelberg-New York, 1990.

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

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

[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 *Lecture Notes in Computer Science*, pages 319–330. Springer, 1998.

[Loh99] M. Lohrey. Complexity results for confluence problems. Technical Report 1999/05, University of Stuttgart, Germany, 1999. Available via ftp.informatik.uni-stuttgart.de/pub/library/ncstrl.ustuttgart_fi/TR-1999-05/.

[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.

[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.

[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, 1998.