INFORMATIQUE THÉORIQUE ET APPLICATIONS

A. ARNOLD

An extension of the notions of traces and of asynchronous automata

Informatique théorique et applications, tome 25, nº 4 (1991), p. 355-393.

<http://www.numdam.org/item?id=ITA_1991__25_4_355_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 25, n° 4, 1991, p. 355 à 393)

AN EXTENSION OF THE NOTIONS OF TRACES AND OF ASYNCHRONOUS AUTOMATA (*)

by A. ARNOLD [1]

Communicated by R. CORI

Abstract. – *We extend the notion of a trace (or element of a partially commutative monoid) in considering a commutation relation over letters which depends on the context where it is applied. Accordingly, we extend the notion of an asynchronous automaton and we prove a generalizaton of Zielonka's theorem for these notions.*

Résumé. – *Nous généralisons la notion de trace, c'est-à-dire d'un élément d'un monoïde partiellement commutatif, en faisant dépendre la relation de commutation entre lettres du contexte dans lequel elle est appliquée. Nous généralisons de façon correspondante la notion d'automate asynchrone et nous généralisons le théorème de Zielonka à ces notions.*

1. INTRODUCTION

Words, interpreted as sequences of actions, are very natural formalizations of sequential computations, and the free monoid is the best mathematical structure in which one can speak of words. Indeed, monoid theory is one of the corner stones of Theoretical Computer Science.

But a word cannot formalize a concurrent computation. One needs more complex objects, which are all more or less related to partially ordered sets. Many works are devoted to the definition of such objects and to the construction of mathematical structures which are to concurrent computations what the monoid is to sequential ones. Let us cite just a few of them

- partial words and partial languages [6];

(*) Received November 1989.

(1) Laboratoire d'Informatique, Unité de Recherche associée au Centre National de la Recherche Scientifique n° 1304, Université de Bordeaux-I, 351, cours de la Libération, 33405 Talence, France.

- histories and processes [16, 17];
- event structures [11];
- occurence nets and processes [5];
- process monoids [7];
- concurrent histories [4];
- distributed monoids [14].

Among these attempts, the partially commutative monoid, or trace monoid, [7, 3, 1] is especially fruitful, at least at the mathematical level, in the sense that this theory leads to nice and deep theorems [10, 12]. One of the most interesting results in this theory is Zielonka's theorem [18] which asserts the equivalence between two different notions of recognizability, one defined by algebraic criteria, the other one by specific devices: the so-called asynchronous automata. In this theory, traces are equivalence classes of words under the equivalence relation $\sim_\theta$ which is the least monoid congruence generated by the set $\{ab \sim_\theta ba\}$ for all pairs $\langle a, b \rangle$ belonging to some symmetric irreflexive relation θ. A trace can be represented by a partially ordered set where two occurences of letters are not ordered if and only if these occurrences can commute in some word of the equivalence class. An asynchronous automaton is a finite state automaton which is able to have traces as inputs in the sense that two independent occurrences of letters have independent effects on the states of the automaton.

Our aim was to generalize this theory in the following manner:

- to characterize equivalence relations on the free monoid such that equivalence classes can be represented by partially ordered sets;
- to define a kind of automaton which can be considered as reading such partially ordered sets.

It was achieved in the following way:

- we define an extension of the notion of a trace, called P-trace; it is a labelled partially ordered finite set, such that two independent elements must have different labels;
- we define particular sets of P-traces, called CCI sets, such that elements of these sets can be identified with equivalence classes of words;
- we define a generalization of the notion of an asynchronous automaton, called P-asynchronous automaton, which takes as inputs the elements of a CCI set.

Indeed we define various kinds of CCI sets.

Projective: It is defined as the "projection" of a subset of a partially commutative monoid.

r-projective: It is defined as the "projection" of a *recognizable* subset of a partially commutative monoid.

Regular: A CCI set is regular if its syntactic right semi-congruence is of finite index.

a-regular: A CCI set is a-regular if the equivalence relation associated with it is defined by a P-asynchronous automaton, in the same way as an asynchronous automaton defines a commutation relation and, therefore, a partially commutative monoid.

The main result we prove in this paper is a generalization of Zielonka's theorem: every recognizable subset of an a-regular CCI set P is recognized by a P-asynchronous automaton such that the equivalence relation associated with it is the equivalence relation which characterizes P. Indeed it is a consequence of Zielonka's theorem!

We also try to characterize the equivalence relations which define *a-regular* CCI sets. Unfortunately, we were only able to show that a CCI set is regular if and only if it is r-projective, and we propose as a conjecture the fact that a CCI set is a-regular if and only if it is regular and the equivalence relation associated with it satisfies some additional property. The proofs in this part of the paper are rather similar to the proofs in Zielonka's paper; the difficulty lies in the fact that the notions introduced by Zielonka have to be adapted to the case we deal with. We expect the extensions of these notions could have some interest in their own right.

