

Reliable Generalized and Context Dependent Commutation Relations ^{*}

Isabelle Biermann¹ and Brigitte Rozoy²

¹ CNRS URA 369, L.I.F.L., Université de Lille I, Bât. M3,
F-59655 Villeneuve D'Ascq Cedex, France

² CNRS URA 410, L.R.I., Université de Paris-Sud, Bât. 490,
F-91405 Orsay Cedex, France

Abstract. Trace theory was introduced to capture the behavior of 1-safe Petri nets, it is based on partial commutation relations where two adjacent letters are allowed to permute. More general nets need more general commutation relations: we consider generalized and context dependent ones. We give necessary conditions and sufficient conditions for such relations to be reliable (preserve recognizability) and we provide a semi-algorithm to compute the closure of a language.

1 Introduction

The notion of trace was introduced by Mazurkiewicz [12] in order to model concurrent processes. Trace theory has now been systematically investigated and has a well developed mathematical theory [7]. A trace can be seen as the set of all possible sequential observations of a concurrent process. More formally, a trace is an equivalent class for a congruence generated by the set of pairs (ab, ba) where (a, b) is in the so called independence relation. If two sequential observations are equivalent (belong to the same trace), it is possible to go from the first to the second one by basic steps, each step corresponding to the commutation of two consecutive actions. These commutations are called partial commutations.

But partial commutations were introduced to capture the behavior of 1-safe Petri nets, therefore they fail to model more general concurrent processes: they cannot represent the well known Producer-Consumer paradigm. The need for more general models linked with general Petri nets (Place-Transitions nets) led several authors to introduce more general commutation relations [4, 9].

Here we will consider two kinds of commutation relations: generalized relations and context dependent relations. In Mazurkiewicz traces, the elements of a commutation relation are of the form (ab, ba) where a and b are letters. Lazare in [10] presented the notion of generalized commutation relation where the elements of the relation are of the form (u, v) , u and v being two commutatively equivalent words. These relations extend the commutation from permuting two adjacent letters to a permutation of the letters of a word. In [3], the authors introduced the notion of context traces, these traces are build from a context

^{*} This work was achieved while the first author was at L.R.I., Université de Paris-Sud.

dependent commutation relation. In such relations, the elements are of the form (abc, acb) with a , b and c being letters, it represents the fact that two letters b and c are only allowed to commute when preceded by a (left) context a . Context traces have proved to be suitable to model the Producer-Consumer paradigm. Moreover, Baugot and Gastin [1] have shown that any partial order representable and right-cancellative congruence can be generated by a w -context dependent commutation relation (by w -context dependent, we mean that the context may be more than a letter, for example a word w).

The notion of recognizable language has proved to be an accurate one when confronted with real (bounded memory) machines. In trace theory, the recognizability of a set of traces is equivalent to the recognizability of the underlying set of words. This leads to the question: given a recognizable language of words and a commutation relation, is the closure of the language by the commutation relation still recognizable? For the partial commutation case, the problem (in relation with the star problem) has been intensively studied [6, 15, 13, 8].

Here we ask a slightly different question: given a commutation relation, is it reliable? That is: can we be sure that, for any recognizable language, the closure by the relation will still be recognizable?

In [10], Lacaze gives a sufficient condition on generalized commutation relations to be reliable, in section 3 we give a necessary one. But this kind of relations very quickly appear to be too "general" for more advanced work. Therefore the remaining of the paper is dedicated to the study of context dependent commutation relations. In section 4, we prove a necessary condition for these relations to be reliable, it is based on the banning of "carrying circuit". Intuitively a carrying circuit will allow a "crossing" letter to commute repetitively with a word, thus generating a classical problem of "counting" letters which is not compatible with recognizability. Section 5 gives a semi-algorithm to compute the closure of a recognizable language by a context dependent commutation relation. This procedure takes as input the relation and an automaton recognizing the language and, at each step, adds paths to the automaton. If the procedure stops, the language of the resulting automaton is the closure of the original language. Related work can be found in [14], where Métivier, Richome and Wacrenier use a procedure S that computes the closure of a recognizable language by a partial commutation relation. Section 6 uses the semi-algorithm introduced in section 5 to prove a sufficient condition.

In a more general framework, this work can be related to [5] where Clerbout and Roos give a characterization of the semi-commutation relations such that, for any recognizable language L , the set of words that can be derived from words of L is algebraic.

2 Definitions and Preliminaries

Let A be a finite alphabet, then $A^* = \bigcup_{i=0}^{\infty} A^i$ is the classical free monoid on A . If w is a word in A^* and a is a letter in A , then $|w|_a$ stands for the number of occurrences of the letter a in the word w . The empty word is denoted by ϵ .

