On congruences and partial orders*

Serge Bauget Paul Gastin

Institut Blaise Pascal, LITP

4, place Jussieu, F-75252 Paris Cedex 05

{Serge.Bauget,Paul.Gastin}@litp.ibp.fr

Abstract

Mazurkiewicz trace theory is not powerful enough to describe concurrency paradigms as, for instance, the "Producer / Consumer". We propose in this paper a generalization of Mazurkiewicz trace monoids which allows to model such problems. We consider quotients of the free monoids by congruences which preserve the commutative images of words. An equivalence class in the quotient monoid consists of all the sequential observations of a distributed computation. In order to characterize congruences which do model concurrency, we study the relationship of this approach and the classical representation of distributed computations with partial orders. We show that the only congruences for which the classes can be represented by partial orders and the concatenation transfers modularly to partial orders are congruences generated by commutations, that is trace congruences. We prove necessary conditions and sufficient conditions on congruences so that their classes can be represented by partial orders. In particular, an important sufficient condition covers both trace congruences and the "Producer / Consumer" congruence.

1 Introduction

In computer science, Mazurkiewicz introduced trace monoids in order to describe behaviors of distributed systems [Maz77]. In such systems, actions executed by different components may be completely independent. Thus, considering a distributed computation, different external observers may perceive independent actions in different orders. Therefore a distributed computation can be modelized by the collection of all its sequential observations, called a trace, which is precisely an equivalence class in some partially commutative monoid. If A is the alphabet which represents the possible actions and $I \subseteq A \times A$ is the symmetric and irreflexive relation which contains the independent letters, the trace monoid over (A, I) is the quotient of the free monoid A^* by the congruence generated by $\{(ab, ba) \mid (a, b) \in I\}$. For detailed presentations of traces, the reader is referred to surveys [Maz87, AR88, Per89], to the monograph [Die90] or to the Book of Traces [DR95].

*This research has been supported by the ESPRIT Basic Research Actions No. 6317 ASMICS II and No. 6067 CALIBAN.

However, trace theory is not powerful enough to describe the behaviors of all concurrency problems. For instance, the classical "Producer / Consumer" example cannot be described within this model. But on the other hand, the "Producer / Consumer" paradigm can all the same be described by the quotient of the free monoid $\{P, C\}^*$ by the congruence generated by the pair (PPC, PCP), where P and C stand for *produce* and *consume*. This congruence is different from a trace congruence because a commutation of P and C depends on a left context. In this paper, we will study congruences which allow us to describe distributed systems which cannot be represented in trace theory. In this way, an equivalence class will represent the sequential observations of a parallel computation. These observations are of course composed with the same actions but the order of actions may change. So we require that the congruences preserve the number of occurrences of each letter of a word. Recently, other works has adopted the same framework [Lac92, BR94].

The theory of semi-commutations [CL87, CLR95] provides another generalization of Mazurkiewicz traces in which, for instance, the "Producer / Consumer" example can be represented. However, the point of view of semi-commutation is fully different since it consists of a rewriting system which may be non symmetric. Yet another generalization of traces linked with Petri nets and which covers in particular the "Producer / Consumer" paradigm can be found in [HKT92].

Other well-known models of distributed systems are based on partial orders, see for instance [Win88, NW95]. In such approaches, the description does not insist on the possible observations of a behavior, but on the occurrences of its actions and on their causal relations described by the partial order. In trace theory, the two approaches are equivalent. The sequential observations of a distributed computation are precisely the linear extensions of its associated partial order. Conversely, the partial order associated with a distributed computation is the intersection of (the total orders defined by) its sequential observations. In our generalized theory, we require that a congruence preserves the number of occurrences of each letter of a word. Hence, we may still associate a partial order with a distributed computation: the intersection of the total orders of its sequential observations. However, in general, the set of linear extensions of the partial order may strictly contain the set of sequential observations. When modeling distributed systems, this situation is clearly undesirable. We say that a congruence is represented by a partial order if each equivalence class is equal to the set of linear extensions of the associated partial order. The aim of this paper is to characterize congruences which are represented by partial orders. In other words, we want to find the congruences which do model concurrency.

When a congruence $\equiv$ is represented by a partial order, one may transfer the structure of the quotient monoid $A^*/\equiv$ to the set of partial orders which represent classes. In Section 3 we characterize congruences for which the concatenation of partial orders is modular, that is, the concatenation of two partial orders may only add edges from the first one to the second one. Surprisingly, we prove that a congruence is represented by a partial order with a modular concatenation if and only if it is a trace congruence.

In the following, we are looking for characterizations of congruences which are represented by partial orders. In Section 4 we give some necessary conditions. In particular, we show that a right cancellative congruence which is represented by a partial order is necessarily generated by commutations under left contexts.