This paper is organized as follows. In section 2, we define the notion of a P-trace and the notion of a CCI set of P-traces. In section 3, we characterize the equivalence relations such that the set of equivalence classes is a CCI set. In section 4, we define the projective and r-projective CCI sets. In section 5, we define the notion of a P-asynchronous automaton and we prove the extension of Zielonka's theorem. In section 6, we define the syntactic semi-congruence and the syntactic congruence of a CCI set; we extend some notions introduced by Zielonka, and we prove the equivalence between the notion of a regular CCI set and of an r-projective CCI set. In section 7, we propose our conjecture about the characterization of a-regular CCI sets.

2. P-TRACES

DEFINITION 2.1: A *P-trace t* over a finite alphabet A is a triple $\langle E_t, \leqq_t, \lambda_t \rangle$ where $\langle E_t, \leqq_t \rangle$ is a partially ordered set and λ_t is a mapping from E_t into A which satisfies the following property:

$$\forall x, y \in E_t, \quad \lambda_t(x) = \lambda_t(y) \Rightarrow x \leqq_t y \text{ or } y \leqq_t x. \quad \square \tag{1}$$

The partially ordered set $\langle E_t, \leqq_t \rangle$ expresses a kind of "causality" relation between the elements of E_t and the label $\lambda_t(x)$ of an element x names the "event" associated with this element. Therefore a P-trace is nothing but a *pomset* in some terminologies [13]. The condition (1) means that two occurrences of the same event have to be causally related and this property is sometimes termed as "non autoconcurrency" [13] or "self-dependency" [8].

Indeed, properly speaking, P-traces are equivalence classes under isomorphism of such triples, but, due to the property (1), there is a canonical representative of each class, having the following form: replace each element x of E_t by the pair $\langle \lambda_t(x), n_x \rangle$ where n_x is the number of elements $y \in E_t$ such that $y \leqq_t x$ and $\lambda_t(x) = \lambda_t(y)$. From now on a P-trace will be identified with this canonical representative.

DEFINITION 2.2: A P-trace $t = \langle E_t, \leqq_t, \lambda_t \rangle$ is a *prefix* of a P-trace $t' = \langle E_{t'}, \leqq_{t'}, \lambda_{t'} \rangle$, denoted by $t \sqsubseteq t'$, if

- $E_t \subseteq E_{t'}$;
- $\forall x \in E_t, \lambda_t(x) = \lambda_{t'}(x)$;
- $\leqq_t = \leqq_{t'} \cap (E_t \times E_t)$ (*i.e.*, $\leqq_t$ is the restriction of $\leqq_{t'}$ to E_t);
- if $x \in E_t$, $y \in E_{t'}$, and $y \leqq_{t'} x$ then $y \in E_t$. $\square$

DEFINITION 2.3: For a word u of A^*, we denote by alph(u) the set of letters occurring in u. $\square$

DEFINITION 2.4: If t_1 and t_2 are two prefixes of a P-trace t, then one can define the *intersection* $t_1 \sqcap t_2$ of t_1 and t_2 as the "largest" prefix common to t_1 and t_2 (its domain is $E_{t_1} \cap E_{t_2}$) and their *union* $t_1 \sqcup t_2$ as the least prefix of t having t_1 and t_2 as prefixes (its domain is $E_{t_1} \cup E_{t_2}$). $\square$

DEFINITION 2.5: If a P-trace is a *totally* ordered set, it is a *word* over A, *i.e.*, an element of the free monoid A^*. A *linear extension* of a P-trace t is a word u, considered as a P-trace, whose order $\leqq_u$ is compatible with $\leqq_t$. More formally $u = \langle E_u, \leqq_u, \lambda_u \rangle$ is a linear extension of $t = \langle E_t, \leqq_t, \lambda_t \rangle$

if $\leqq_u$ is a total order, $E_u = E_t$, $\lambda_u = \lambda_t$ and $\leqq_t \subseteqq \leqq_u$ (*i.e.*, $\forall x, y \in E_u = E_t$, $x \leqq_t y \Rightarrow x \leqq_u y$).

We shall denote by $LE(t)$ the set of linear extensions of a P-trace t, which is a subset of A^*. □

The following proposition is more or less folklore, although the result is sometimes attributed to Szpilrajn [15].