Definition 1. A *commutation relation* over A is a finite set of pairs (u, v) with u and v in A^* and such that, for any a in A , $|u|_a = |v|_a$. Particular cases of commutation relations are:

- *generalized commutations* where R is symmetric,
- *context dependent commutations* where R is a generalized commutation relation and elements of R are of the form (abc, acb) with a, b and c in A .
- *semi-commutations* where elements of R are of the form (ab, ba) with a and b in A .
- *partial commutations* where R is a semi-commutation relation and R is a generalized commutation relation.

A equivalence relation \sim on A^* is a *congruence* if it satisfies $\forall u, v, w, w' \in A^*$, $u \sim v \Rightarrow ww' \sim v w'$. If R is a generalized commutation relation on A^* , the *congruence generated by R* is the symmetric, reflexive and transitive closure of $\{(ww', v w' | (u, v) \in R\}$. It is the smallest congruence containing R .

Let L be a language of A^* , the *closure* of L by a generalized commutation relation R , denoted $[L]_R$, is the set $\{w \in A^* | \exists w' \in L, w \sim_R w'\}$.

A *finite automaton* is a 5-tuple (A, Q, q_I, Q_F, δ) where A is an alphabet, Q is a finite set of states, $q_I \in Q$ is the initial state, $Q_F \subseteq Q$ is the set of final states, $\delta \subseteq Q \times A \times Q$ is the set of transitions.

A *path* of an automaton $\mathcal{A} = (A, Q, q_I, Q_F, \delta)$ is a finite sequence of transitions $(q_0, a_1, q_1), (q_1, a_1, q_2), \dots, (q_{i-1}, a_i, q_i), \dots, (q_{n-1}, a_n, q_n)$, n is the length of the path, $a_1 \dots a_n$ is the label of the path. The notation $p \xrightarrow{u} q$ means that there is a path from p to q with label u .

A word u is *accepted* by an automaton \mathcal{A} if there is a path from q_I to a state of Q_F labelled by u . The *language accepted by the automaton*, denoted $L(\mathcal{A})$, is the set of all the words accepted by \mathcal{A} . We also say that \mathcal{A} *recognizes* $L(\mathcal{A})$. A language L is *recognizable* if there exists a finite automaton which recognizes L . We denote by $Rec(A^*)$ the set of all recognizable languages of A^* . $Rec(A^*)$ is closed by intersection.

Definition 2. Let R be a generalized commutation relation over the alphabet A , R is *reliable* if $\forall L \in A^*$, $L \in Rec(A^*) \Rightarrow [L]_R \in Rec(A^*)$.

Let us immediately point out that there are no (non trivial) reliable partial or semi commutation relations. Indeed, as soon as (ab, ba) , $a \neq b$, belongs to the relation, the closure of the recognizable language $(ab)^*$ is not recognizable. Indeed, for partial commutations $[(ab)^*] = \{w \in \{a, b\}^* | |w|_a = |w|_b\}$ and for semi-commutations $[(ab)^*] = \{w \in \{a, b\}^* | \forall u, \exists v, uv = w, |u|_b \geq |u|_a\}$.

3 Generalized Commutations

In 1992, Lacaze [10] proved a sufficient condition for a generalized commutation relation to be reliable. The condition forces any two elements of the relation to have disjoint alphabets and the two words of any element not to have a total overlapping. This sufficient condition allows, thanks to a close watch over of the

overlappings, to guaranty recognizability. We will see at the end of section 6 that some commutation relations may be reliable without satisfying to this condition.

In order to give a necessary condition for a commutation relation to be reliable, we first need two lemmas.

Lemma 3 (Lothaire [11]). For any non empty words u and v in A^* the following conditions are equivalent:

- i) $uv = vu$
- ii) $\exists \alpha \in A^+, \exists i, j \in \mathbb{N}^+, u = \alpha^i$ and $v = \alpha^j$
- iii) $\exists l, m \in \mathbb{N}^+, u^l = v^m$

Lemma 4. For any u and v of A^* : $uv = vu \Leftrightarrow \{u^k v^k, k \in \mathbb{N}\} \in \text{Rec}(A^*)$.

Proof. If $uv = vu$, by lemma 3, $\exists \alpha \in A^+$ and $\exists i, j \in \mathbb{N}^+$ such that $u = \alpha^i$ and $v = \alpha^j$. Thus $\{u^k v^k, k \in \mathbb{N}\} = (\alpha^{i+j})^*$ is recognizable.