In the last section, we focus on congruences which are generated by commutations under left contexts. We show that, in general, they neither are represented by partial orders, nor are right cancellative. We give sufficient conditions for such congruences to ensure that they are represented by partial orders and are right cancellative. One such condition is that the width of each partial order is at most two. This is the case for instance for the "Producer / Consumer" example. This condition may seem restrictive but is the key point for a much more general one. We say that two letters are independent if they commute under some context. We show that a congruence is represented by a partial order (and is right cancellative) if whenever there exist three pairwise independent letters, these letters commute without any context. This sufficient condition covers in particular trace congruences and the "Producer / Consumer" congruence.

2 Preliminaries

In this section, we recall basic definitions of trace theory. We consider a finite alphabet A and $A^\star$ denotes the free monoid over A, which is the set of the finite words over A with the usual concatenation. For u in $A^\star$, $alph(u)$ denotes the set of letters of u, $|u|$ the length of u and $|u|_a$ the number of occurrences of the letter a in the word u. In the trace model, the letters of A are considered as actions or events in a distributed system. And, intuitively, two events a and b are said to be independent when they can be executed concurrently or in an irrelevant order. More precisely, let $I \subseteq A \times A$ be a symmetric and irreflexive relation over the alphabet A, called the independence relation. The trace monoid over (A, I), denoted by $M(A, I)$, is the quotient of the free monoid $A^\star$ by the congruence generated by $\{(ab, ba) \mid (a, b) \in I\}$. However, the trace theory is not powerful enough to describe the behaviors of all concurrency problems. The classical "Producer / Consumer" example cannot be described within this model. On the other hand, the "Producer / Consumer" paradigm can be described by the quotient of the free monoid $\{P, C\}^\star$ by the congruence generated by the pair (PPC, PCP), where P and C stand for *produce* and *consume*.

We are looking in the following to generalize these situations. The aim is still to represent a behavior of a distributed systems by its set of sequential observations. We will retain the structure of a quotient monoid of $A^\star$ through a congruence. Since the different observations of a distributed computation are all composed with the same actions (in an order which may change), we will simply require that the congruence preserves the number of the occurrences of any letter and therefore the length of any word. We will say that a congruence $\equiv$ on $A^\star$ *preserves the commutative image* if $u \equiv v$ implies $|u|_a = |v|_a$ for all $a \in A^\star$. If $\mathcal{R}$ is a relation on $A^\star \times A^\star$, we denote by $\equiv_\mathcal{R}$ (or simply $\equiv$ if it is not ambiguous) the congruence generated by $\mathcal{R}$, that is the transitive and reflexive closure of the relation $\{(w'uw'', w'vw'') \mid w', w'' \in A^\star, (u, v) \in \mathcal{R}\}$. For any word u in $A^\star$, we denote by $[u]_\mathcal{R}$ (or simply by $[u]$) the class of u in $A^\star / \equiv_\mathcal{R}$. We associate with a word u its occurrence total order $To(u) = (E_u, \leq_u)$ defined by: $E_u = \{(a, i) \mid a \in A, 1 \leq i \leq |u|_a\}$ and $(a, i) \leq_u (b, j)$ if and only if the i-th occurrence of a precedes the j-th occurrence of b in u.

From now on, we will assume that the congruence $\equiv$ preserves the commutative image. Note that if $u \equiv v$ then $E_u = E_v$. This allows us to associate with the class of

u its occurrence partial order, denoted by $Po(u)$, which is obtained as the intersection of the occurrence (total) orders of its representatives, that is to say:

$$Po(u) = \bigcap_{v \in [u]} To(v) = (E_u, \leq_{[u]}).$$

The relation $\leq_{[u]}$ is thus defined for all occurrences $x, y \in E_u$ by: $x \leq_{[u]} y \iff \forall v \in [u], x \leq_v y$.

To illustrate this definition, let us consider the "Producer / Consumer" example. Let $A = \{P, C\}$ and the congruence be generated by the relation $\mathcal{R} = \{(PPC, PCP)\}$. Let $u = PPCCP$. We have $[u] = \{PPPCC, PPCPC, PPCCP, PCPPC, PCPCP\}$ and

$$Po(u) = \left\{ \begin{tikzcd}[row sep=small, column sep=small] P \arrow[r] \arrow[dr] & P \arrow[r] \arrow[dr] & P \\ & C \arrow[r] & C \end{tikzcd} \right. .$$

Note that, in order to lighten the pictures, we always draw the transitive reductions (the Hasse diagrams) of partial orders.

Clearly we have, for all words u and v: $u \equiv v \Longrightarrow Po(u) = Po(v)$. However, the converse property: $Po(u) = Po(v) \Longrightarrow u \equiv v$ is not generally true. We can verify this with the following example: Let $A = \{a, b, c\}$ and $\equiv$ be generated by the relation $\mathcal{R} = \{(acb, cab), (bac, bca), (acb, bac), (abc, cba)\}$. Let $u = acb$. We have $[u] = \{acb, cab, bac, bca\}$ and

$$Po(u) = \left\{ \begin{array}{l} a \\ b \\ c \end{array} \right. .$$

Now let $v = abc$. We have $[v] = \{abc, cba\}$ and $Po(v) = Po(u)$ but $u \not\equiv v$.