PROPOSITION 2.1: *If t is a P-trace, then $\leqq_t$ is the intersection of all $\leqq_u$ for $u \in LE(t)$.*

DEFINITION 2.6: Two words u and v in A^* are said to be *Parikh-equivalent* if, for every letter $a \in A$, the numbers of a's in u and in v are equal. □

Obviously, all words in $LE(t)$ are Parikh-equivalent. Conversely, with every set L of Parikh-equivalent words, one can associate the P-trace t_L defined by $\leqq_{t_L} = \bigcap_{u \in L} \leqq_u$. It is obvious that $L \subseteqq LE(t_L)$.

DEFINITION 2.7: Let P be a set of P-traces over a given alphabet A. We say that P is *consistent* and *complete*, CC for short, if

- $\bigcup_{t \in P} LE(t) = A^*$ (completeness);
- $\forall t, t' \in P$, $t \neq t' \Rightarrow LE(t) \cap LE(t') = \varnothing$ (consistency). □

DEFINITION 2.8: If a set P of P-traces is CC, it follows, from the very definition, that every word u in A^* is the linear extension of one and only one P-trace t in P. Let us denote by $\varphi : A^* \rightarrow P$ the mapping which associates with each word the unique P-trace which contains it in its linear extension. This mapping allows us to define an equivalence relation denoted by $\sim_P$, or more simply $\sim$ when P can be understood from the context, by $u \sim_P v$ iff $\varphi(u) = \varphi(v)$.

This mapping is said to be *monotonic* if $u \sqsubseteq v \Rightarrow \varphi(u) \sqsubseteq \varphi(v)$. □

DEFINITION 2.9: We say that a set P of P-traces is *ideal* if it is closed under prefix, *i.e.*, if $t \in P$ and $t' \sqsubseteq t$ then $t' \in P$. □

For CC sets, we have the following equivalent definition of "ideality".

PROPOSITION 2.2: *A* CC *set P of P-traces is ideal if and only if the mapping φ is monotonic.*

Proof 1: Let t be in P and let t' be a prefix of t. Let u' be any linear extension of t'. Then t has a linear extension u in the form $u'u''$ and $u' \sqsubseteq u$. Hence, $t'=\varphi(u') \sqsubseteq \varphi(u)=t$ and $t'=\varphi(u')$ is in P.

2. Let u and v be two words in A^* such that u is a prefix of v. Let t be $\varphi(v)$ and let us consider the prefix t_u of t whose domain is E_u. To be sure that this prefix does exist, it is sufficient to prove that $x \in E_u$ and $y \leqq_t x$ implies $y \in E_u$. Indeed, $y \leqq_t x \Rightarrow y \leqq_v x$, since v is a linear extension of t, and since $u \sqsubseteq v$, we get $y \in E_u$. Moreover, u is a linear extension of t_u (if $x, y \in E_u = E_{t_u}$ and $x \leqq_{t_u} y$, then $x \leqq_t y$, which implies $x \leqq_v y$, which implies $x \leqq_u y$, since x and y are both in E_u), thus $\varphi(u)=t_u \sqsubseteq t=\varphi(v)$. □

In all the rest of this paper we shall consider only ideal CC sets of P-traces, CCI sets for short. Indeed if processes are modelled by P-traces, and set of processes associated with some machinery (device or program) by sets of P-traces, as far as the "beginning" of a process is still a process, the set of P-traces has to be closed under prefix. The condition of completeness can always be satisfied by suitably extending the set of P-traces under consideration. The main restriction in limiting ourself to CCI sets is due to the property of consistency; it excludes the case if *mixed causality* illustrated by the following example, due to Mazurkiewicz [9]. On the other hand it is a natural generalization of the notion of partially commutative monoid [3], as shown by the next example.

Example 2.1: Let us consider the Petri net pictured in figure 1. If we fire sequentially a then c then b, the firing of c is made possible by the firing of a, and the firing of b can be considered as independent, thus the "causality ordering" between these events is reduced to $a \leqq c$. On the other hand, if we fire sequentially b then c then a, the firing of c is made possible by the firing of b, and the firing of a can be considered as independent, and the "causality ordering" is reduced to $b \leqq c$. The total order $a \leqq b \leqq c$ is a linear extension of the two previous partial orders; thus, in this example, the requirement of "consistency" is not satisfied. □

Example 2.2: Let us consider an alphabet A and let θ be a symmetric irreflexive relation in $A \times A$. We define, over A^*, the least congruence relation $\sim_\theta$ generated by (ab, ba) for every pair $\langle a, b \rangle$ in θ. Since congruent words are obviously Parikh-equivalent, one can define $\varphi(u)$ as the P-trace which is the intersection of all $\leqq_v$ for all v congruent to u. It is easy to show that

$LE(\varphi(u))$ is exactly the congruence class of u and that $u \sqsubseteq v \Rightarrow \varphi(u) \sqsubseteq \varphi(v)$; hence, $\varphi(A^*)$ is a CCI set of P-traces. □