Now let N be the number of states of an automaton recognizing $\{u^k v^k, k \in \mathbb{N}\}$, let q_p be the state after reading u^p . After reading u^N , there exist l and m ($l \leq m$) such that $q_l = q_m$. If $j = m - l$, we get: $u^l (u^j)^* u^{N-m} v^N \subseteq \{u^k v^k, k \in \mathbb{N}\}$ and in particular: $u^{N-j} v^N \in \{u^k v^k, k \in \mathbb{N}\}$. This implies $\exists r \in [N - j, N]$: $u^{N-j} v^N = u^r v^r$ and $v^{N-r} = u^{r-N+j}$, thus by lemma 3, $uv = vu$. \square

Proposition 5. If R is a generalized commutation relation over A , then:

$$R \text{ reliable} \implies \forall (uv, vu) \in R, \exists p, q \in \mathbb{N}^+, \forall a \in A, p|u|_a = q|v|_a.$$

Proof. Let u and v be in A^* if $uv = vu$, by lemma 3, it is true. Otherwise u and v are nonempty. Let (uv, vu) be in R and consider $Z = [(uv)^*]_R \cap u^* v^*$. If R is reliable then Z is recognizable. We show that Z recognizable implies $\exists p, q \in \mathbb{N}^+, \forall a \in A, p|u|_a = q|v|_a$. We have $\{u^k v^k, k \in \mathbb{N}\} \subseteq Z \subseteq \{u^i v^j, i, j \in \mathbb{N}\}$. If $\{u^k v^k, k \in \mathbb{N}\} = Z$, by lemmas 4 and 3, it is true. Otherwise, $\exists i \neq 0, j$ such that $u^{i+j} v^j \in Z$ (the case $u^i v^{i+j} \in Z$ is symmetrical). Thus $\exists k$ such that $u^{i+j} v^j \sim_R (uv)^k$. Let $p = |i+j-k|$ and $q = |k-j|$, we have, $\forall a \in A, p|u|_a = q|v|_a$. Because u and v are non empty and $i \neq 0$, p and q are non zero. \square

This condition applies to cases that are too restricted, moreover finding more general conditions proves to be very tricky. That is why we turn now to more structured commutation relations: the context dependent commutations.

4 Context Dependent Commutations

We first define the notion of carrying circuit: intuitively, a carrying circuit will allow a letter to commute repetitively with a word.

Definition 6. Let RC be a context dependent commutation relation over the letters of an alphabet A . A subset CC of RC is a *carrying circuit* of RC if

$$\exists n \geq 1, \exists a_1, a_2, \dots, a_n \in A, \exists x \in A,$$

$$CC = \{(a_1 a_2 x, a_1 x a_2), \dots, (a_i a_{i+1} x, a_i x a_{i+1}), \dots, (a_n a_1 x, a_n x a_1)\}.$$

The letter x is named the *crossing letter* of CC .

Theorem 7. Let RC be a context dependent commutation relation over the letters of an alphabet A^* .

RC is reliable $\Rightarrow RC$ does not have any carrying circuit.

Proof. Suppose $CC = \{(a_1 a_2 x, a_1 x a_2), \dots, (a_n a_1 x, a_n x a_1)\}$ is a minimal (with respect to set inclusion) carrying circuit of RC . Consider $L = (a_1 \dots a_n x)^* \in Rec(A^*)$ and $Z = \{(a_1 \dots a_n)^p x^p, p \in \mathbb{N}\}$. Because CC is minimal, we have $\forall i, j (i \neq j) : a_i \neq x$ and $a_i \neq a_j$, thus $a_1 \dots a_n x \neq x a_1 \dots a_n$, and by lemma 4, $Z \notin Rec(A^*)$. We show that $Z = [L]_{RC} \cap (a_1 \dots a_n)^* x^*$, thus implying $[L]_{RC} \notin Rec(A^*)$. Z is trivially included in $(a_1 \dots a_n)^* x^*$ and, as CC is a carrying circuit, we have $(a_1 \dots a_n)^p x^p \sim (a_1 \dots a_n x)^p$, thus $Z \subseteq [L]_{RC}$. For the other direction, let u be in $[L]_{RC} \cap (a_1 \dots a_n)^* x^*$, then there exist k, l, m such that $(a_1 \dots a_n x)^k \sim u = (a_1 \dots a_n)^l x^m$. By hypothesis, for any $i \neq j$, we have $a_i \neq a_j$ and $a_i \neq x$, thus $|(a_1 \dots a_n x)^k|_{a_i} = k$, $|(a_1 \dots a_n)^l x^m|_{a_i} = l$, $|(a_1 \dots a_n x)^k|_{x} = k$ and $|(a_1 \dots a_n)^l x^m|_{x} = m$. Thus $l = k = m$ and $Z = [L]_{RC} \cap (a_1 \dots a_n)^* x^*$. \square

There exist some reliable context dependent commutation relations. For example consider a relation $\{(abc, acb), (acb, abc)\}$ with a, b, c different, this relation is reliable. In order to prove it, next section gives a procedure to compute the closure of a recognizable language by a commutation relation.

5 Semi-Algorithm Computing the Closure of a Language

The aim of this section is, given a context dependent commutation relation RC and a finite automaton \mathcal{A} , to build an automaton recognizing the closure of $L(\mathcal{A})$ by the congruence generated by RC . We will apply three transformations on the considered automaton, the two first are meant to prepare the automaton in such a way that, when applying the third one, we are sure that we are adding the right kind of words to the language.

