# On the Confluence of Trace Rewriting Systems

Markus Lohrey

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

**Abstract.** In [NO88], a particular trace monoid $M$ is constructed such that for the class of length–reducing trace rewriting systems over $M$, confluence is undecidable. In this paper, we show that this result holds for every trace monoid, which is neither free nor free commutative. Furthermore we will present a new criterion for trace rewriting systems that implies decidability of confluence.

## 1 Introduction

The theory of *free partially commutative monoids* generalizes both the theory of free monoids and the theory of free commutative monoids. In computer science, free partially commutative monoids are commonly called *trace monoids* and their elements are called *traces*. Both notions are due to Mazurkiewicz [Maz77], who recognized trace monoids as a model of concurrent processes. [DR95] gives an extensive overview about current research trends in trace theory.

The relevance of trace theory for computer science can be explained as follows. Assume a finite alphabet $\Sigma$. An element of the free monoid over $\Sigma$, i.e., a finite word over $\Sigma$, may be viewed as the sequence of actions of a sequential process. In addition to a finite alphabet $\Sigma$, the specification of a trace monoid (over $\Sigma$) requires a binary and symmetric independence relation on $\Sigma$. If two symbols $a$ and $b$ are independent then they are allowed to commute. Thus, the two words $sabt$ and $sbat$, where $s$ and $t$ are arbitrary words, denote the same trace. This trace may be viewed as the sequence of actions of a concurrent process where the two independent actions $a$ and $b$ may occur concurrently and thus may be observed either in the order $ab$ or in the order $ba$.

This point of view makes it interesting to consider *trace rewriting systems*, see [Die90]. A trace rewriting system is a finite set of rules, where the left–hand and right–hand side of each rule are traces. Trace rewriting systems generalize both semi–Thue systems and vector replacement systems. Considered in the above framework of concurrent processes, a trace rewriting system may be viewed as a set of transformations that translate sequences of actions of one process into sequences of actions of another process. Thus, trace rewriting systems may for instance serve as a formal model of abstraction.

For all kinds of rewriting systems, the notion of a terminating system and the notion of a confluent system are of central interest. Together, these two properties guarantee the existence of unique normal forms. Unfortunately, both properties are undecidable even for the class of all semi–Thue systems. On the other hand it is a classical result that for the class of terminating semi–Thue systems, confluence is decidable. Unfortunately even this result does not hold in general if trace rewriting systems are considered. More precisely, in [NO88] a concrete trace monoid is presented such that for the class of length–reducing trace rewriting systems over this trace monoid, confluence is undecidable. Therefore it remains the problem to determine those trace monoids for which confluence is decidable for the class of terminating trace rewriting systems. This question will be solved in Section 4, where we prove that confluence of length–reducing systems is decidable only for free or free commutative monoids. This result will be obtained by a reduction to the case of trace monoids with three generators, see Section 3. This undecidability result leads to the question whether there exist (sufficiently large) subclasses of trace rewriting systems for which confluence becomes decidable, see [Die90] for such a subclass. In Section 5 we present a new criterion which implies decidability of confluence. Due to space limitations some proofs are only sketched or completely omitted. They can be found in [Loh98].

## 2 Preliminaries

In this section we will introduce some notions concerning trace theory. For a more detailed study, see [DR95]. The interval $\{1, \ldots, n\}$ of the natural numbers is denoted by $\overline{n}$. Given an alphabet $\Sigma$, the set of all finite words over $\Sigma$ is denoted by $\Sigma^*$. The empty word is denoted by 1. As usual, $\Sigma^+ = \Sigma^* \backslash \{1\}$. The length of $s \in \Sigma^*$ is denoted by $|s|$. For $\Gamma \subseteq \Sigma$ we define a *projection morphism* $\pi_\Gamma : \Sigma^* \longrightarrow \Gamma^*$ by $\pi_\Gamma(a) = a$ if $a \in \Gamma$ and $\pi_\Gamma(a) = 1$ otherwise. Given a word $s$ and factorizations $s = tlu = vmw$, we say that $t$ is *generated* by the occurrences of $l$ and $m$ in $s$ (that are uniquely defined by the two factorizations above) if $t = 1 = u$ or $v = 1 = w$ or ($l \neq 1 \neq m$ and ($t = 1 = w$ or $u = 1 = v$)). A *deterministic finite automaton*, briefly dfa, over $\Sigma$ is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where $Q$ is the finite set of states, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. The language $L(\mathcal{A}) \subseteq \Sigma^*$ that is accepted by $\mathcal{A}$ is defined as usual.

An *independence alphabet* is an undirected graph $(\Sigma, I)$, where $\Sigma$ is a finite alphabet and $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric relation, called *independence relation*. The complement $I^c = (\Sigma \times \Sigma) \backslash I$ is called a *dependence relation*. The pair $(\Sigma, I^c)$ is called a *dependence alphabet*. Given an independence alphabet $(\Sigma, I)$ 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 $s, t \in \Sigma^*$. Elements of $\mathbb{M}(\Sigma, I)$, i.e., equivalence classes of words, are called *traces*. The trace that contains the word $s$ is denoted by $[s]_{\equiv_I}$ or briefly $[s]$. For the rest of this section let $(\Sigma, I)$ be an arbitrary independence alphabet and let $M = \mathbb{M}(\Sigma, I)$. Since for all words $s, t \in \Sigma^*$, $s \equiv_I t$ implies $|s| = |t|$, we can define $|[s]| = |s|$. If $\Gamma \subseteq \Sigma$ then the trace monoid $N = \mathbb{M}(\Gamma, I \cap \Gamma \times \Gamma)$ is a submonoid of $M$ and we may view $\pi_\Gamma : \Sigma^* \longrightarrow \Gamma^*$ as a morphism $\pi_\Gamma : M \longrightarrow N$ between trace monoids. A *clique covering* of the dependence alphabet $(\Sigma, I^c)$ is a sequence $(\Sigma_i)_{i\in\overline{n}}$ of alphabets such that $\Sigma = \bigcup_{i=1}^n \Sigma_i$ and $I^c = \bigcup_{i=1}^n \Sigma_i \times \Sigma_i$. Given a clique covering $\Pi = (\Sigma_i)_{i\in\overline{n}}$, we will use the abbreviation $\pi_i = \pi_{\Sigma_i}$. Furthermore let $\pi_\Pi : M \longrightarrow \prod_{i=1}^n \Sigma_i^*$ be the morphism that is defined by $\pi_\Pi(u) = (\pi_i(u))_{i\in\overline{n}}$. It is well–known that $\pi_\Pi$ is injective. Thus, $M$ is isomorphic to its image under $\pi_\Pi$ which we denote by $\langle\Pi\rangle$. Its elements are also called *reconstructible* tuples, see [CM85] pp. 186. A necessary but not sufficient condition for $(s_i)_{i\in\overline{n}} \in \langle\Pi\rangle$ is $\pi_i(s_j) = \pi_j(s_i)$ for all $i, j \in \overline{n}$. If the cliques $\Sigma_1, \ldots, \Sigma_n$ are pairwise disjoint then $\pi_\Pi$ is also surjective and $M \simeq \prod_{i=1}^n \Sigma_i^*$. In the rest of this section let $\Pi = (\Sigma_i)_{i\in\overline{n}}$ be an arbitrary clique covering of the dependence alphabet $(\Sigma, I^c)$ and let $\pi = \pi_\Pi$. Given a factorization $u = vw$ of $u \in M$, we obtain a unique factorization $\pi(u) = \pi(v)\pi(w)$ of the reconstructible tuple $\pi(u)$ (where concatenation of tuples is defined component wise). But the converse is false. A factorization $\pi(u) = (s_i)_{i\in\overline{n}}(t_i)_{i\in\overline{n}}$ corresponds to a factorization of $u$ only if $(s_i)_{i\in\overline{n}} \in \langle\Pi\rangle$ (which implies $(t_i)_{i\in\overline{n}} \in \langle\Pi\rangle$). Since $\pi(u) \in \langle\Pi\rangle$, this already holds if the weaker condition $\pi_i(s_j) = \pi_j(s_i)$ holds for all $i, j \in \overline{n}$. Given a trace $l \in M$ and a factorization $\pi(u) = (s_i)_{i\in\overline{n}}\,\pi(l)(t_i)_{i\in\overline{n}}$, we say that the occurrence of $\pi(l)$ in $\pi(u)$ that is defined by this factorization is reconstructible if $\pi_i(s_j) = \pi_j(s_i)$ for all $i, j \in \overline{n}$. This implies $(s_i)_{i\in\overline{n}}, (t_i)_{i\in\overline{n}} \in \langle\Pi\rangle$ and thus the factorization above defines a unique occurrence of the trace $l$ in $u$.