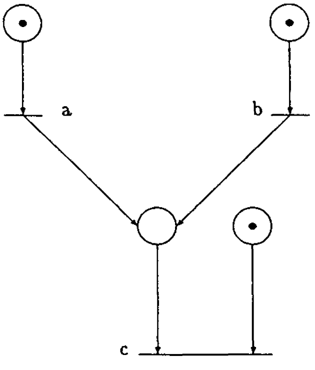

Figure 1

3. AN ALGEBRAIC CHARACTERIZATION OF CCI SETS OF P-TRACES

Given a CCI set P of P-traces, one can associate with it an equivalence relation $\sim$ over A^*, as explained in the previous section. This equivalence is included in the Parikh-equivalence and has other properties; some of them will be listed below. Conversely, if an equivalence relation over A^* is included in the Parikh-equivalence, it allows us to associate with each word u the P-trace t_u which is the intersection of all $\leqq_v$ for all v equivalent to u. The set of all these t_u is not necessarily a CCI set. We shall give sufficient conditions on this equivalence to get this property, so that CCI sets could be identified with equivalence relations satisfying these conditions.

PROPOSITION 3.1: *The equivalence relation $\sim$ over A^* associated with a* CCI *set P of P-traces has the following properties:*

P1: *It is included in the Parikh equivalence.*

P2: *For any words u, v and for any letter a, $u \sim v$ if and only if $ua \sim va$.*

P3: *For any words u, $v \in A^*$ and for any letters a, b with $a \neq b$, if $ua \sim vb$, then there exists a word w such that $u \sim wb$ and $v \sim wa$.*

P4: *For any words u, v, w and any letters a, b such that* (i) *$a \neq b$;* (ii) *w does not contain b;* (iii) *vw does not contain a, if $uavbw \sim uvwba$ then $uvwab \sim uvwba$.*

Proof: P1 is a consequence of the definition of $\sim$.

Proof of P2: Since $u \sqsubseteq ua$ and $v \sqsubseteq va$, by proposition 2.2, $\varphi(u) \sqsubseteq \varphi(ua)$ and $\varphi(v) \sqsubseteq \varphi(va)$.

If $u \sim v$, then v is a linear extension of $\varphi(u)$, and it follows that va is a linear extension of $\varphi(ua)$; hence, $ua \sim va$.

If $ua \sim va$, we have $t = \varphi(ua) = \varphi(va)$ and $\varphi(u)$ and $\varphi(v)$ are both prefixes of t. Since their domains are equal, they are equal.

Proof of P3: Let us assume that $\varphi(ua) = \varphi(vb) = t$ with $a \neq b$. Both $\varphi(u)$ and $\varphi(v)$ are prefixes of t. Let t' be their intersection and w be some linear extension of t'. It remains to prove that $\varphi(wb) = \varphi(u)$, *i.e.*, that wb is a linear extension of $\varphi(u)$; by the same reasoning we will get $\varphi(wa) = \varphi(v)$.

By definition of t' we have $E_{t'} = E_w = E_u \cap E_v$. Since ua and vb are Parikh-equivalent, u is equal to $u_1 b u_2$ and v to $v_1 a v_2$ with (i) b does not occur in u_2, (ii) a does not occur in v_2 and (iii) $u_1 u_2$ and $v_1 v_2$ are Parikh-equivalent; hence, $E_u = E_w \cup \{\langle b, n \rangle\}$, where n is the number of b's in u. Let us prove that for any $x, y \in E_u$ we have $x \leqq_{\varphi(u)} y \Rightarrow x \leqq_{wb} y$:

If $x, y \in E_w$, $x \leqq_{\varphi(u)} y \Rightarrow x \leqq_{t'} y \Rightarrow x \leqq_w y \Rightarrow x \leqq_{wb} y$;

if $x \in E_w$, $y = \langle b, n \rangle$, obviously $x \leqq_{wb} y$;

if $x = \langle b, n \rangle$, $y \in E_w$, it is impossible that $x \leqq_{\varphi(u)} y$, since, because $\varphi(u) \sqsubseteq t$, this implies $x \leqq_y y$, and, since vb is a linear extension of t, $x \leqq_{vb} y$, which is not possible.

Proof of P4: Let us assume that $t = \varphi(uavbw) = \varphi(uvwba)$ and let us prove that $uvwab$ is a linear extension of t. Let us set $f = uavbw$, $g = uvwba$, and $h = uvwab$. Since $\leqq_t \subseteq \leqq_f \cap \leqq_g$, it suffices to show that $\forall x, y \in E_t$, $x \leqq_f y$ and $x \leqq_g y \Rightarrow x \leqq_h y$ which is easily shown, considering the 25 cases where x and y belong to E_u, $E_{uv} - E_u$, $E_{uvw} - E_{uv}$, $E_{uvwb} - E_{uvw}$, $E_g - E_{uvwb}$. □

Conversely, if an equivalence relation $\sim$ over A^* is included in the Parikh-equivalence, with every word u in A^* we associate the P-trace $\varphi(u)$ defined by $\leqq_{\varphi(u)} = \bigcap_{v \sim u} \leqq_v$. If, moreover, this equivalence relation satisfies P2, P3, and P4 then $\varphi(A^*)$ is a CCI set of P-traces. To prove this result, we need the following lemma.