In the following, we consider only *clean* context dependent commutation relations, by clean we mean irreflexive ((u, u) never belongs to the relation) and without any element of the form (aba, aab) or (aab, aba) . This restriction is not a real one because reflexivity adds nothing to the generated congruence and, insofar as we are interested in reliable relations and that (aba, aab) and (aab, aba) are carrying circuits.

5.1 Getting Ready

The method used to add words to the language recognized by the automaton is the following: we look for a path labelled by abc with (abc, acb) in the relation and we add a path labelled by acb . The main problem is to remain "inside" the closure of the set. For example, suppose (abc, acb) is in the relation and consider the case depicted in figure 1. We want to add the dashed arcs to the automaton in order to compute the closure, but then we will also add the path $\overline{q_1} \xrightarrow{\overline{a}} q_2 \xrightarrow{c} q'_3 \xrightarrow{b} q_4$ to the automaton and this is not correct. In order to deal with these situations

we will prepare the automaton such that any two arrows arriving on a state are labelled with the same letter, we will say that this automaton satisfies to the *ACD* property (acronym of Anti-Co-Determinism).

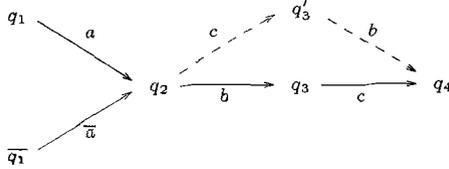


Fig. 1. Conflicting situation.

But first, we change the initial and the final states of the automaton, by adding two new states and linking the automaton to these two states by a new letter that will not be concerned by the commutations. This is the purpose of the Ω transformation.

Definition 8. Let $\mathcal{A} = (A, Q, q_I, Q_F, \delta)$ be a finite automaton, $\Omega(\mathcal{A}) = (A_\Omega, Q_\Omega, q_{I\circ}, \{q_{F\circ}\}, \delta_\Omega)$ is defined by
 $A_\Omega = A \cup \{\circ\}$ and $A \cap \{\circ\} = \emptyset$,
 $Q_\Omega = Q \cup \{q_{I\circ}, q_{F\circ}\}$,
 $\delta_\Omega = \delta \cup \{(q_{I\circ}, \circ, q_I)\} \cup \{(q, \circ, q_{F\circ}) \mid q \in Q_F\}$.

It is easy to check that $L(\Omega(\mathcal{A})) = \circ.L(\mathcal{A}).\circ$. Next transformation, Θ , turns the automaton in an automaton that satisfies to the *ACD* property (see figure 2).

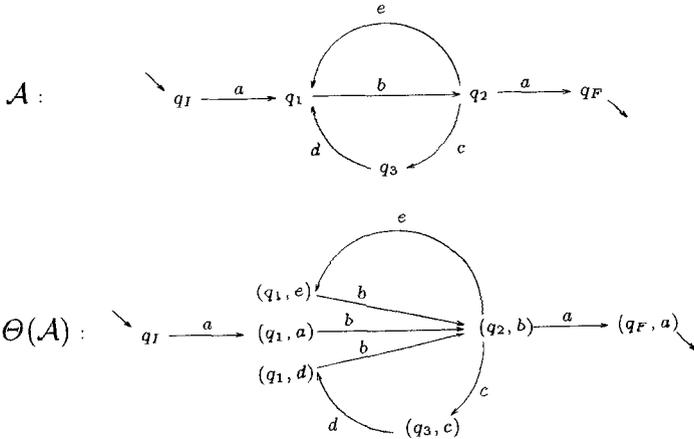


Fig. 2. Forcing the Anti-Co-Determinism: the Θ transformation.

Definition 9. Let $\mathcal{A} = (A, Q, q_I, Q_F, \delta)$ be a finite automaton such that $q_I \notin Q_F$ and no transition arrives onto q_I , $\Theta(\mathcal{A}) = (A, Q_\Theta, q_I, Q_{F\Theta}, \delta_\Theta)$ is given by:

$$Q_\Theta = \{q_I\} \cup (Q - \{q_I\}) \times A,$$

$$Q_{F\Theta} = Q_F \times A,$$

$$\delta_\Theta \text{ is defined by: } \begin{cases} \delta_\Theta(q_I, a) = \delta(q_I, a) \times \{a\}, \\ \delta_\Theta((q, a), b) = \delta(q, b) \times \{b\}, \text{ if } q \neq q_I. \end{cases}$$

We can easily check that $L(\Theta(\mathcal{A})) = L(\mathcal{A})$ and $L(\Theta(\mathcal{A}))$ has the *ACD* property. In fact, after applying Ω then Θ to an automaton the resulting automaton satisfies some specific properties that will allow us to add states and arrows to it while being sure that we stay “inside” the closure. We call such an automaton a \mathcal{P} -automaton.