In the following we introduce some notions concerning trace rewriting systems. For a more detailed study, see [Die90]. Let $\rightarrow$ be an arbitrary binary relation (for which we use infix notation) on an arbitrary set $A$. The reflexive and transitive closure of $\rightarrow$ is denoted by $\rightarrow^*$. The inverse relation $\rightarrow^{-1}$ of $\rightarrow$ is also denoted by $\leftarrow$. A $\rightarrow$*–merging* for $a, b \in A$ is a finite sequence of the form $a = a_1 \rightarrow \ldots \rightarrow a_n = b_m \leftarrow \ldots \leftarrow b_1 = b$. We say that the situation $a \leftarrow c \rightarrow b$ is $\rightarrow$*–confluent* if there exists a $\rightarrow$–merging for $a$ and $b$. The notion of a *terminating* (*confluent*, *locally confluent*) relation is defined as usual, see e.g. [BO93]. Newman's lemma states that a terminating relation is confluent iff it is locally confluent. A *trace rewriting system*, briefly TRS, over $M$ is a finite subset of $M \times M$. Let $\mathcal{R}$ be a TRS over $M$. An element $(l, r) \in \mathcal{R}$ is usually denoted by $l \rightarrow r$. Let $c = (l \rightarrow r) \in \mathcal{R}$. For $u, u' \in M$, we write $u \rightarrow_c u'$ if $u = vlw$ and $u' = vrw$ for some $v, w \in M$. We write $u \rightarrow_\mathcal{R} u'$ if $u \rightarrow_d u'$ for some $d \in \mathcal{R}$. We say that the TRS $\mathcal{R}$ is *terminating* (*confluent*, *locally confluent*) if $\rightarrow_\mathcal{R}$ is terminating (confluent, locally confluent). For $i \in \overline{n}$ we define the rules $\pi_i(c)$ and $\pi(c)$ by $\pi_i(c) = (\pi_i(l) \rightarrow \pi_i(r))$ and $\pi(c) = ((\pi_i(l))_{i\in\overline{n}} \rightarrow (\pi_i(r))_{i\in\overline{n}})$. The TRSs $\pi_i(\mathcal{R})$ and $\pi(\mathcal{R})$ are defined in the obvious way. Note that $\pi(u) \rightarrow_{\pi(c)} \pi(v)$ ( $\pi(u) \rightarrow_{\pi(\mathcal{R})} \pi(v)$ ) need not necessarily imply $u \rightarrow_c v$ ($u \rightarrow_\mathcal{R} v$). The reason is that in the rewrite step $\pi(u) \rightarrow_{\pi(c)} \pi(v)$ a non reconstructible occurrence of $\pi(l)$ may be replaced by $\pi(r)$. We introduce the following notations. Given $(s_i)_{i\in\overline{n}}, (t_i)_{i\in\overline{n}} \in \langle\Pi\rangle$, we write $(s_i)_{i\in\overline{n}} \Rightarrow_{\pi(c)} (t_i)_{i\in\overline{n}}$ if for some $(u_i)_{i\in\overline{n}}, (v_i)_{i\in\overline{n}}$ it holds $s_i = u_i\pi_i(l)v_i$, $t_i = u_i\pi_i(r)v_i$, and $\pi_i(u_j) = \pi_j(u_i)$ for all $i, j \in \overline{n}$. We write $(s_i)_{i\in\overline{n}} \Rightarrow_{\pi(\mathcal{R})} (t_i)_{i\in\overline{n}}$ if $(s_i)_{i\in\overline{n}} \Rightarrow_{\pi(d)} (t_i)_{i\in\overline{n}}$ for some $d \in \mathcal{R}$. With these notations it is obvious that $u \rightarrow_c v$ iff $\pi(u) \Rightarrow_{\pi(c)} \pi(v)$ and $u \rightarrow_\mathcal{R} v$ iff $\pi(u) \Rightarrow_{\pi(\mathcal{R})} \pi(v)$. In particular, $\mathcal{R}$ is confluent iff $\Rightarrow_{\pi(\mathcal{R})}$ is confluent on $\langle\Pi\rangle$. $\mathcal{R}$ is called *length–reducing* if $|l| > |r|$ for every $(l \rightarrow r) \in \mathcal{R}$. Obviously, if $\mathcal{R}$

Rules for the absorbing symbol 0:
(1a) $(1, x0) \to (1, 0)$ for $x \in \Gamma$
(1b) $(1, 0x) \to (1, 0)$ for $x \in \Gamma$
(1c) $(c, 0) \to (1, 0)$

Rules for deleting non well–formed words:
(2a) $(1, \lhd y) \to (1, 0)$ for $y \in \Gamma \backslash \{\$\}$
(2b) $(1, \lhd \$ y) \to (1, 0)$ for $y \in \Gamma \backslash \{\$\}$
(2c) $(1, x \rhd) \to (1, 0)$ for $x \in \Gamma$

Main rules:
(3a) $(c, \rhd A^{|w|+2}) \to (1, \rhd q_0 w \lhd)$
(3b) $(c, B) \to (1, 0)$

Rules for shifting \$-symbols to the left:
(4a) $(1, a\beta\$\$) \to (1, a\$\beta)$ for $\beta \in \Sigma \cup \{\lhd\}$
(4b) $(1, a\$\beta\$\$) \to (1, a\$\$\beta)$ for $\beta \in \Sigma \cup \{\lhd\}$

Rules for simulating $\mathcal{M}$: Let $q, q' \in Q$ and $a' \in \Sigma \backslash \{\Box\}$.
(5a) $(1, q \lhd \$\$) \to (1, a'q'\lhd)$ if $\delta(q, \Box) = (q', a', R)$
(5b) $(1, bq \lhd \$\$) \to (1, q'ba'\lhd)$ if $\delta(q, \Box) = (q', a', L)$
(5c) $(1, qa\$\$) \to (1, a'q')$ if $\delta(q, a) = (q', a', R)$
(5d) $(1, bqa\$\$) \to (1, q'ba')$ if $\delta(q, a) = (q', a', L)$
(5e) $(1, \rhd qa\$\$) \to (1, \rhd q'\Box a')$ if $\delta(q, a) = (q', a', L)$

**Fig. 1.** The system $\mathcal{R}_w$. Let $a, b \in \Sigma$ be arbitrary.

is length–reducing, then $\mathcal{R}$ is also terminating. By the well–known critical pair lemma it is possible to construct a finite set of critical pairs for a terminating semi–Thue system, see [BO93] for more details. Therefore it is decidable whether a terminating semi–Thue system is confluent. In [NO88], a trace monoid $M$ is presented such that even for the class of length–reducing TRSs over $M$, confluence is not decidable. This problem will be considered in the next section for arbitrary trace monoids. More precisely, let CONFL($M$) be the following computational problem:
INPUT: A length–reducing TRS $\mathcal{R}$ over $M$ QUESTION: Is $\mathcal{R}$ confluent ?
For technical reasons we will also consider the problem $CONFL_{\neq 1}(M)$ which is defined in the same way, but where the input is a length–reducing TRS whose right–hand sides are all different from the empty trace 1.