LEMMA 3.2: *Let us assume that the equivalence relation* $\sim$ *satisfies* P2 *and* P3. *If* $uav \sim wa$ *and if the word* v *does not contain the letter* a *then* $uv \sim w$.

Proof: The proof is by induction on the length of v. If v is the empty word, we have $ua \sim wa$ and we get $u \sim w$ by P2. If $v = v'b$, since a does not occur in v, $b \neq a$, and, by P3, $uav'b \sim wa$ implies that there exists a word w' such that $uav' \sim w'a$ and $w \sim w'b$. By induction hypothesis $uv' \sim w'$, and by P2, $uv'b \sim w'b$; hence, $uv \sim w$. □

Now we can prove the previously announced result.

PROPOSITION 3.3: *If an equivalence relation* $\sim$ *satisfies* P1, P2, P3, *and* P4, *it defines a* CCI *set* P *of* P*-traces such that the* $\sim$*-equivalence classes are the linear extensions of elements of* P.

Proof: Since u is by definition a linear extension of $\varphi(u)$, the completeness property is satisfied.

Let us prove, by induction on the length of v, that $\varphi(u) \sqsubseteq \varphi(uv)$. If v is the empty word, this is trivially true. Let v be equal to aw and let us assume that $\varphi(ua) \sqsubseteq \varphi(uaw)$; it is sufficient to show that $\varphi(u) \sqsubseteq \varphi(ua)$, *i.e.*

1. $\forall x, y \in E_u$, $x \leqq_{\varphi(u)} y$ iff $x \leqq_{\varphi(ua)} y$;
2. $\forall x, y \in E_{ua}$, if $y \in E_u$ and $x \leqq_{\varphi(ua)} y$ then $x \in E_u$.

Proof of 1: Let x and y be in E_u. If $x \leqq_{\varphi(ua)} y$, for each v such that $u \sim v$, we have $ua \sim va$; hence, va is a linear extension of $\varphi(ua)$, and thus $x \leqq_{va} y$; since x and y are in $E_u = E_v$, we get $x \leqq_v y$ and by definition of φ, $x \leqq_{\varphi(u)} y$. If $x \leqq_{\varphi(u)} y$ and if $x \nleqq_{\varphi(ua)} y$, there exists, by definition of φ, a word w such that $w \sim ua$ and $y \leqq_w x$; the word w can be written as $w_1 a w_2$ where a does not occur in w_2, and, by Lemma 3.2, $u \sim w_1 w_2$ with $E_{w_1 w_2} = E_u$; it follows that $w_1 w_2$ is a linear extension of $\varphi(u)$ and since $x \leqq_{\varphi(u)} y$, we have $x \leqq_{w_1 w_2} y$; on the other hand, since $y \leqq_w x$, we have $y \leqq_{w_1 w_2} x$, thus $x = y$, a contradiction with $x \nleqq_{\varphi(u)} y$.

Proof of 2: If $x \leqq_{\varphi(ua)} y$ then $x \leqq_{ua} y$; hence, if $y \in E_u$ then $x \in E_u$.

Finally let us prove the consistency property: for every $u \in A^*$, if v is a linear extension of $\varphi(u)$, then $v \sim u$. This is proved by induction on the length of u.

If u is the empty word, v is also the empty word.

Let $t = \varphi(ua)$ and $vb \in LE(t)$. We have already proved that φ is monotonic; hence, $t' = \varphi(u) \sqsubseteq t$. If $a = b$, then $E_u = E_v$ and v is a linear extension of t'. By induction hypothesis, $u \sim v$, and, by P2, $ua \sim va = vb$. If $a \neq b$ we have $u = u_1 b u_2$ with b not occurring in u_2 and $v = v_1 a v_2$ with a not occurring in v_2.