Definition 10. \mathcal{B} is a \mathcal{P} -automaton if

- it satisfies *ACD* property,
- has a unique initial state q_I such that no transition arrives on it,
- has a unique final state q_F such that no transition comes out of it,
- there exists a letter \diamond such that any transition going out of q_I and any transition arriving on q_F is labelled by \diamond and no other transition is labelled by \diamond .

By definition, for any finite automaton \mathcal{A} , $\Theta(\Omega(\mathcal{A}))$, noted $\Theta \circ \Omega(\mathcal{A})$, is a \mathcal{P} -automaton and $L(\Theta \circ \Omega(\mathcal{A})) = \diamond.L(\mathcal{A}).\diamond$.

5.2 Adding Paths

Now we are ready to add paths to the automaton. In order to have a decision criterion where to add the paths we define two notions: a bridge and a potential bridge of an automaton.

Definition 11. Let \mathcal{A} be an automaton, RC a clean context dependent commutation relation on the alphabet of \mathcal{A} , $p_2, p_3, p'_3, p_4, p'_4$ be states of \mathcal{A} and b, c belong to the alphabet of \mathcal{A} .

- $(p_2, b, p_3, p'_3, c, p_4, p'_4)$ is a *bridge* of \mathcal{A} if
 - $p_2 \xrightarrow{b} p_3 \xrightarrow{c} p_4$ and $p_2 \xrightarrow{c} p'_3 \xrightarrow{b} p'_4$,
 - there exist states p_1 and p_5 and letters a and x such that: $p_1 \xrightarrow{a} p_2$, $p_4 \xrightarrow{x} p_5$, $p'_4 \xrightarrow{x} p_5$, and $(abc, acb) \in RC$,
 - for any state p and for any letter y : $p_4 \xrightarrow{y} p \Leftrightarrow p'_4 \xrightarrow{y} p$.
- (p_2, b, p_3, c, p_4) is a *potential bridge* of \mathcal{A} if
 - $p_2 \xrightarrow{b} p_3 \xrightarrow{c} p_4$,
 - there exist states p_1 and p_5 and letters a and x such that: $p_1 \xrightarrow{a} p_2$, $p_4 \xrightarrow{x} p_5$ and $(abc, acb) \in RC$,
 - for any state p'_3 and p'_4 : $p_2 \not\xrightarrow{c} p'_3$ or $p'_3 \not\xrightarrow{b} p'_4$ or there exist a state p and a letter y such that $p_4 \xrightarrow{y} p \not\Leftrightarrow p'_4 \xrightarrow{y} p$.

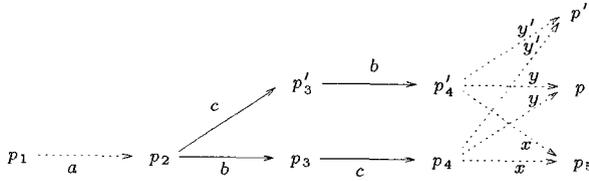


Fig. 3. A bridge.

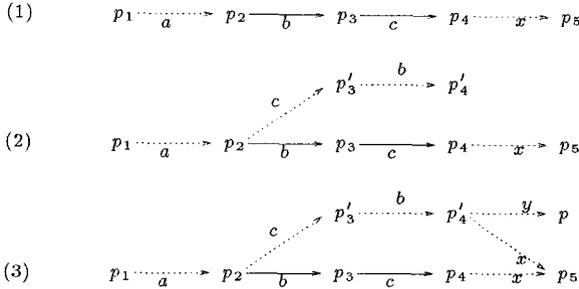


Fig. 4. Potential bridges.

Next transformation, Br , by turning a potential bridge into a bridge, adds some words equivalent to words of the original language. In order to choose the potential bridge on which Br is applied, we suppose the set of states and the alphabet are completely ordered inducing a total order on the potential bridges.

Definition 12. Let RC be a clean context dependent commutation relation over A , $\mathcal{A} = \langle A, Q, q_I, Q_F, \delta \rangle$ be a finite automaton with a minimal potential bridge (p_2, b, p_3, c, p_4) . $Br(\mathcal{A}) = \langle A, Q_{Br(\mathcal{A})}, q_I, Q_F, \delta_{Br(\mathcal{A})} \rangle$ is defined by:

$$Q_{Br(\mathcal{A})} = Q \cup \{p'_3, p'_4\}, \quad Q \cap \{p'_3, p'_4\} = \emptyset \text{ and } \forall q \in Q, q < p'_3 < p'_4$$

$$\delta_{Br(\mathcal{A})} = \delta \cup \{(p_2, c, p'_3)\} \cup \{(p'_3, b, p'_4)\} \cup \{(p'_4, x, p) \mid (p_4, x, p) \in \delta\}$$

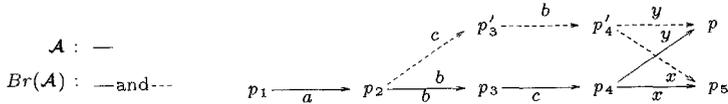


Fig. 5. The Br transformation.