## 3 Independence alphabets with three vertices

A trace monoid $M = \mathbb{M}(\Sigma, I)$ with $|\Sigma| = 2$ is either the free monoid $\{a, b\}^*$ or the free commutative monoid $\{a\}^* \times \{b\}^*$. In both cases CONFL($M$) is decidable. If $|\Sigma| = 3$ then there exist up to isomorphism two cases, where $M$ is neither free nor free commutative. The first case arises from the independence alphabet that is defined by the graph $[a - c - b]$ and will be considered in this section. The corresponding trace monoid is $\{a, b\}^* \times \{c\}^*$. The second case arises from the independence alphabet $[a - b \quad c]$ and is considered in [Loh98].

**Lemma 1.** $\mathrm{CONFL}_{\neq 1}(\{a, b\}^* \times \{c\}^*)$ is undecidable.

*Proof.* First we prove the undecidability of $\mathrm{CONFL}_{\neq 1}(\Gamma^* \times \{c\}^*)$ for a finite alphabet $\Gamma = \{a_1, \ldots, a_n\}$ where $n > 2$. This alphabet $\Gamma$ can be encoded into the alphabet $\{a, b\}$ via the morphism $\phi : a_i \mapsto aba^{i+1}b^{n-i+2}$ for $i \in \overline{n}$. The following proof is a variant of a construction given in [NO88]. Let $\mathcal{M} = (Q, \Sigma, \Box, \delta, q_0, \{q_f\})$ be a deterministic one–tape Turing machine, where $Q$ is the finite set of states, $\Sigma$ is the tape alphabet, $\Box \in \Sigma$ is the blank symbol, $\delta : Q\backslash\{q_f\} \times \Sigma \to Q \times (\Sigma\backslash\{\Box\}) \times \{L, R\}$ is the transition function, $q_0$ is the initial state and $q_f \neq q_0$ is the final state. We may assume that $\delta$ is a total function. Thus, $\mathcal{M}$ terminates iff it reaches the final state $q_f$. Assume that the problem whether $\mathcal{M}$ halts on a given input $w \in (\Sigma\backslash\{\Box\})^+$ is undecidable. For instance, $\mathcal{M}$ may be a universal Turing machine. Let $\Gamma$ be the disjoint union $\Gamma = Q \cup \Sigma \cup \{0, \rhd, \lhd, A, B, \$\}$. For every $w \in (\Sigma\backslash\{\Box\})^+$, we define a TRS $\mathcal{R}_w$ over $\{c\}^* \times \Gamma^*$ by the rules of Figure 1. Note that $\mathcal{R}_w$ is length–reducing and that all right–hand sides are non empty. Since we excluded the case $w = 1$, we do not have to consider the pair $(1, \rhd q \lhd \$\$)$ in the last group of rules. We claim that $\mathcal{R}_w$ is confluent iff $\mathcal{M}$ does not halt on input $w$. Note that for every rule $(l_1, l_2) \to (r_1, r_2) \in \mathcal{R}_w$ it holds $r_i = 1$ if $l_i = 1$ for $i \in \{1, 2\}$. This property assures that $\mathcal{R}_w$ is confluent iff all situations $(t_1, t_2) \ {}_{\mathcal{R}_w}\!\!\leftarrow (s_1, s_2) \to_{\mathcal{R}_w} (t_1, t_2)$ are confluent, where the replaced occurrences of left–hand sides $(l_1, l_2), (m_1, m_2)$ in $(s_1, s_2)$ satisfy the following: For every $i \in \{1, 2\}$, $s_i$ is generated by the occurrences of $l_i$ and $m_i$ in $s_i$ and for some $i \in \{1, 2\}$ these two occurrences are non disjoint in $s_i$, i.e., have some letters in common. Most of these situations are easily seen to be confluent. For instance, in the situation $(1, Bv \rhd q_0 w \lhd) \ {}_{(3a)}\!\!\leftarrow (c, Bv \rhd A^{|w|+2}) \to_{(3b)} (1, 0v \rhd A^{|w|+2})$ (where $v \in \Gamma^*$ is arbitrary) both traces can be reduced to $(1, 0)$ since the left trace contains a factor of the form $x\rhd$ which may be rewritten to 0 with rule (2c). The only difficult situation is $(1, \rhd q_0 w \lhd vB) \ {}_{(3a)}\!\!\leftarrow (c, \rhd A^n vB) \to_{(3b)} (1, \rhd A^n v0)$, where $v \in \Gamma^*$ is arbitrary. Since $(1, \rhd A^n v0) \to^*_{(1a)} (1, 0)$, the truth of the following claim proves the lemma.

*Claim:* $\mathcal{M}$ does not halt on input $w$ iff $\forall v \in \Gamma^* : (1, \rhd q_0 w \lhd vB) \to^*_{\mathcal{R}_w} (1, 0)$.

Let $\mathcal{R}_\mathcal{M}$ be the subsystem that consists of the rules in group (4) and (5). First assume that $\mathcal{M}$ halts on input $w$. Then there exists an $m \geq 1$ such that $(1, \rhd q_0 w \lhd \$^m B) \to^*_{\mathcal{R}_\mathcal{M}} (1, \rhd u q_f v_1 \$^2 v_2 \$^2 \ldots v_{l-1} \$^2 v_l \$^2 \lhd \$^k B)$, where $u \in \Sigma^*, l \geq 0, v_1, \ldots, v_l \in \Sigma$ and $k \geq 2$. Since $\mathcal{M}$ cannot move from the final state $q_f$, the last pair is irreducible. Now assume that $\mathcal{M}$ does not halt on input $w$. First consider the case $v = \$^m$ for $m \geq 0$. We obtain

$$(1, \rhd q_0 w \lhd \$^m B) \to^*_{\mathcal{R}_\mathcal{M}} (1, \rhd u q v_1 \$^{\alpha_1} v_2 \$^{\alpha_2} \ldots v_{l-1} \$^{\alpha_{l-1}} v_l \$^{\alpha_l} \lhd \$^{\alpha_{l+1}} B),$$

where $u \in \Sigma^*, l \geq 0, v_1, \ldots, v_l \in \Sigma$, and $\alpha_1, \ldots, \alpha_{l+1} \in \{0, 1\}$. By rule (2a) or rule (2b) the last pair can be rewritten to $(1, \rhd u q v_1 \$^{\alpha_1} v_2 \$^{\alpha_2} \ldots v_{l-1} \$^{\alpha_{l-1}} v_l \$^{\alpha_l} 0)$ which reduces to $(1, 0)$ with the rules of group (1). Now assume $v = \$^m y v'$, where $m \geq 0, y \in \Gamma\backslash\{\$\}$ and $v' \in \Gamma^*$. Similarly to the derivation above, we obtain

$$(1, \rhd q_0 w \lhd \$^m y v' B) \to^*_{\mathcal{R}_\mathcal{M}} (1, \rhd u q v_1 \$^{\alpha_1} v_2 \$^{\alpha_2} \ldots v_{l-1} \$^{\alpha_{l-1}} v_l \$^{\alpha_l} \lhd \$^{\alpha_{l+1}} y v' B),$$

where $u \in \Sigma^*, l \geq 0, v_1, \ldots, v_l \in \Sigma$, and $\alpha_1, \ldots, \alpha_{l+1} \in \{0, 1\}$. Since $y \in \Gamma\backslash\{\$\}$ again either by rule (2a) or rule (2b) the last pair can be rewritten to $(1, \rhd u q v_1 \$^{\alpha_1} v_2 \$^{\alpha_2} \ldots v_{l-1} \$^{\alpha_{l-1}} v_l \$^{\alpha_l} 0 v' B)$ which reduces to $(1, 0)$ with the rules of group (1). This concludes the proof.