It follows that $v_1 v_2 b$ is a linear extension of $\varphi(u)$: since $\varphi(u) \sqsubseteq \varphi(ua)$, we have $\leqq_{\varphi(u)} = \leqq_{\varphi(ua)} \cap (E_u \times E_u)$ which is included in $\leqq_{vb} \cap (E_u \times E_u)$, and, since $E_u = E_{v_1 v_2 b}$, this is equal to $\leqq_{v_1 v_2 b}$. By induction hypothesis, we get

$$v_1 v_2 b \sim u = u_1 b u_2, \tag{2}$$

and, by Lemma 3.2,

$$v_1 v_2 \sim u_1 u_2. \tag{3}$$

Thus we get

$$v_1 v_2 ba \sim ua \sim u_1 u_2 ba. \tag{4}$$

Since $vb = v_1 a v_2 b$ is a linear extension of $t = \varphi(ua)$, one cannot have $x_b \leqq_t x_a$ (where x_a and x_b are the last occurrences of a and b in t); this implies that there exists $w \sim ua$ such that $x_a \leqq_w x_b$; therefore, $w = w_1 a w_2 b w_3$ with b not occurring in w_3 and a not occurring in $w_2 w_3$. Hence,

$$w = w_1 a w_2 b w_3 \sim ua. \tag{5}$$

By Lemma 3.2 we get, from (5),

$$w_1 w_2 b w_3 \sim u. \tag{6}$$

From (2) and (6), and by Lemma 3.2 we get

$$v_1 v_2 \sim w_1 w_2 w_3. \tag{7}$$

From (4), (5), and (7), we get

$$w_1 a w_2 b w_3 \sim w_1 w_2 w_3 ba. \tag{8}$$

By property P4, we have

$$w \sim w_1 w_2 w_3 ab. \tag{9}$$

Hence, $\varphi(ua) = \varphi(w_1 w_2 w_3 ab)$ and vb is a linear extension of $\varphi(w_1 w_2 w_3 ab)$; now we are in the previous case: $v \sim w_1 w_2 w_3 a$ and $vb \sim w_1 w_2 w_3 ab \sim ua$. □

Definition 3.1: If t is a P-trace in a CCI set P of P-traces over an alphabet A, and if w is a word of A^*, we denote by $t.w$ the P-trace $\varphi(uw)$ where u is any linear extension of t. By P2, the result is indeed independent of the choice of u in $LE(t)$. □

In particular, if $t \sqsubseteq t'$, there exists a word w such that $t' = t.w$.

LEMMA 3.4: *Let t_1 and t_2 be two prefixes of t, let t_3 be their intersection and t_4 their union, as defined in the previous section. Let v_1 and v_2 be the words such that $t_1 = t_3 . v_1$ and $t_2 = t_3 . v_2$. Then* $\mathrm{alph}(v_1) \cap \mathrm{alph}(v_2) = \varnothing$ *and* $t_4 = t_3 . v_1 v_2 = t_3 . v_2 v_1$.

Proof: Let us assume that v_1 and v_2 both contain a letter a. Let n be the number of a's in t_3; then $\langle a, n+1 \rangle$ is in $E_{t_1} \cap E_{t_2} = E_{t_3}$, a contradiction.

Let u be a linear extension of t_3; we have to prove that both $uv_1 v_2$ and $uv_2 v_1$ are linear extensions of t_4. Let us assume that one of them, say $w = uv_1 v_2$, is not. Since w is totally ordered, this implies that there exist $x \neq y$ in $E_{t_4} = E_{t_1} \cup E_{t_2} = E_w$ such that $x \leqq_{t_4} y$ and $y \leqq_w x$. These elements x and y cannot be both in $E_{t_1} = E_{uv_1}$ or in $E_{t_2} = E_{uv_2}$, because in these cases $x \leqq_w y$. Therefore, we must have $x \in E_{t_1} - E_u$ and $y \in E_{t_2} - E_u$. Since $x \leqq_t y$ and $y \in E_{t_2}$ with $t_2 \sqsubseteq t$, we have $x \in E_{t_1}$; hence, $x \in E_{t_1} \cap E_{t_2} = E_{t_3} = E_u$, a contradiction. □

Now we introduce an example which will be used also later on.

Example 3.1: Let $A = \{a, b, c\}$ be an alphabet. Let us consider the least right semi-congruence $\sim$ over A^* generated by all pairs $\langle uab, uba \rangle$ and $\langle uba, uab \rangle$ such that u is a word of A^* having an even number of c's. It is left as an exercise to the reader to prove that this semi-congruence satisfies properties P1-4 and, thus, defines a CCI set P_{ex}.

Let us remark that this CCI set P_{ex} is defined in a way very similar to a partially commutative monoid: two words are equivalent if one of them can be derived from the other one by transposing some consecutive letters. The difference with the partially commutative monoid resides in the fact that for the partially commutative monoid this transposition can be performed at every location in the word, whilst in this example the possibility of performing this transposition depends on the prefix of the word preceding this location. □

4. PROJECTIVE CCI SETS OF *P*-TRACES

We are now going to define some particular CCI sets which can be defined from subsets of a partially commutative monoid. Let us recall the definitions given in example 2.2.