The following lemma states that the resulting automaton still satisfies the wanted properties.

Lemma 13. Let \mathcal{B} be a \mathcal{P} -automaton, then $Br(\mathcal{B})$ is also a \mathcal{P} -automaton.

Proof. It is technical but not difficult to check that it follows from the hypothesis and the definition of Br (see [2] for details). \square

5.3 Computing the Closure

We have now a way to add paths to the automaton, we still must check that this will add the right kind of words to the language.

Proposition 14. *Let \mathcal{B} be a \mathcal{P} -automaton and RC a clean context dependent commutation relation on the alphabet of \mathcal{B} . Then: $L(Br(\mathcal{B})) \subseteq [L(\mathcal{B})]_{RC}$.*

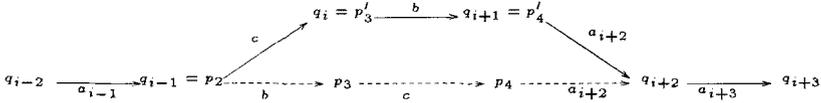


Fig. 6. Illustration of the proof of proposition 14.

Proof. Let (p_2, b, p_3, c, p_4) be the minimal potential bridge of \mathcal{B} and $q_I \xrightarrow{\circ} q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} q_n \xrightarrow{\circ} q_F$ be a path in $Br(\mathcal{B})$. Let i be the smallest index such that q_i does not belong to \mathcal{B} , then, with the notation used in the definition of Br , q_i is p'_3 , q_{i+1} is p'_4 , a_i is c , a_{i+1} is b and q_{i-1} is p_2 (see figure 6). The path $q_I \xrightarrow{\circ a_1 \dots a_{i-1}} q_{i-1} = p_2 \xrightarrow{b} p_3 \xrightarrow{c} p_4 \xrightarrow{a_{i+2}} q_{i+2}$ belongs to \mathcal{B} and $\diamond a_1 \dots a_{i-1} b c a_{i+2} \sim \diamond a_1 \dots a_{i-1} c b a_{i+2}$. Let now $j > i + 2$ be the next index such that q_j does not belong to \mathcal{B} . In the same way then for index i , we show that $\diamond a_1 \dots a_{i-1} c b a_{i+2} \dots a_{j-1} c b \sim \diamond a_1 \dots a_{i-1} b c a_{i+2} \dots a_{j-1} b c$. By iterating this process, looking always for the next “new state” of $Br(\mathcal{B})$, we show the existence of w the label of a path of \mathcal{B} such that $w \sim \diamond a_1 a_2 \dots a_n \diamond$. \square

The next corollary generalizes this proposition to any finite automaton prepared by the transformations Θ and Ω and to successive applications of the transformation Br .

Corollary 15. *Let \mathcal{A} be a finite automaton and RC a clean context dependent commutation relation, then: $\forall n \in \mathbb{N}, L(Br^n \circ \Theta \circ \Omega(\mathcal{A})) \subseteq \diamond.[L(\mathcal{A})]_{RC} \diamond$*

Proof. For any \mathcal{P} -automaton \mathcal{B} , $L(Br^n(\mathcal{B})) \subseteq [L(Br^{n-1}(\mathcal{B}))] \subseteq [L(Br^{n-2}(\mathcal{B}))] \subseteq \dots \subseteq [L(Br(\mathcal{B}))] \subseteq [L(\mathcal{B})]$. As $\Theta \circ \Omega(\mathcal{A})$ is a \mathcal{P} -automaton and $L(\Theta \circ \Omega(\mathcal{A})) = \diamond.L(\mathcal{A}).\diamond$, we have, for any n , $L(Br^n \circ \Theta \circ \Omega(\mathcal{A})) \subseteq \diamond.[L(\mathcal{A})]_{RC} \diamond$. \square

We show now that when the automaton has no more potential bridge, the language it recognizes is exactly the closure of the original language.

Theorem 16. *Let \mathcal{A} be an finite automaton and RC a clean context dependent commutation relation, then:*

$$\begin{aligned} \exists n \in \mathbb{N}, Br^n \circ \Theta \circ \Omega(\mathcal{A}) \text{ has no potential bridge} \\ \implies \\ L(Br^n \circ \Theta \circ \Omega(\mathcal{A})) = [L(\Theta \circ \Omega(\mathcal{A}))]_{RC}. \end{aligned}$$