However, this converse property holds in the trace monoid $M(A, I)$. Therefore, in $M(A, I)$ we have: for all words u and v, $u \equiv v \iff Po(u) = Po(v)$. The mapping Po from $M(A, I)$ into the set $\mathcal{PO}$ of finite partial orders is thus injective and we can identify $M(A, I)$ with its image through Po. In other words, we can identify a trace and its partial order. In order to clarify the relevance in associating the equivalence class and the partial order, let us take a look at three examples, after having given the following definition. If $P = (E_P, \leq_P)$ is a partial order, a word u is called *a linear extension of* P if $E_u = E_P$ and $\leq_P \subseteq \leq_u$, that is, $x \leq_P y \Longrightarrow x \leq_u y$ for all $x, y \in E_u$. We denote by $LE(P)$ the set of linear extensions of a partial order P.

1) The trace monoid: $A = \{a, b, c\}$ and $I = c \text{ --- } a \text{ --- } b$ is the independence relation. The congruence $\equiv$ is thus generated by: $\mathcal{R} = \{(ab, ba), (ac, ca)\}$. Let $u = acbac$. We have $[u] = \{acbac, cabac, acabc, acbca, caabc, cabca, aacbc, cbaac, cbaca, cbcaa\}$ and

$$Po(u) = \left\{ \begin{array}{l} a \longrightarrow a \\ c \longrightarrow b \longrightarrow c \end{array} \right. .$$

We can clearly see that, in this example, we have $LE(Po(u)) = [u]$.

2) Consider again the "Producer / Consumer" example presented above with $u = PPCCP$. We also have $LE(Po(u)) = [u]$.

3) Let $A = \{a, b\}$ and $\equiv$ be generated by $\mathcal{R} = \{(aab, baa)\}$. Let $u = baa$. We have $[u] = \{baa, aab\}$ and

$$Po(u) = \left\{ \begin{array}{l} a \longrightarrow a \\ b \end{array} \right. .$$

In this case, we have $LE(Po(u)) \neq [u]$. For instance, $aba \in LE(Po(u)) \setminus [u]$. The following definition will allow us to distinguish the previous situations.

Definition 2.1 *We will say that a congruence $\equiv$ is represented by a partial order when it preserves the commutative image and when for each word $u \in A^\star, [u] = LE(Po(u))$.*

When the congruence $\equiv$ preserves the commutative image, the mapping Po from $A^\star/\equiv$ into $\mathcal{PO}$ allows us to associate with each equivalence class a partial order. However in the case of a congruence represented by a partial order, we can be more precise:

Lemma 2.2 *If a congruence $\equiv$ is represented by a partial order, then the mapping Po from $A^\star/\equiv$ into $\mathcal{PO}$ is injective.*

Almost all the work which follows aims at characterizing the congruences represented by partial orders. The study of traces has already provided us with an example. We will also show that the case of the "Producer / Consumer" is a similar example.

3 Modular concatenation

We start here with the study of the behavior of the concatenation with respect to partial orders. We will prove that the congruences for which the concatenation transposes itself naturally over partial orders are trace congruences only. Within this section, $\equiv$ denotes a congruence represented by a partial order. Lemma 2.2 is of great importance. First of all, it allows us to identify an equivalence class and its partial order. However it also allows us to define a concatenation over the set of the partial orders associated with words from $A^\star$. Indeed, the mapping Po from $A^\star/\equiv$ into $\mathcal{PO}$ is injective and $A^\star/\equiv$ is a monoid, thus we can define a structure of monoid on $Po(A^\star/\equiv)$ by setting down: $Po(u) \cdot Po(v) = Po(uv)$, for all $u, v \in A^\star$.

The concatenation on partial orders which are associated with equivalence classes does not have the same properties in the case of traces as in that of the "Producer / Consumer". To illustrate this difference, let us take another look at the previous examples.

1) The trace monoid: Let $A = \{a, b, c, d\}$ and $I = c \text{ --- } a \text{ --- } b \qquad d$ be the independence relation. Let $u = acbac$ and $v = dab$. The concatenation $Po(u) \cdot Po(v)$ of their partial orders is

$$\begin{pmatrix} a & \longrightarrow & a & & \\ & & & & \\ c & \longrightarrow & b & \longrightarrow & c \end{pmatrix} \cdot \begin{pmatrix} & & a \\ & \nearrow & \\ d & & \\ & \searrow & \\ & & b \end{pmatrix} = \left\{ \begin{array}{ccccccccc} & & a & \longrightarrow & a & & & & a \\ & & & & & \searrow & & \nearrow & \\ & & & & & & d & & \\ & & & & & \nearrow & & \searrow & \\ c & \longrightarrow & b & \longrightarrow & c & & & & b \end{array} \right. .$$

In this example, the concatenation is simply obtained by taking the disjoint union of the partial orders and adding vertices from the first one to the second one.