Note that the system $\mathcal{R}_w$ is not length–increasing in both components, i.e., for all $(l_1, l_2) \to (r_1, r_2) \in \mathcal{R}_w$, $|l_1| \geq |r_1|$ and $|l_2| \geq |r_2|$. Together with Theorem 2 of Section 5 this gives a very sharp borderline between decidability and undecidability for the case of a direct product of free monoids. The following result can be shown using similar techniques.

**Lemma 2.** $\mathrm{CONFL}_{\neq 1}(\mathbb{M}(\{a,b,c\}, \{(a,b),(b,a)\}))$ is undecidable.

## 4 The general case

A confluent semi–Thue system remains confluent if we add an additional symbol (that does not appear in the rules) to the alphabet. This trivial fact becomes wrong for TRSs, see [Die90], pp. 125 for an example. Thus, the following lemma is not a triviality.

**Lemma 3.** Let $(\Sigma, I)$ be an independence alphabet and let $\Gamma \subseteq \Sigma$. Let $M = \mathbb{M}(\Sigma, I)$ and let $N = \mathbb{M}(\Gamma, I \cap \Gamma \times \Gamma)$. Thus, $N \subseteq M$. If $\mathrm{CONFL}_{\neq 1}(M)$ is decidable then $\mathrm{CONFL}_{\neq 1}(N)$ is also decidable.

*Proof.* Given a length–reducing TRS $\mathcal{P}$ over $N$ whose right–hand sides are all non empty, we will construct a length–reducing TRS $\mathcal{R}$ over $M$ whose right–hand sides are also all non empty such that $\mathcal{P}$ is confluent iff $\mathcal{R}$ is confluent. The case $\Gamma = \Sigma$ is trivial. Thus, let us assume that there exists a $0 \in \Sigma \backslash \Gamma$. Let $\mathcal{R} = \mathcal{P} \cup \{[ab] \to [0] \mid a \in \Sigma \backslash \Gamma \text{ or } b \in \Sigma \backslash \Gamma\}$. Note that $\mathcal{R}$ is length–reducing. Assume that $\mathcal{P}$ is confluent and consider a situation $u_1 \;{}_{\mathcal{R}}\!\!\leftarrow u \to_{\mathcal{R}} u_2$. If $u \in N$ then we must have $u_1 \;{}_{\mathcal{P}}\!\!\leftarrow u \to_{\mathcal{P}} u_2$. Confluence of $\mathcal{P}$ implies that $u_1 \to_{\mathcal{P}}^* v \;{}_{\mathcal{P}}^{*}\!\!\leftarrow u_2$ for some $v \in N$ and thus $u_1 \to_{\mathcal{R}}^* v \;{}_{\mathcal{R}}^{*}\!\!\leftarrow u_2$. If $u \notin N$ then $u$ must contain some letter from $\Sigma \backslash \Gamma$. This must also hold for $u_1$ and $u_2$. Furthermore $u_i \notin \Sigma \backslash (\Gamma \cup \{0\})$ for $i \in \{1,2\}$, which holds since all right–hand sides of $\mathcal{P}$ are non empty. Thus, $u_1$ and $u_2$ can both be reduced to $[0]$. Now assume that $\mathcal{R}$ is confluent and consider a situation $u_1 \;{}_{\mathcal{P}}\!\!\leftarrow u \to_{\mathcal{P}} u_2$. Thus, $u_1 \;{}_{\mathcal{R}}\!\!\leftarrow u \to_{\mathcal{R}} u_2$ and confluence of $\mathcal{R}$ implies $u_1 \to_{\mathcal{R}}^* v \;{}_{\mathcal{R}}^{*}\!\!\leftarrow u_2$ for some $v \in M$. Since symbols from $\Sigma \backslash \Gamma$ do not appear in $u_1$ or $u_2$ it follows $u_1 \to_{\mathcal{P}}^* v \;{}_{\mathcal{P}}^{*}\!\!\leftarrow u_2$.

Now we are able to prove our first main result.

**Theorem 1.** $\mathrm{CONFL}(M)$ is decidable iff $M$ is a free monoid or a free commutative monoid.

*Proof.* The decidability of $\mathrm{CONFL}(M)$ in the case of a free or free commutative monoid is well–known. Thus, assume that $M = \mathbb{M}(\Sigma, I)$ is neither free nor free commutative. First note that $M$ is a direct product of free monoids iff the dependence alphabet $(\Sigma, I^c)$ is a disjoint union of complete graphs iff $(\Sigma, I)$ does not contain an induced subgraph of the form $[a{-}b \quad c]$. Thus, by Lemma 2 and Lemma 3, if $M$ is not a direct product of free monoids then $\mathrm{CONFL}_{\neq 1}(M)$ (and thus also $\mathrm{CONFL}(M)$) is undecidable. Thus, assume that $M = \prod_{i=1}^{n} \Sigma_i^*$. Since $M$ is neither free nor free commutative we have $n > 1$ and there exists an $i \in \overline{n}$ such that $|\Sigma_i| > 1$. But then $[a{-}c{-}b]$ is an induced subgraph of $(\Sigma, I)$. Lemma 1 and Lemma 3 imply the undecidability of $\mathrm{CONFL}_{\neq 1}(M)$ and $\mathrm{CONFL}(M)$.

## 5 A decidability criterion

In this section we present a new and non trivial criterion that implies decidability of confluence for terminating TRSs. For our considerations, we need the concept of a recognizable trace language. For a more detailed introduction into this topic, see for instance chapter 6 of [DR95]. One of the fundamental results about recognizable trace languages states that a trace language is recognizable iff it is recognized by a special kind of automaton, namely a so called asynchronous automaton, see [Zie87]. Since we will need only this type of automata, we use it for the definition of recognizable trace languages.

A (finite) *asynchronous automaton* $\mathcal{A}$ over the trace monoid $M = \mathbb{M}(\Gamma, I)$ is a tuple $\mathcal{A} = (Q, \Gamma, (\delta_a)_{a\in\Gamma}, q_0, F)$, where

- $Q = \prod_{i=1}^{m} Q_i$ is a direct product of finite sets of (local) states,
- for every symbol $a \in \Gamma$, there exists a non empty set $dom(a) \subseteq \overline{m}$ such that for all $(a, b) \in I$ it holds $dom(a) \cap dom(b) = \emptyset$,
- for every symbol $a \in \Gamma$, $\delta_a$ is a (partially defined) local transition function $\delta_a : \prod_{i\in dom(a)} Q_i \to \prod_{i\in dom(a)} Q_i$,
- $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states.