Let A be some alphabet and θ be a symmetric irreflexive relation over A. Let $\sim_\theta$ be the congruence relation over A^* generated by $ab \sim_\theta ba$ for all pairs $\langle a, b \rangle$ in θ. It is very easy to check that $\sim_\theta$ satisfies P1, P2, P3, P4; hence, the partially commutative monoid is a special case of CCI set. (We will come back to this fact later on.)

Let us consider now a mapping π from A into B such that $\pi(a)=\pi(b) \Rightarrow \langle a, b \rangle \notin \theta$, and its canonical extension, denoted also by π, from A^* to B^*.

PROPOSITION 4.1: *If L is a subset of A^* closed under $\sim_\theta$* (i.e., *$u \sim_\theta v$ and $u \in L$ imply $v \in L$) and under prefix* (i.e., *if $v \in L$ and $u \sqsubseteq v$ then $u \in L$), and if π is a bijection between L and B^*, then the relation $\sim$ over B^*, defined by $u \sim v$ if and only if $\pi^{-1}(u) \cap L \sim_\theta \pi^{-1}(v) \cap L$, is an equivalence relation which satisfies* P1, P2, P3, *and* P4.

Proof: The relation $\sim$ is obviously an equivalence relation. Since $\sim_\theta$ is included in the Parikh-equivalence and since $\pi(\pi^{-1}(u))=u$, $\sim$ is also included in the Parikh-equivalence.

Let u, v, ua, va be words in B^* and u', v', $u''a_1, v''a_2$ their images in L under π^{-1}. Since L is closed under prefix, u'' and v'' are also in L. Since $\pi(u')=\pi(u'')=u$ and $\pi(v')=\pi(v'')=v$, and since π is injective on L, we get $u'=u''$ and $v'=v''$. If $u \sim v$, then, by definition, $u'' \sim_\theta v''$; hence, $u''a_1 \sim_\theta v''a_1$, and since $u''a_1 \in L$, we have also $v''a_1 \in L$. But $\pi(v''a_1)=\pi(v''a_2)=va$; hence, $a_1=a_2$ and by definition of $\sim$, $ua \sim va$. Conversely, if $ua \sim va$, then $u''a_1 \sim_\theta v''a_2$. Since a_1 and a_2 do not commute because their images under π are equal, we must have $a_1=a_2$. Hence, $u'' \sim_\theta v''$ and thus $u \sim v$.

P3 and P4 can be proved in the same way: the proof is left as an exercise for the reader. □

DEFINITION 4.1: A CCI set P is said to be *projective* if its associated equivalence $\sim$ satisfies the hypothesis of Proposition 4.1. If, moreover the language L used in this hypothesis is recognizable, then P is said to be *r-projective*. □

Example 4.1: Let us consider the two alphabets $B=\{a, b, a', b', c\}$ and $A=\{a, b, c\}$. Let $\pi: B \to A$ be defined by

$$\pi(a)=\pi(a')=a$$

$$\pi(b)=\pi(b')=b$$

$$\pi(c)=c.$$

Let $\theta=\{\langle a, b\rangle, \langle b, a\rangle\}$ be a commutation relation over B, and let $L=(\{a, b\}^* c \{a', b'\}^* c)^*$ be a recognizable language of B^*. The language Pref (L) consisting of all the prefixes of L is still recognizable and satisfies the hypothesis of Proposition 4.1. It is easy to check that the equivalence relation $\sim$ over A^* defined by this proposition is exactly the right semi-congruence defined in Example 3.1. Thus, P_{ex} is a r-projective CCI set. $\square$

It is an open question to know wether every CCI set is projective. We now define a family of CCI sets for which this property holds. We shall characterize later on, in Theorem 6.16, the CCI sets which are r-projective.

We show that a CCI set P of P-traces is projective whenever the equivalence $\sim$ associated with it satisfies the property M stated here.

(*M*) *For all letters a, b, c with $a \neq b$ and $a \neq c$, and for any word u, $uab \sim uba$ and $uac \sim uca$ if and only if $uabc \sim ubca$.*

Let us remark that this property is not always satisfied and that if $uabc \sim ubca$, Lemma 3.2 allows to deduce only $uab \sim uba$. On the other hand, any partially commutative monoid obviously satisfies this condition. (But a partially commutative monoid is also obviously r-projective!).

For every word u and every letter a we define the set

$$\Gamma_u(a)=\{b \neq a \mid uab \sim uba\}.$$

We consider the new alphabet $B=A \times \mathcal{P}(A)$ and the projection π from B into A. We define also the commutation relation θ by $\langle\langle a, X\rangle, \langle b, Y\rangle\rangle \in \theta$ if and only if (i) $a \neq b$, (ii) $a \in Y$ and $b \in X$. From (i) we get that for a', $a'' \in B$, $\pi(a')=\pi(a'') \Rightarrow \langle a', a''\rangle \notin \theta$.