2) The "Producer / Consumer" example: Let $u = P$ et $v = CP$. We have

$$Po(u) \cdot Po(v) = (P) \cdot (C \longrightarrow P) = \left\{ \begin{array}{ccc} P & \longrightarrow & P \\ & \searrow & \\ & & C \end{array} \right. .$$

In this case, the link $C \longrightarrow P$ of the second partial order has been destroyed by the concatenation.

The different behaviors which are observed in the previous two examples lead us to set out the following definition. To formalize this definition, we introduce the translation $t_u : A \times \mathbb{N} \longrightarrow A \times \mathbb{N}$ defined by $t_u(a,i) = (a, i + |u|_a)$. Note that, for $u, v \in A^\star$, we have $E_{uv} = E_u \cup t_u(E_v)$.

Definition 3.1 *We say that a congruence is modularly represented by a partial order if it is represented by a partial order and if the concatenation of two partial orders is obtained by taking their disjoint union and by possibly adding vertices from the first one to the second one, that is, if for all words u and v of $A^\star$, it holds:*

1. $\leq_{[uv]} \cap E_u \times E_u = \leq_{[u]}$
2. $\leq_{[uv]} \cap t_u(E_v) \times t_u(E_v) = t_u(\leq_{[v]})$
3. $\leq_{[uv]} \cap t_u(E_v) \times E_u = \emptyset$

If a congruence is modularly represented by a partial order, we will show that it can be generated by a relation in which the rules are all commutations of two letters. And thus, we prove that the congruences which are modularly represented by partial orders are exactly the trace congruences.

Theorem 3.2 *Let $\equiv_{\mathcal{R}}$ be a congruence generated by a relation $\mathcal{R}$. Let $\mathcal{C} = \{(u,v) \in \mathcal{R} \mid |u| = |v| = 2\}$ and $\equiv_{\mathcal{C}}$ the congruence generated by $\mathcal{C}$. If $\equiv_{\mathcal{R}}$ is modularly represented by a partial order, then $\equiv_{\mathcal{R}} = \equiv_{\mathcal{C}}$.*

Proof: Although we deal with two congruences $\equiv_{\mathcal{R}}$ and $\equiv_{\mathcal{C}}$, we still denote by $[u]$ and $Po(u)$ the equivalence class and the partial order for $\mathcal{R}$. If u and v are two words such that $u \equiv_{\mathcal{C}} v$, we have of course $u \equiv_{\mathcal{R}} v$. Let us now assume that there exists a pair of words (u,v) such that (*) $u \equiv_{\mathcal{R}} v$ and $u \not\equiv_{\mathcal{C}} v$. Let (u,v) be one such pair such that the length of u (and of v) is minimal.

We can distinguish the two following cases: u and v start with the same letter or not. First assume that u and v start with the same letter: $u = au'$ and $v = av'$. Clearly, $u' \not\equiv_{\mathcal{C}} v'$. Moreover we have $u' \not\equiv_{\mathcal{R}} v'$ since u and v are words of a minimal length which satisfy (*). Since the mapping Po from $A^\star/\equiv_{\mathcal{R}}$ into $\mathcal{PO}$ is injective, it follows $Po(u') \neq Po(v')$. Hence, there exist $x, y \in E_{u'} = E_{v'}$ such that, for example $x \leq_{[u']} y$ and $x \not\leq_{[v']} y$. Since $\equiv$ is modularly represented by a partial order, we deduce that $t_a(x) \leq_{[u]} t_a(y)$ and $t_a(x) \not\leq_{[v]} t_a(y)$. This contradicts the fact that $Po(u) = Po(v)$ and thus that $u \equiv_{\mathcal{R}} v$.

Assume now that u and v do not start with the same letter: $u = au_1bu_2$, $b \notin alph(u_1)$ and $v = bv_1av_2$, $a \notin alph(v_1)$. Let $w = v_1av_2$. Suppose that there exists $x \in E_{v_1}$ such that $x \leq_{[w]} (a,1)$. Since $\equiv$ is modularly represented by a partial order, we deduce that $t_b(x) \leq_{[v]} t_b((a,1)) = (a,1)$. But this contradicts the fact that $(a,1)$ is a minimal vertex of $Po(u) = Po(v)$. Now suppose that for all $x \in E_{v_1}, x \not\leq_{[w]} (a,1)$. Since $|w| < |v|$, we have $[w]_{\mathcal{R}} = [w]_{\mathcal{C}}$. Hence, $v = bv_1av_2 \equiv_{\mathcal{C}} bav_1v_2$. If $ba \not\equiv_{\mathcal{C}} ab$, $[ba] = \{ba\}$ and $(b,1) \leq_{[ba]} (a,1)$. As $\equiv$ is modularly represented by a partial order, we deduce that $(b,1) \leq_{[v]} (a,1)$. However that contradicts the fact that $(a,1)$ is a minimal vertex of $Po(u)$. Finally, if $ba \equiv_{\mathcal{C}} ab$, $v = bv_1av_2 \equiv_{\mathcal{C}} bav_1v_2 \equiv_{\mathcal{C}} abv_1v_2$. This brings us back to the first case. $\square$

Remark 3.3 We can deduce from this proposition that if $(u,v) \in \mathcal{R}$ and $|u| \geq 3$, the equivalence $u \equiv v$ is a consequence of the rules of commutation of the relation $\mathcal{C}$. As for example when $\mathcal{R} = \{(ab, ba), (aab, aba)\}$, which is a relation in which the second rule can be deduced from the first.