The (partially defined) global transition function $\delta : Q \times \Gamma \to Q$ of $\mathcal{A}$ is defined as follows. For $(p_i)_{i\in\overline{m}} \in Q$ and $a \in \Gamma$, $\delta((p_i)_{i\in\overline{m}}, a)$ is defined iff $\delta_a((p_i)_{i\in dom(a)})$ is defined. If this is the case it holds $\delta((p_i)_{i\in\overline{m}}, a) = (q_i)_{i\in\overline{m}}$, where (i) $q_i = p_i$ for $i \notin dom(a)$ and (ii) $(q_i)_{i\in dom(a)} = \delta_a((p_i)_{i\in dom(a)})$. For a word $s \in \Gamma^*$, $\delta((p_i)_{i\in\overline{m}}, s)$ is then defined in the usual way. Note that for $(a, b) \in I$ it holds $\delta((p_i)_{i\in\overline{m}}, ab) = \delta((p_i)_{i\in\overline{m}}, ba)$. Thus, it is possible to define $\delta((p_i)_{i\in\overline{m}}, [s])$ for $[s] \in M$. The language accepted by $\mathcal{A}$ is $L(\mathcal{A}) = \{u \in M \mid \delta(q_0, u) \in F\}$. A trace language $L \subseteq M$ is called *recognizable* iff there exists an asynchronous automaton $\mathcal{A}$ over $M$ with $L = L(\mathcal{A})$. Since it holds $L(\mathcal{A}) = M$ iff $\mathcal{A}$, considered as a dfa over $\Gamma$, accepts $\Gamma^*$ the following holds.

**Fact 1** The following problem is decidable.
INPUT: An asynchronous automaton $\mathcal{A}$ over $M$ QUESTION: $L(\mathcal{A}) = M$?

**Fact 2** If $K \subseteq M$ is recognizable and $L \subseteq N$ is recognizable then $K \times L \subseteq M \times N$ is also recognizable (use the well–known product construction).

The following lemma is crucial for the proof of the next theorem.

**Lemma 4.** Let $\mathcal{R}$ be a terminating TRS over $\mathbb{M}(\Sigma, I)$ and let $\Pi = (\Gamma_i)_{i\in\overline{n}}$ be a clique covering of $(\Sigma, I^c)$ such that

- For all $i \in \overline{n}$ and all $(l \to r) \in \mathcal{R}$ it holds $\pi_i(l) \neq 1$.
- For all $i, j \in \overline{n}$ and all $(l \to r) \in \mathcal{R}$, if $\pi_i(\pi_j(l)) = 1$ then also $\pi_i(\pi_j(r)) = 1$.

Then $\Rightarrow_{\pi_\Pi(\mathcal{R})}$ is confluent on $\langle \Pi \rangle$ (and thus $\to_{\mathcal{R}}$ is confluent) iff all situations

$$(t'_i)_{i\in\overline{n}} = (s_i\pi_i(p)s'_i)_{i\in\overline{n}} \;{}_{\pi_\Pi(\mathcal{R})}\!\Leftarrow\; (s_i\pi_i(l)s'_i)_{i\in\overline{n}} = (t_i)_{i\in\overline{n}} = (u_i\pi_i(m)u'_i)_{i\in\overline{n}} \Rightarrow_{\pi_\Pi(\mathcal{R})} (u_i\pi_i(r)u'_i)_{i\in\overline{n}} = (t''_i)_{i\in\overline{n}} \quad (1)$$

with the following properties are $\Rightarrow_{\pi_\Pi(\mathcal{R})}$–confluent.

– $(l \to p), (m \to r) \in \mathcal{R}$ and $(t_i)_{i\in\overline{n}}, (s_i)_{i\in\overline{n}}, (u_i)_{i\in\overline{n}} \in \langle \Pi \rangle$.
– For all $i \in \overline{n}$, the occurrences of $\pi_i(l) \neq 1$ and $\pi_i(m) \neq 1$ in $t_i$ generate $t_i$.
– For some $i \in \overline{n}$, the occurrences of $\pi_i(l)$ and $\pi_i(m)$ are non disjoint in $t_i$.

The fact that the two replaced occurrences of $\pi(l)$ and $\pi(m)$ can be assumed to generate $t_i$ is easy to see. Non–disjointness of the occurrences can be assumed due to the two conditions imposed on $\mathcal{R}$. These conditions assure that if two disjoint and reconstructible occurrences of $\pi(l)$ and $\pi(m)$ exist in $(t_i)_{i\in\overline{n}} \in \langle \Pi \rangle$ and (say) $\pi(l)$ is replaced by $\pi(p)$ then the occurrence of $\pi(m)$ in the resulting tuple is still reconstructible. Let us state now our second main result.

**Theorem 2.** The following problem is decidable.
INPUT: A TRS $\mathcal{R}$ over a trace monoid $M = \mathbb{M}(\Sigma, I)$ such that there exists a clique covering $(\Sigma_i)_{i\in\overline{n}}$ of the dependence alphabet $(\Sigma, I^c)$ with the following properties:

> $\pi_i(\mathcal{R})$ is terminating for every $i \in \overline{n}$ and $\pi_i(\pi_j(l)) = 1$ implies $\pi_i(\pi_j(r)) = 1$ for all $i, j \in \overline{n}$ with $i \neq j$ and all $(l \to r) \in \mathcal{R}$.

QUESTION: Is $\mathcal{R}$ confluent?

Note that for instance the second condition on $\mathcal{R}$ trivially holds if $M = \prod_{i=1}^{n} \Sigma_i^*$ with the clique covering $(\Sigma_i)_{i\in\overline{n}}$. Another class to which the theorem may be applied are special TRSs (i.e. every rule has the form $l \to 1$) such that for every left–hand side $l$ and every clique $\Sigma_i$ it holds $\pi_i(l) \neq 1$. On the other hand, Theorem 2 cannot be applied to the system $\mathcal{R}_w$ from the proof of Lemma 1, since for instance the projection onto the first component is not terminating.