Proof. We show that $[L(\Theta \circ \Omega(\mathcal{A}))]_{RC} \subseteq L(Br^n \circ \Theta \circ \Omega(\mathcal{A}))$. Let v be a word of $[L(\Theta \circ \Omega(\mathcal{A}))]$, there exist u in $L(\Theta \circ \Omega(\mathcal{A})) \subseteq L(Br^n \circ \Theta \circ \Omega(\mathcal{A}))$ such that $u \sim v$. Thus there exist $u_1 = u, u_2, \dots, u_p = v$ such that, for any $i < p$, there exists words γ, γ' , letters a, b and c such that $u_i = \gamma abc \gamma'$, $u_{i+1} = \gamma acb \gamma'$ and $(abc, acb) \in RC$. Suppose that $\gamma abc \gamma' \in L(Br^n \circ \Theta \circ \Omega(\mathcal{A}))$ and $(abc, acb) \in RC$, then there exists in $Br^n \circ \Theta \circ \Omega(\mathcal{A})$ a path $q_I \xrightarrow{\gamma} q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_3 \xrightarrow{c} q_4 \xrightarrow{x} q_5 \xrightarrow{\nu} q_F$ where x is a letter and ν a word such that $x\nu = \gamma'$. x exists because any path of $L(Br^n \circ \Theta \circ \Omega(\mathcal{A}))$ ends by label \diamond which is not concerned by the commutations. Because $L(Br^n \circ \Theta \circ \Omega(\mathcal{A}))$ has no potential bridge, there exists a path $q_2 \xrightarrow{cbx} q_5$ and $\gamma acb \gamma' \in L(Br^n \circ \Theta \circ \Omega(\mathcal{A}))$. By iterating this argument from $u = u_1$ to $u_p = v$, we conclude that, if v belongs to $[L(\Theta \circ \Omega(\mathcal{A}))]$, then there exists u in $L(\Theta \circ \Omega(\mathcal{A})) \subseteq L(Br^n \circ \Theta \circ \Omega(\mathcal{A}))$ such that $u \sim v$, thus $v \in L(Br^n \circ \Theta \circ \Omega(\mathcal{A}))$. \square

6 Application to Reliability

We have proved in section 4 that any reliable context dependent commutation relation does not have a carrying circuit. Example 1 shows what happens when a carrying circuit exists.

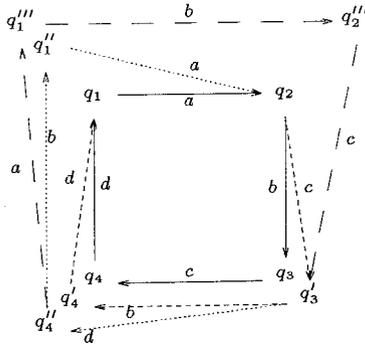


Fig. 7. When the process never ends (Example 1).

Example 1. Let $\{(abc, acb), (cbd, cdb), (dba, dab)\}$ be a carrying circuit of a relation RC , the crossing letter is b , consider an automaton with a loop $q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_3 \xrightarrow{c} q_4 \xrightarrow{d} q_1$ (see figure 7).

Suppose we start with the potential bridge (q_2, b, q_3, c, q_4) , we then add to the automaton states q'_3 and q'_4 and transitions $q_2 \xrightarrow{c} q'_3 \xrightarrow{b} q'_4 \xrightarrow{d} q_1$. This creates potential bridge (q'_3, b, q'_4, d, q_1) and we then add states q''_4 and q''_1 and transitions $q'_3 \xrightarrow{d} q''_4 \xrightarrow{b} q''_1 \xrightarrow{a} q_2$. By repeating this process one more time on the newly appeared potential bridge $(q''_4, b, q''_1, a, q_2)$, we can see that the external loop of

the automaton (path $q_4'' \xrightarrow{a} q_1''' \xrightarrow{b} q_2''' \xrightarrow{c} q_3' \xrightarrow{d} q_4''$) has exactly the same label then the starting loop, thus the process will be iterated infinitely many times.

Example 1 shows which kind of problems may arise when Br adds transitions that create new potential bridges in the automaton. In order to guarantee the end of the process, it is possible to forbid the creation of “new” potential bridges. The sufficient condition we present is based on this idea. Intuitively, if the letters that are contexts are different then the letters which commute, adding transition will never make appear a new potential bridge. Thus the number of applications of Br will be bounded by the number of potential bridges in the original automaton.

Proposition 17. *Let RC be a context dependent commutation relation, then:*

$$\{a \mid \{abc, acb\} \in RC\} \cap \{b \mid \{abc, acb\} \in RC\} = \emptyset \Rightarrow RC \text{ is reliable.}$$

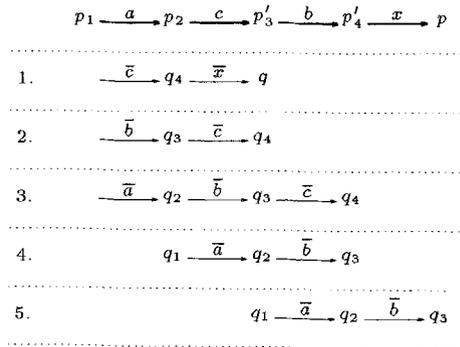


Fig. 8. Different possible cases for using the newly appeared transitions.