4 Necessary conditions

In this section, we study some necessary conditions for congruences to be represented by partial orders.

Proposition 4.1 *Let $\equiv$ be a congruence represented by a partial order. If $ua \equiv vb$ with $a \neq b$ then $ua \equiv wba \equiv wab \equiv vb$ for some $w \in A^\star$.*

Proof: As $ua \equiv vb$, the last occurrences of a and b are maximal in the partial order $Po(ua) = Po(vb)$. Let w be any linear extension of the partial order which is obtained by canceling from $Po(ua)$ the last occurrence of a and the last occurrence of b. Thus wab and wba are two linear extensions of $Po(ua)$ which proves the proposition since $\equiv$ is represented by a partial order. $\square$

This proposition can easily be generalized to:

Proposition 4.2 *Let $\equiv$ be a congruence represented by a partial order.*
If $uav \equiv u'bv$ with $a \neq b$ then $uav \equiv wbav \equiv wabv \equiv u'bv$ for some $w \in A^\star$.
If $uav \equiv ubv'$ with $a \neq b$ then $uav \equiv uabw \equiv ubaw \equiv ubv'$ for some $w \in A^\star$.

We will now study right cancellative congruences. Recall that a congruence is right cancellative if $uw \equiv vw$ implies $u \equiv v$ for all $u, v, w \in A^\star$. We will show that a congruence represented by a partial order and which is right cancellative is necessarily a congruence which can be generated by a relation in which all the rules are commutations under a left context. First, note that if there exists a rule of the form $ua \equiv va$ in the relation $\mathcal{R}$ which defines $\equiv$, we then know that $u \equiv v$. The equivalence $ua \equiv va$ is thus a consequence of the equivalence of u and v. Hence the pair (ua, va) can be removed from the relation $\mathcal{R}$ without changing the generated congruence. Therefore, without loss of generality, we may assume that the rules which define $\equiv$ are formed by words which end with distinct letters. Next, remark that Proposition 4.1 becomes now:

Proposition 4.3 *Let $\equiv$ be a right cancellative congruence represented by a partial order. If $ua \equiv vb$ with $a \neq b$ then $u \equiv wb$ and $v \equiv wa$ for some $w \in A^\star$.*

We can now obtain the following main result. Indeed, this theorem admits a symmetric version when one consider left cancellative congruences.

Theorem 4.4 *Let $\equiv$ be a congruence represented by a partial order and which is right cancellative, then $\equiv$ can be generated by a relation in which the rules are all commutations under a left context, that is to say rules of the form: $uab \equiv uba$.*

Proof: As mentioned above, we may assume that $(ua, vb) \in \mathcal{R}$ implies $a \neq b$. Using Proposition 4.3, we associate with each rule $(ua, vb) \in \mathcal{R}$ a commutation under left context $\Phi(ua, vb) = (wab, wba)$ where $u \equiv wb$ and $v \equiv wa$. Let $\mathcal{S} = \Phi(\mathcal{R})$. We claim that $\equiv_{\mathcal{R}} = \equiv_{\mathcal{S}}$.

First let $(ua, vb) \in \mathcal{R}$ and let $\mathcal{R}_1 = \mathcal{R} \setminus \{(ua, vb)\}$. Let $(wab, wba) = \Phi(ua, vb)$ and let $\mathcal{R}' = \mathcal{R}_1 \cup \{(wab, wba)\}$. Since $|wa| = |wb| < |ua| = |vb|$, we have $u \equiv_{\mathcal{R}_1} wb$ and $v \equiv_{\mathcal{R}_1} wa$. Hence, we deduce $ua \equiv_{\mathcal{R}'} vb$ and $wab \equiv_{\mathcal{R}} wba$. Therefore, $\mathcal{R}$ and $\mathcal{R}'$ do in fact generate the same congruence. If $\mathcal{R}$ is finite, it follows by induction on the number of rules in $\mathcal{R}$ that $\equiv_{\mathcal{R}} = \equiv_{\mathcal{S}}$.