*Proof.* Let $\Pi = (\Sigma_i)_{i\in\overline{n}}$ be a clique covering of $(\Sigma, I^c)$ such that $\pi_i(\mathcal{R})$ is terminating for every $i \in \overline{n}$ and let $\pi = \pi_\Pi$. Thus, $\mathcal{R}$ must be terminating. Moreover, there cannot exist a rule $(1 \to \pi_i(r)) \in \pi_i(\mathcal{R})$, i.e., for every $(l \to r) \in \mathcal{R}$ and every $i \in \overline{n}$ it holds $\pi_i(l) \neq 1$ and therefore Lemma 4 applies to $\mathcal{R}$. Fix two rules $(l \to p), (m \to r) \in \mathcal{R}$ and let $l_i = \pi_i(l)$ and similarly for $p$, $m$, and $r$. Consider the situation shown in (1) in Lemma 4. The conditions imposed on (1) imply that for every $i \in \overline{n}$ either (1) $t_i = l_i w_i m_i$ for some $w_i \in \Sigma_i^*$ or (2) $t_i = m_i w_i l_i$ for some $w_i \in \Sigma_i^*$ or (3) $l_i$ and $m_i$ are non disjoint and generate $t_i$. Furthermore there must exist at least one $i \in \overline{n}$ such that case (3) holds. Thus, for every partition $\overline{n} = I_1 \cup I_2 \cup I_3$ with $I_3 \neq \emptyset$ we have to consider all $(t_i)_{i\in\overline{n}} \in \langle \Pi \rangle$ such that for all $i \in I_k$ case (k) holds. Fix such a partition $\overline{n} = I_1 \cup I_2 \cup I_3$ and let $I_{1,2} = I_1 \cup I_2$, $\Sigma^{(k)} = \bigcup_{i\in I_k} \Sigma_i$ for $k \in \{1, 2, 3\}$. Moreover, since for every $i \in I_3$ there exist only finitely many possibilities for the string $t_i$ we may also fix the tuple $(t_i)_{i\in I_3}$. After these two choices, the only unbounded component in the tuple $(t_i)_{i\in\overline{n}}$ is the reconstructible factor $(w_i)_{i\in I_{1,2}}$. Since the occurrences $(l_i)_{i\in\overline{n}}$ and $(m_i)_{i\in\overline{n}}$ must be reconstructible in $(t_i)_{i\in\overline{n}}$ it is easy to see that $\pi_j(t_i) = \pi_i(t_j) = 1$ for all $i \in I_1, j \in I_2$. Similarly for all $i \in I_3, j \in I_{1,2}$ it must hold $\pi_i(w_j) = 1$. Thus, we have to consider all tuples $(w_i)_{i\in I_{1,2}} \in M_1 \times M_2$, where $M_1 = \langle (\Sigma_i \backslash (\Sigma^{(2)} \cup \Sigma^{(3)}))_{i\in I_1} \rangle \simeq \mathbb{M}(\Sigma^{(1)} \backslash (\Sigma^{(2)} \cup \Sigma^{(3)}), I)$ and $M_2 = \langle (\Sigma_i \backslash (\Sigma^{(1)} \cup \Sigma^{(3)}))_{i\in I_2} \rangle \simeq \mathbb{M}(\Sigma^{(2)} \backslash (\Sigma^{(1)} \cup \Sigma^{(3)}), I)$. We will prove that the set of all tuples $(w_i)_{i\in I_{1,2}} \in M_1 \times M_2$ for which there exists a $\Rightarrow_{\pi(\mathcal{R})}$–merging for the corresponding tuples $(t'_i)_{i\in\overline{n}}$ and $(t''_i)_{i\in\overline{n}}$ (that are uniquely determined by the $w_i$ via the two choices made above) is a recognizable trace language, which proves the theorem by Fact 1.

Let $\mathcal{R}_k = \{(\pi_i(l))_{i\in I_k} \to (\pi_i(r))_{i\in I_k} \mid (l \to r) \in \mathcal{R}\}$ for $k \in \{1,2,3\}$ and let $\mathcal{R}_{1,2} = \{(\pi_i(l))_{i\in I_{1,2}} \to (\pi_i(r))_{i\in I_{1,2}} \mid (l \to r) \in \mathcal{R}\}$. Obviously, there exists a $\Rightarrow_{\pi(\mathcal{R})}$–merging for $(t'_i)_{i\in\overline{n}}$ and $(t''_i)_{i\in\overline{n}}$ iff there exist a $\Rightarrow_{\mathcal{R}_{1,2}}$–merging for $(t'_i)_{i\in I_{1,2}}$ and $(t''_i)_{i\in I_{1,2}}$ as well as a $\Rightarrow_{\mathcal{R}_3}$–merging for $(t'_i)_{i\in I_3}$ and $(t''_i)_{i\in I_3}$ such that both mergings can be combined to a $\Rightarrow_{\pi(\mathcal{R})}$–merging, i.e., both mergings have the same length, in the $k$–th step for some $(l \to r) \in \mathcal{R}$ the rules $(\pi_i(l))_{i\in I_{1,2}} \to (\pi_i(r))_{i\in I_{1,2}}$ and $(\pi_i(l))_{i\in I_3} \to (\pi_i(r))_{i\in I_3}$, respectively, are applied, and finally the two replaced reconstructible occurrences of $(\pi_i(l))_{i\in I_{1,2}}$ and $(\pi_i(l))_{i\in I_3}$ give a reconstructible occurrence of $\pi(l)$. Since $\mathcal{R}_3$ is terminating it is possible to construct all $\Rightarrow_{\mathcal{R}_3}$–mergings for the fixed tuples $(t'_i)_{i\in I_3}$ and $(t''_i)_{i\in I_3}$. Fix one of these mergings. Since a finite union of recognizable trace languages is again recognizable it suffices to prove that the set of all tuples $(w_i)_{i\in I_{1,2}} \in M_1 \times M_2$ such that there exists a $\Rightarrow_{\mathcal{R}_{1,2}}$–merging for $(t'_i)_{i\in I_{1,2}}$ and $(t''_i)_{i\in I_{1,2}}$ which can be combined with the fixed $\Rightarrow_{\mathcal{R}_3}$–merging to a $\Rightarrow_{\pi(\mathcal{R})}$–merging is recognizable.

Recall that $\pi_j(t_i) = \pi_i(t_j) = 1$ for all $i \in I_1, j \in I_2$. The properties of $\mathcal{R}$ imply that every tuple $(s_i)_{i\in I_{1,2}}$ that appears in a $\Rightarrow_{\mathcal{R}_{1,2}}$–merging for $(t'_i)_{i\in I_{1,2}}$ and $(t''_i)_{i\in I_{1,2}}$ also satisfies $\pi_j(s_i) = \pi_i(s_j) = 1$ for all $i \in I_1, j \in I_2$. But this implies that every $\Rightarrow_{\mathcal{R}_1}$–merging for $(t'_i)_{i\in I_1}$ and $(t''_i)_{i\in I_1}$ can be combined with every $\Rightarrow_{\mathcal{R}_2}$–merging for $(t'_i)_{i\in I_2}$ and $(t''_i)_{i\in I_2}$, assumed that both mergings fit to our chosen $\Rightarrow_{\mathcal{R}_3}$–merging. By Fact 2 it suffices to prove that the set of all tuples $(w_i)_{i\in I_1} \in M_1$ such that there exists a $\Rightarrow_{\mathcal{R}_1}$–merging for $(t'_i)_{i\in I_1} = (p_i w_i m_i)_{i\in I_1}$ and $(t''_i)_{i\in I_1} = (l_i w_i r_i)_{i\in I_1}$ that can be combined with our fixed $\Rightarrow_{\mathcal{R}_3}$–merging is recognizable (the corresponding statement for $M_2$ can be proven analogously).

In order to allow this combination only rather restricted $\Rightarrow_{\mathcal{R}_1}$–mergings for $(p_i w_i m_i)_{i\in I_1}$ and $(l_i w_i r_i)_{i\in I_1}$ are allowed. More precisely, let the $k$–th step (in the $\Rightarrow_{\mathcal{R}_3}$–part or the $\Leftarrow_{\mathcal{R}_3}$–part) in our fixed $\Rightarrow_{\mathcal{R}_3}$–merging be of the form $(s_i)_{i\in I_3} \Rightarrow_{\mathcal{R}_3} (s'_i)_{i\in I_3}$, where $s_i = u_i \pi_i(l) u'_i$, $s'_i = u_i \pi_i(r) u'_i$, $(l \to r) \in \mathcal{R}$. Then we have to consider exactly those $\Rightarrow_{\mathcal{R}_1}$–mergings for $(p_i w_i m_i)_{i\in I_1}$ and $(l_i w_i r_i)_{i\in I_1}$ such that the $k$–th step has the form $(s_i)_{i\in I_1} \Rightarrow_{\mathcal{R}_1} (s'_i)_{i\in I_1}$, where $s_i = v_i \pi_i(l) v'_i$, $s'_i = v_i \pi_i(r) v'_i$, and $\pi_j(v_i) = \pi_i(u_j)$ for all $i \in I_{1,2}$, $j \in I_3$. In particular, the rule from $\mathcal{R}_1$ that must be applied in each step of a $\Rightarrow_{\mathcal{R}_1}$–merging for $(p_i w_i m_i)_{i\in I_1}$ and $(l_i w_i r_i)_{i\in I_1}$ is fixed. For the further consideration we may therefore exchange its left– and right–hand side. Thus, we only have to deal with a fixed sequence of rules over $\langle(\Sigma_i \backslash \Sigma^{(2)})_{i\in I_1}\rangle$. For the following let $\Gamma_i = \Sigma_i \backslash \Sigma^{(2)}$ for $i \in I_1$ and let $\Gamma = \Sigma^{(1)} \backslash \Sigma^{(2)}$.