Proof. We first show that for any \mathcal{P} -automaton \mathcal{B} with minimal potential bridge $(p_2, \bar{b}, p_3, c, p_4)$, if $(q_2, \bar{b}, q_3, \bar{c}, q_4)$ is a potential bridge of $Br(\mathcal{B})$, then it was already a potential bridge of \mathcal{B} .

If $(q_2, \bar{b}, q_3, \bar{c}, q_4)$ was not a potential bridge of \mathcal{B} , then it uses one of the transitions added by Br , figure 8 summarizes the different possible cases (notations are the ones used in the definition of Br).

Because $Br(\mathcal{B})$ satisfies property ACD , case 1 implies that $a = \bar{c}$, case 2 that $a = \bar{b}$, case 4, $c = \bar{a}$, case 5, $b = \bar{a}$, which are impossible by hypothesis. Case 3 implies $a = \bar{a}$, $c = \bar{b}$, $b = \bar{c}$, $p_2 = q_2$, $p'_3 = q_3$ and $p'_4 = q_4$, this means that $(q_2, \bar{b}, q_3, p'_3, \bar{c}, q_4, p'_4)$ is a bridge of $Br(\mathcal{B})$. Thus $(q_2, \bar{b}, q_3, \bar{c}, q_4)$ cannot be a potential bridge of $Br(\mathcal{B})$.

The number of potential bridges of any finite automaton is bounded by $|Q| \times d^2$ where $|Q|$ is the number of states and d is the output degree of the automaton.

Thus for any recognizable language L , if \mathcal{A} is an automaton recognizing L , if N is the number of potential bridges of $\Theta \circ \Omega(\mathcal{A})$, $Br^N \circ \Theta \circ \Omega(\mathcal{A})$ has no

potential bridges. And, by theorem 16, $L(Br^N \circ \Theta \circ \Omega(\mathcal{A})) = \diamond.[L].\diamond$ and $[L]$ is recognizable. \square

If we compare the above obtained sufficient condition for a context dependent commutation relation to be reliable to the one proposed by Lacaze [10], we first remark that it seems less general because it applies only to context dependent commutation. But this is not the case, as in [10] any two elements of the relation have a different alphabet. Here we allow letters to appear in more than one commutation rule.

Acknowledgements The authors are indebted to A. Petit for fruitful discussions.

References

1. S. Bauget and P. Gastin. On congruences and partial orders. *Lecture Notes in Computer Science*, 969:434–443, 1995. MFCS'95.
2. I. Biermann. *Extensions structurelles des traces de commutation*. PhD thesis, Université Paris-Sud, Orsay, France, 1995.
3. I. Biermann and B. Rozoy. Context traces and transition systems. In *ISCIS IX*, Antalya, Turkey, 1994.
4. M. Clerbout and M. Latteux. Semi-commutations. *Information and Computation*, 73:59–74, 1987.
5. M. Clerbout and Y. Roos. Semi commutations and algebraic languages. *Theoretical computer science*, 103:39–49, 1992.
6. R. Cori and D. Perrin. Automates et commutations partielles. *RAIRO Theoretical Informatics and Applications*, 19:21–32, 1985.
7. V. Diekert and G. Rozenberg. *The book of traces*. World Scientific Publ. Co., Singapour, 1995.
8. P. Gastin, E. Ochmanski, A. Petit, and B. Rozoy. Decidability of the star problem in $a^* \times \{b\}^*$. *Information Processing Letters*, 44:65–71, 1992.
9. P. Hoogers, H.C.M. Kleijn, and P.S. Thiagarajan. A trace semantics for Petri nets. Technical Report 92-03, Leiden University, 1992. to appear in 1995 in *Information and Computation*.
10. J. Lacaze. Parties reconnaissables de monoïdes définis par générateurs et relations. *R.A.I.R.O. Informatique théorique et applications*, 26(6):541–552, 1992.
11. M. Lothaire. *Combinatorics on words*, volume 17 of *Encyclopedia of Mathematics*. Addison-Wesley, 1982.
12. A. Mazurkiewicz. Concurrent program schemes and their interpretations. Aarhus University, DAIMI Rep. PB 78, 1977.
13. Y. Métivier. Une condition suffisante de reconnaissabilité dans un monoïde partiellement commutatif. *R.A.I.R.O. Theoretical Informatics and Applications*, 20:121–127, 1986.
14. Y. Métivier, G. Richomme, and P. A. Wacrenier. Computing the closure of sets of words under partial commutations. *Lecture Notes in Computer Science*, 944:75–86, 1995. ICALP'95.
15. E. Ochmanski. Regular behaviour of concurrent systems. *Bulletin of EATCS*, 27:56–67, October 1985.