Let $n \in \mathbb{N}$. We set $\mathcal{R}_n = \{(u,v) \in \mathcal{R} \mid |u| = |v| \leq n\}$ and $\mathcal{S}_n = \{(u,v) \in \mathcal{S} \mid |u| = |v| \leq n\}$. Clearly, $\mathcal{R}_n$ is finite and $\mathcal{S}_n = \Phi(\mathcal{R}_n)$. Hence, $\equiv_{\mathcal{R}_n} = \equiv_{\mathcal{S}_n}$. Since our congruences preserve the length of words, it follows for all $x, y \in A^\star$

$$x \equiv_{\mathcal{R}} y \Longleftrightarrow x \equiv_{\mathcal{R}_{|x|}} y \Longleftrightarrow x \equiv_{\mathcal{S}_{|x|}} y \Longleftrightarrow x \equiv_{\mathcal{S}} y.$$

$\square$

Remark 4.5 Let us now return to the case where $\equiv$ is represented by a partial order without being right cancellative. And let us assume that there exists a rule which defines $\equiv$ of the form $uav \equiv u'bv$. We know from Proposition 4.2 that there exists a word $w \in A^\star$ such that $uav \equiv wbav \equiv wabv \equiv u'bv$. But here it is not in general possible to replace the rule $uav \equiv u'bv$ by the new rule $wabv \equiv wbav$ in the relation which defines $\equiv$ without changing the generated congruence.

For instance, let $\mathcal{R} = \{(cac, cca), (cab, cba), (bccab, cacbb)\}$. The relation obtained as above by transforming the third rule is $\mathcal{R}' = \{(cac, cca), (cab, cba), (ccabb, ccbab)\}$. And we can verify that $bccab \not\equiv_{\mathcal{R}'} cacbb$.

We have just established that a congruence represented by a partial order and which is right cancellative can be generated by a relation in which the rules are all commutations under left contexts (Theorem 4.4). There are of course two questions which must be addressed. Let $\equiv$ be a congruence which is defined by commutations under left contexts. Is $\equiv$ right cancellative? Can $\equiv$ be represented by a partial order? The answer to these questions is no, not necessarily. To realize this, let us consider the congruence defined by the relation $\mathcal{R} = \{(cab, cba), (bac, bca), (abc, acb)\}$. $\equiv$ is not right cancellative: it is easy to show that $abcbca \equiv acbcba$ and $abcbc \not\equiv acbcb$. Hence, $\equiv$ is not represented by a partial order. Let $u = abca$. We have $[u] = \{abca, acba, acab, abac\}$ and

$$Po(u) = \left\{ \begin{array}{ccc} & & a \\ & \nearrow & \\ a & \rightarrow & b \\ & \searrow & \\ & & c \end{array} \right. .$$

There are six linear extensions of $Po(u)$ and only four elements in $[u]$.

In fact, this example is very characteristic. The letters which commute (under a certain context) are a and b, a and c, and b and c. And if we define the independence relation I by $I = \{(a,b) \in A \times A \mid a \neq b \text{ and } wab \equiv wba \text{ for some } w \in A^\star\}$, then the graph of I is a triangle. We will come back to this situation latter.

5 Commutations under left contexts

The purpose of this section is to determine a family of congruences defined by commutations under left contexts and which can in addition be represented by a partial order. The following results depend on the "width" of the partial orders associated with the words. If $P = (E_P, \leq_P)$ is a partial order, recall that *an antichain* of P is a set of vertices pairwise incomparable for $\leq_P$ and *the width of* P is the maximal length of antichains of P. The first essential result which we will prove is the following:

Theorem 5.1 *Let $\equiv$ be a congruence generated by (a relation in which the rules are) commutations under left contexts. If for each word $u \in A^\star$, the partial order $Po(u)$ is of width at most two, then the congruence $\equiv$ is represented by a partial order and it is right cancellative.*

This theorem is a direct consequence of the following three lemmas.

Lemma 5.2 *Let $\equiv$ be a congruence which preserves the commutative image. Let u be a word of $A^\star$ such that the partial order $Po(u)$ has at most two maximal elements. If an occurrence of the letter a is maximal in $Po(u)$, then there exists a word v such that $u \equiv va$.*

Proof: Assume that $u = u'b$ with $b \neq a$. Let x be the last occurrence of a and y be the last occurrence of b. Since x is maximal in $Po(u)$, we have $x \not\leq_{[u]} y$. Hence $u \equiv w$ for some $w = w_1 b w_2 a w_3 \in A^\star$ with $a \notin alph(w_3)$ and $b \notin alph(w_2 w_3)$. Since $Po(u)$ has at most two maximal vertices, it follows $w_3 = \varepsilon$, which proves the lemma. $\square$

For instance, let $\equiv$ be the congruence generated by $\mathcal{R} = \{(cca, cac), (cab, cba)\}$. Let $u = cacab$, we have $[u] = \{cacab, ccaab, cacba, ccaba, ccbaa\}$. The letter a has a maximal occurrence in $Po(u)$ and we do have $u \equiv va$ if we take for example $v = ccab$. Let us now compare $Po(u)$ and $Po(v)$. We have $[v] = \{ccab, cacb, ccba\}$ and we can see that $Po(v)$ is obtained by canceling in $Po(u)$ the last occurrence of a. The next lemma state that this is a general fact.

$$Po(u) = \left\{ \begin{tikzcd} & a \arrow[r] & a \\ c \arrow[ur] \arrow[dr] & & \\ & c \arrow[r] \arrow[uur] & b \end{tikzcd} \right. \qquad Po(v) = \left\{ \begin{tikzcd} & a & \\ c \arrow[ur] \arrow[dr] & & \\ & c \arrow[r] & b \end{tikzcd} \right.$$

Let $P = (E_P, \leq_P)$ be a partial order and let $x \in E_P$ be a maximal vertex of P, we denote by $P \setminus \{x\}$ the partial order $(E \setminus \{x\}, \leq_P \bigcap (E \setminus \{x\} \times E \setminus \{x\})$.