From the previous discussion it follows that in order to prove the theorem it suffices to prove that for given $\alpha \in \mathbb{N}$ (which is the length of the fixed sequence of rules over $\langle(\Gamma_i)_{i\in I_1}\rangle$), $(l_{i,k})_{i\in I_1}, (r_{i,k})_{i\in I_1} \in \langle(\Gamma_i)_{i\in I_1}\rangle$, $u_{i,j,k} \in \Gamma_i \cap \Sigma_j$ (that are fixed for the rest of this section) where $i \in I_1$, $j \in I_3$, $k \in \overline{\alpha + 1}$ the set of all tuples $(w_i)_{i \in I_1} \in M_1$ such that

$$\begin{aligned} &\exists (s_{i,1})_{i\in I_1}, \dots, (s_{i,\alpha+1})_{i \in I_1} \in \langle (\Gamma_i)_{i\in I_1}\rangle \; \forall k \in \overline{\alpha}, i \in I_1, j \in I_3 : \\ &\quad s_{i,1} = p_i w_i m_i, \quad s_{i,\alpha+1} = l_i w_i r_i, \quad s_{i,k} = v_{i,k} l_{i,k} v'_{i,k}, \\ &\quad s_{i,k+1} = v_{i,k} r_{i,k} v'_{i,k}, \quad (v_{i,k})_{i\in I_1} \in \langle (\Gamma_i)_{i\in I_1}\rangle, \quad \pi_j(v_{i,k}) = u_{i,j,k} \end{aligned} \tag{2}$$

is recognizable in $M_1$. Since $M_1$ is recognizable in $\langle (\Gamma_i)_{i\in I_1}\rangle$ and recognizable languages are closed under intersection it suffices to prove that set of all tuples $(w_i)_{i\in I_1} \in \langle (\Gamma_i)_{i\in I_1}\rangle$ such that (2) holds is recognizable. This will be proven in the rest of this section. We need the following notions.

For the rest of the section let $X_i$ for $i \in I_1$ be an infinite enumerable set of variable symbols with $X_i \cap X_j = \emptyset$ for $i \neq j$ and let $\bigcup_{i\in I_1} X_i = X$. We assume $X \cap \Gamma = \emptyset$. For all $i \in I_1$, let $x_i \in X_i$ be a distinguished variable. For $S \in (\Gamma \cup X)^*$ let $\mathrm{Var}(S)$ denote the set of all variables that appear in $S$. Let $\mathrm{Lin}_i$ denote the set of all words $S \in (\Gamma_i \cup X_i \backslash \{x_i\})^*$ such that every variable $x \in \mathrm{Var}(S)$ appears only once in $S$. We write $S \preceq_i T$ iff $S, T \in \mathrm{Lin}_i$ and the word $\pi_X(S) \in X_i^*$ is a prefix of $\pi_X(T)$. We write $S \simeq_i T$ iff $S \preceq_i T$ and $T \preceq_i S$. A *substitution* is a function $\tau : X \to \Gamma^*$ such that $\tau(x) \in \Gamma_i^*$ for all $x \in X_i$ and $i \in I_1$. The homomorphic extension of $\tau$ to the set $(\Gamma \cup X)^*$ that is defined in the obvious way is denoted by $\tau$ as well. For every $x \in X$ and $i \in I_1$, we introduce a new symbol $\pi_i(x)$. Let $\pi_i(X) = \{\pi_i(x) \mid x \in X\}$. Every $\pi_i$ may be viewed as a morphism $\pi_i : (X \cup \Gamma)^* \to (\pi_i(X) \cup \Gamma_i)^*$. For a substitution $\tau$ we define $\tau(\pi_i(x)) = \pi_i(\tau(x))$. A *system of word equations with regular constraints*, briefly SWE, is a finite set $\Delta$ that consists of *equations* $S = T$ with $S, T \in (\Gamma_i \cup X_i \cup \pi_i(X))^*$ for some $i \in I_1$ and *regular constraints* of the form $x \in L(\mathcal{A})$ where $x \in X_i \backslash \{x_i\}$ and $\mathcal{A}$ is a dfa over $\Gamma_i$ for some $i \in I_1$. A *solution* $\tau$ for $\Delta$ is a substitution $\tau : X \to \Gamma^*$ such that $\tau(S) = \tau(T)$ for all $(S = T) \in \Delta$ and $\tau(x) \in L(\mathcal{A})$ for all $(x \in L(\mathcal{A})) \in \Delta$. Let $L(\Delta) = \{(\tau(x_i))_{i\in I_1} \in \langle (\Gamma_i)_{i\in I_1}\rangle \mid \tau \text{ is a solution of } \Delta\}$.

In (2) we ask for the set of all $(w_i)_{i\in I_1} \in \langle (\Gamma_i)_{i\in I_1}\rangle$ such that $(p_i w_i m_i)_{i\in I_1}$ can be transformed into $(l_i w_i r_i)_{i\in I_1}$ by a fixed sequence of rules where furthermore the projections onto the alphabets $\Sigma_j$ ($j \in I_3$) of the prefixes that precede the replaced occurrences of the left–hand sides are fixed. In the simpler case of a fixed sequence of ordinary string rewrite rules over some $\Gamma_i$ ($i \in I_1$) it is easy to construct finitely many data of the form $S^{(i)} \simeq_i T^{(i)}$ such that $s \in \Gamma_i^*$ can be transformed into $t \in \Gamma_i^*$ by an application of that sequence iff for some of the constructed data $S^{(i)}, T^{(i)}$ and some substitution $\tau$ it holds $s = \tau(S^{(i)})$, $t = \tau(T^{(i)})$. If we further fix the projections onto every alphabet $\Sigma_j$ ($j \in I_3$) of the prefixes that precede the replaced occurrences of the left–hand sides then we have to add to the above data fixed values for some of the $\pi_{\Sigma_j}(x)$ ($j \in I_3$, $x \in \mathrm{Var}(S^{(i)})$). These additional data may be expressed as regular constraints for the variables in $\mathrm{Var}(S^{(i)})$. If we go one step further and consider the direct product $\prod_{i\in I_1} \Gamma_i^*$, instead of a free monoid, the only thing that changes is that we need for every component $i \in I_1$ data of the above form. Finally if we consider the situation in (2), we have to enrich the above data further since we have to guarantee that reconstructible occurrences of left–hand sides will be replaced. This can be achieved by adding a (synchronization) equation of the form $\pi_i(S_k^{(j)}) = \pi_j(S_k^{(i)})$ (where $i, j \in I_1$, $i \neq j$ and $S_k^{(i)} \preceq S^{(i)}$ for every $i \in I_1$) for the $k$–th rewrite step, which assures reconstructibility. These consideration lead to the following lemma that can be proven by induction on $\alpha \geq 0$.