Lemma 5.3 *Let $\equiv$ be a congruence generated by commutations under left contexts. Let u be a word of $A^\star$ such that the partial order $Po(u)$ is of width at most two. If α is an occurrence of the letter a maximal in $Po(u)$ and v is a word such that $u \equiv va$, then $Po(u) \setminus \{\alpha\} = Po(v)$.*

Proof: Given the definition of $Po(u) \setminus \{\alpha\}$, we need to prove that $E_v = E_u \setminus \{\alpha\}$ and that for all $x, y \in E_v$, $x \leq_{[u]} y \iff x \leq_{[v]} y$. The first point is obvious. Let $x, y \in E_v$ and assume that $x \leq_{[u]} y$. Let w be any equivalent of v, we have $x \leq_{wa} y$. As $x, y \in E_w$, we even have $x \leq_w y$. Thus $x \leq_{[v]} y$.

Conversely, assume that $x \leq_{[v]} y$, and let us suppose that $x \not\leq_{[u]} y$. Then there exists an equivalent w of u, such that $y <_w x$. We are now interested in the successive applications of rules which allow us to transform va into w:

$$va = v'_0 a v''_0 \equiv v'_1 a v''_1 \equiv \cdots \equiv v'_n a v''_n = w$$

where $v''_0 = \varepsilon$ and $a \notin alph(v''_i)$ for $0 \leq i \leq n$. And each transformation corresponds exactly to a commutation under left context. Clearly, $v = v'_0 v''_0$, but from $x \leq_{[v]} y$ and $y <_w x$ w deduce that $v \not\equiv v'_n v''_n$. Let i be then the smallest index such that $v \not\equiv v'_{i+1} v''_{i+1}$. We consider the commutation under a left context which has been carried out in order to transform $v'_i a v''_i$ into $v'_{i+1} a v''_{i+1}$. We claim that the last occurrence of a is necessarily in the context of this commutation. Indeed, in the two other cases (either this occurrence of a is one of the two letters which commute, or this occurrence of a does not commute and is not part of the context), we deduce that $v'_i v''_i \equiv v'_{i+1} v''_{i+1}$ which contradicts the definition of i. Therefore, the two occurrences of the letters which commute in v''_i and this occurrence of a form an antichain of length three of $Po(u)$, which is a contradiction. □

Lemma 5.4 *Let $\equiv$ be a congruence generated by commutations under left contexts. $\equiv$ is represented by a partial order and is right cancellative if and only if*

(i) for any word u of A^*, if the letter a has a maximal occurrence in $Po(u)$, then there exits a word v such that $u \equiv va$

(ii) for all words u and v such that $u \equiv va$, if we denote by α the last occurrence of a in u, then $Po(u) \setminus \{\alpha\} = Po(v)$.

Proof: Assume that $\equiv$ is represented by a partial order and is right cancellative. It is easy to prove the first point (i). To prove (ii), we go back to the method proposed in the proof of Lemma 5.3. The first implication may be proved in the same way. And we now have to prove that $\forall x, y \in E_v$, $x \leq_{[v]} y \Longrightarrow x \leq_{[u]} y$. Assume that $x \leq_{[v]} y$. If $x \not\leq_{[u]} y$, we can find a linear extension of $Po(u)$ of the form $w = w_1 y w_2 x w_3 a$ which is also an equivalent of u. But using the right cancellability we get $v \equiv w_1 y w_2 x w_3$. This is impossible because $x \leq_{[v]} y$.

Conversely, we first prove by induction on the length of u that each linear extension of $Po(u)$ is a word equivalent to u. This is clear if $|u| = 1$. We suppose that the property is true for every word of length at most n. Let u be a word of length $n+1$ and let $w = u'a$ be any linear extension of $Po(u)$. As a has an occurrence α maximal in $Po(u)$ there exists a word v such that $u \equiv va$ and $Po(u) \setminus \{\alpha\} = Po(v)$. But u' is a linear extension of $Po(u) \setminus \{\alpha\} = Po(v)$. Therefore $u' \equiv v$ (by induction hypothesis). And consequently $u'a = w \equiv va \equiv u$. It remains to prove that $\equiv$ is right cancellative. Assume that $ua \equiv va$. By hypothesis, we have $Po(ua) \setminus \{\alpha\} = Po(v)$. But u is a linear extension of $Po(ua) \setminus \{\alpha\}$ and thus $u \equiv v$, because $\equiv$ is represented by a partial order. □

Corollary 5.5 *When A is an alphabet of two letters, every congruence generated by commutations under left contexts is represented by a partial order and is right cancellative.*

Remark 5.6 It is of course the case of the "Producer / Consumer" paradigm.