**Lemma 5.** There exists a finite set $\mathcal{S}$ of SWEs (which can be constructed effectively), where every $\Delta \in \mathcal{S}$ has the form

$$p_i x_i m_i = S^{(i)},\ l_i x_i r_i = T^{(i)},\ \pi_i(S_k^{(j)}) = \pi_j(S_k^{(i)})\ (i, j \in I_1, k \in \overline{\alpha}),\ \mathcal{C} \quad (3)$$

(with $S_k^{(i)} \preceq S^{(i)} \simeq T^{(i)}$ for all $i \in I_1$, $k \in \overline{\alpha}$ and $\mathcal{C}$ being regular constraints) such that for $(w_i)_{i \in I_1} \in \langle (\mathit{\Gamma}_i)_{i \in I_1} \rangle$ it holds (2) iff $(w_i)_{i \in I_1} \in L(\Delta)$ for some $\Delta \in \mathcal{S}$.

Thus, in order to prove Theorem 2 it suffices to show that for a SWE $\Delta$ of the form (3) the set $L(\Delta)$ is recognizable. The next lemma proves the recognizability of the set of solutions of SWEs of a simpler form.

**Lemma 6.** Let $S^{(i)} \in \mathrm{Lin}_i$, $Y_i = \mathrm{Var}(S^{(i)})$, $Y = \bigcup_{i=1}^n Y_i$ and for all $i, j \in I_1$ with $i \neq j$ let $S^{(i)} = S_{j,1}^{(i)} \ldots S_{j,\alpha_{i,j}}^{(i)}$ be factorizations of $S^{(i)}$, where $\alpha_{i,j} = \alpha_{j,i}$. Let $\Delta$ be the SWE

$$x_i = S^{(i)},\ \ \pi_j(S_{j,k}^{(i)}) = \pi_i(S_{i,k}^{(j)}),\ \ y \in L(\mathcal{A}_y)\ \ (i \neq j \in I_1,\ k \in \overline{\alpha_{i,j}},\ y \in Y). \quad (4)$$

Then $L(\Delta) \subseteq \langle (\mathit{\Gamma}_i)_{i \in I_1} \rangle$ is a recognizable trace language.

*Proof.* Let $\mathcal{A}_x = (Q_x, \mathit{\Gamma}_i, \delta_x, q_x, \{p_x\})$ for $x \in Y_i$, $i \in I_1$. We may assume that for $x \neq y$ it holds $Q_x \cap Q_y = \emptyset$. From these automata we can easily construct a dfa (with 1–transitions) $\mathcal{A}^{(i)} = (Q^{(i)}, \mathit{\Gamma}_i, \delta^{(i)}, q^{(i)}, p^{(i)})$ that recognizes the language $\{\tau(S^{(i)}) \mid \forall x \in Y_i : \tau(x) \in L(\mathcal{A}_x)\}$. The dfa $\mathcal{A}^{(i)}$ reads the word $S^{(i)}$ from left to right but instead of reading a variable $x \in Y_i$ it jumps via a 1–transition into the initial state of the dfa $\mathcal{A}_x$. If it reaches the final state of $\mathcal{A}_x$ the dfa jumps back into the word $S^{(i)}$ at the position after the variable $x$. The idea is to modify the so called mixed product automaton (see [Dub86]) of the dfas $\mathcal{A}^{(i)}$ (which recognizes the language $\{(w_i)_{i \in I_1} \in \langle (\mathit{\Gamma}_i)_{i \in I_1} \rangle \mid \forall i \in I_1 : w_i \in L(\mathcal{A}^{(i)})\}$) such that it checks whether the additional equations $\pi_j(S_{j,k}^{(i)}) = \pi_i(S_{i,k}^{(j)})$ are satisfied for all $i, j \in I_1$, $k \in \overline{\alpha_{i,j}}$ with $i \neq j$. More formally, let $\mathcal{A} = (\prod_{i \in I_1} Q^{(i)}, \mathit{\Gamma}, (\delta_a)_{a \in \mathit{\Gamma}}, (q^{(i)})_{i \in I_1}, \{(p^{(i)})_{i \in I_1}\})$ be the asynchronous automaton, where $\mathit{dom}(a) = \{i \in I_1 \mid a \in \mathit{\Gamma}_i\}$ for every $a \in \mathit{\Gamma}$. The local transition function is defined as follows. For all $i, j \in I_1$, $k \in \overline{\alpha_{i,j}}$ with $i \neq j$ let $Q_{j,k}^{(i)} \subseteq Q^{(i)}$ be the set of states of $\mathcal{A}^{(i)}$ that corresponds to the subword $S_{j,k}^{(i)}$ of $S^{(i)}$ (which contains all states in $Q_x$ if $x \in \mathrm{Var}(S_{j,k}^{(i)})$). Then the idea is to allow an $a$–transition only if for all $i, j \in \mathit{dom}(a)$ with $i \neq j$ there exists a $k \in \overline{\alpha_{i,j}}$ such that the $i$–th component of the asynchronous automata currently is in a state from $Q_{j,k}^{(i)}$ and the $j$–th component of the asynchronous automaton is in a state from $Q_{i,k}^{(j)}$, i.e., both components are in the same layer. Thus, $\delta_a((q_i)_{i \in \mathit{dom}(a)})$

is defined for $(q_i)_{i \in dom(a)} \in \prod_{i \in dom(a)} Q^{(i)}$ iff for all $i, j \in dom(a)$ with $i \neq j$ there exists a $k \in \overline{\alpha_{i,j}}$ such that $q_i \in Q^{(i)}_{j,k}$ and $q_j \in Q^{(j)}_{i,k}$. If this is the case then $\delta_a((q_i)_{i \in dom(a)}) = (\delta^{(i)}(q_i, a))_{i \in dom(a)}$. It is easy to see that $L(\Delta) = L(\mathcal{A})$.

Now the proof of Theorem 2 can be completed by proving that every SWE $\Delta$ of the form (3) can be reduced to a finite set $\mathcal{S}$ of SWEs of the form (4) such that $L(\Delta) = \bigcup\{L(\Theta) \mid \Theta \in \mathcal{S}\}$. The rather technical proof of this fact is carried out in [Loh98].

## 6 Conclusion

We have shown that for the class of length–reducing trace rewriting systems over a given trace monoid $M$, confluence is decidable iff $M$ is free or free commutative. Thus, we have located the borderline between decidability and undecidability for this problem in terms of the underlying trace monoid. Furthermore we have presented a new criterion that implies decidability of confluence for terminating systems. Other interesting classes of systems for which it is an open question, whether confluence is decidable are special or monadic trace rewriting systems (which are defined analogously to the semi–Thue case, see [BO93]) as well as one–rule trace rewriting systems, for which confluence can be decided in almost all cases, see [WD95].

**Acknowledgments** I would like to thank Volker Diekert and the anonymous referees for valuable comments.

## References


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

[CM85] Robert Cori and Yves Métivier. Recognizable subsets of some partially abelian monoids. *Theoretical Computer Science*, 35:179–189, 1985.

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

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

[Dub86] Christine Duboc. Mixed procuct and asynchronous automata. *Theoretical Computer Science*, 48:183–199, 1986.

[Loh98] Markus Lohrey. On the confluence of trace rewriting systems. Available via http://inf.informatik.uni-stuttgart.de/ifi/ti/personen/Lohrey/98a.ps, 1998.

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

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

[WD95] Celia Wrathall and Volker Diekert. On confluence of one-rule trace-rewriting systems. *Mathematical Systems Theory*, 28:341–361, 1995.

[Zie87] Wiesław Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. — Informatique Théorique et Applications*, 21:99–135, 1987.