The following results provide other families of congruences defined by commutations under left contexts which are represented by partial orders and which are right cancellative. If $\equiv$ is generated by a relation $\mathcal{R}$ in which the rules are commutations under left contexts, the independence relation associated with $\equiv$ is $I = \{(a,b) \in A \times A \mid a \neq b \text{ and } (uab, uba) \in \mathcal{R} \text{ for some } u \in A^\star\}$. We say that I contains a *triangle* if it contains a clique of three elements, that is to say when $I \supseteq \{(a,b),(b,c),(c,a),(a,c),(c,b),(b,a)\}$ for some $a,b,c \in A$.

Proposition 5.7 *Let $\equiv$ be a congruence generated by commutations under left contexts. If the independence relation associated with $\equiv$ does not contain a triangle, then, for every word $u \in A^\star$, the partial order $Po(u)$ has a width at most two (and thus $\equiv$ is represented by a partial order and is right cancellative).*

Proof: If the partial order $Po(u)$ contains an antichain of length three then these three occurrences must pairwise commute and the independence relation contains a triangle. □

Theorem 5.8 *Let $\equiv$ be a congruence generated by commutations under left contexts. If, for each triangle $\{(a,b),(b,c),(c,a),(a,c),(c,b),(b,a)\}$ contained in the independence relation I associated with $\equiv$, the rules which commute a and b, b and c and a and c are simply commutations without context, then $\equiv$ is represented by a partial order and is right cancellative.*

Proof: To prove this result we are going to use Lemma 5.4. First, let u be a word and assume that the letter a has a maximal occurrence in $Po(u)$. We claim that there exists a word v such that $u \equiv va$. Assume that $u = u'b$ with $b \neq a$. There exist thus two equivalents of u of the form $u \equiv w'abw'' \equiv w'baw''$, with $a, b \notin alph(w'')$. If $w'' \neq \varepsilon$, each letter of its alphabet forms a triangle with a and b in I. Thus, we have $u \equiv w'abw'' \equiv w'w''ab \equiv w'w''ba$. Hence, u has an equivalent which ends with a. Next, we prove the second point (ii) of Lemma 5.4. Using the technique of the proof of Lemma 5.3, we arrive at the same conclusion: the last occurrence of a must be in the context of the commutation: $v'_i a v''_i \equiv v'_{i+1} a v''_{i+1}$. Hence, the two occurrences of the letters which commute here and the last occurrence of a form a triangle of I in which two letters do not commute under an empty context. This contradicts the hypothesis concerning I. □

References

[AR88] I.J. Aalbersberg and G. Rozenberg. Theory of traces. *Theoretical Computer Science*, 60:1–82, 1988.

[BR94] I. Biermann and B. Rozoy. Context traces and transition systems. In S. Kuru, M.U. Caglayan, E. Gelembe, H.L. Akin, and C. Ersoy, editors, *Proceedings of the 9th International Symposium on Computer and Information Science ISCIS IX*, pages 301–309. Bogazici University Printhouse, Turkey, 1994.

[CL87] M. Clerbout and M. Latteux. Semi-Commutations. *Information and Computation*, 73:59–74, 1987.

[CLR95] M. Clerbout, M. Latteux, and Y. Roos. Semi-commutations. In G. Rozenberg and V. Diekert, editors, *The Book of Traces*, pages 487–552. World Scientific, Singapore, 1995.

[Die90] V. Diekert. *Combinatorics on Traces.* Number 454 in Lecture Notes in Computer Science. Springer Verlag, 1990.

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

[HKT92] P.W. Hoogers, H.C.M. Kleijn, and P.S. Thiagarajan. A trace semantics for petri nets. In W. Kuich, editor, *Proceedings of the 19th International Colloquium on Automata Languages and Programming (ICALP'92)*, number 623 in Lecture Notes in Computer Science, pages 595–604. Springer Verlag, 1992.

[Lac92] J. Lacaze. Parties reconnaissables de monoïdes définis par générateurs et relations. *R.A.I.R.O. — Informatique Théorique et Applications*, 26:541–552, 1992.

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

[Maz87] A. Mazurkiewicz. Trace theory. In W. Brauer et al., editors, *Advances in Petri Nets'86*, number 255 in Lecture Notes in Computer Science, pages 279–324. Springer Verlag, 1987.

[NW95] M. Nielsen and G. Winskel. Trace structures and other models for concurrency. In G. Rozenberg and V. Diekert, editors, *The Book of Traces*, pages 271–306. World Scientific, Singapore, 1995.

[Per89] D. Perrin. Partial commutations. In G. Ausiello et al., editors, *Proceedings of the 16th International Colloquium on Automata, Languages and Programming (ICALP'89)*, number 372 in Lecture Notes in Computer Science, pages 637–651. Springer Verlag, 1989.

[Win88] G. Winskel. An introduction to event structures. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, number 354 in Lecture Notes in Computer Science, pages 123–172. Springer Verlag, 1988.