Now let us consider the sequential mapping $\sigma: A^* \to B^*$ defined by

- $\sigma(\varepsilon)=\varepsilon$;
- $\sigma(ua)=\sigma(u)\langle a, \Gamma_u(a)\rangle$.

Obviously π is a bijection between the set $L=\sigma(A^*)$ and A^*. Moreover, L is closed under prefix, due to the definition of σ. We prove the following lemma which shows that in this case the CCI set is the image of a subset of a partially commutative monoid.

LEMMA 4.2: (i) *The set $L=\sigma(A^*)$ is closed under $\sim_\theta$.*

(ii) *For all u, $v \in A^*$, $u \sim v$ if and only if $\sigma(u) \sim_\theta \sigma(v)$.*

Proof: Firstly, let us establish some preliminary properties of Γ and σ.

(j) $$u \sim v \quad \Rightarrow \quad \Gamma_u(a)=\Gamma_v(a).$$

If $b \in \Gamma_u(a)$, then $uab \sim uba$; by P2 and because $u \sim v$, we get $vab \sim vba$; hence, $b \in \Gamma_v(a)$.

(jj) $u \sim v \;\Rightarrow\; \forall w, \;\; \exists w' : \sigma(uw) = \sigma(u)w'$ *and* $\sigma(vw) = \sigma(v)w'$.

By induction on the length of w: if w is the empty word, then w' is the empty word; if property (jj) is true for w, then $\sigma(uwa) = \sigma(u)w' \langle a, \Gamma_{uw}(a) \rangle$ and $\sigma(vwa) = \sigma(v)w' \langle a, \Gamma_{vw}(a) \rangle$; but $u \sim v$ implies, by P2, $uw \sim vw$, and, by (j), $\Gamma_{uw}(a) = \Gamma_{vw}(a)$.

(jjj) $$b \in \Gamma_w(a) \;\Rightarrow\; \Gamma_w(a) = \Gamma_{wb}(a).$$

Indeed, $c \in \Gamma_w(a)$ if and only if

$c \neq a$ and $wac \sim wca$ if and only if (using property M)

$c \neq a$ and $wabc \sim wbca$ if and only if (because $wba \sim wab$ and P2)

$c \neq a$ and $wbac \sim wbca$ if and only if

$c \in \Gamma_{wb}(a)$.

Now let us prove the first part of the second point of this lemma by induction on the length of u. Let us assume that $u \sim v$. If u is the empty word, v is also the empty word and $\sigma(u) = \sigma(v) = \varepsilon$. If $u = u'a$ and $v = v'a$, we have, by P2, $u' \sim v'$, and, by induction hypothesis, $\sigma(u') \sim_\theta \sigma(v')$; by definition of σ, $\sigma(u'a) = \sigma(u') \langle a, \Gamma_{u'}(a) \rangle$ and $\sigma(v'a) = \sigma(v') \langle a, \Gamma_{v'}(a) \rangle$. But, by (j), $u' \sim v'$ implies $\Gamma_{u'}(a) = \Gamma_{v'}(a)$; hence, $\sigma(u') \sim_\theta \sigma(v')$ implies $\sigma(u'a) \sim_\theta \sigma(v'a)$. Finally let us assume that $u'a \sim v'b$ with $a \neq b$. By P3, there exists w such that $u' \sim wb$ and $v' \sim wa$. By induction hypothesis $\sigma(u') \sim_\theta \sigma(w) \langle b, \Gamma_w(b) \rangle$ and $\sigma(v') \sim_\theta \sigma(w) \langle a, \Gamma_w(a) \rangle$. It follows that

$$\sigma(u'a) \sim_\theta \sigma(w) \langle b, \Gamma_w(b) \rangle \langle a, \Gamma_{wb}(a) \rangle \tag{10}$$

$$\sigma(v'b) \sim_\theta \sigma(w) \langle a, \Gamma_w(a) \rangle \langle b, \Gamma_{wa}(b) \rangle. \tag{11}$$

Since $u' \sim wb$, $v' \sim wa$, and $u'a \sim v'b$, we have also, using P2, $wba \sim wab$. Hence,

$$a \in \Gamma_w(b), \qquad b \in \Gamma_w(a),$$

and thus,

$$\langle \langle a, \Gamma_w(a) \rangle, \qquad \langle b, \Gamma_w(b) \rangle \rangle \in \theta.$$

By (jjj), we get

$$\Gamma_w(a) = \Gamma_{wb}(a), \qquad \Gamma_w(b) = \Gamma_{wa}(b). \tag{12}$$

Considering the equalities 12 and the definition of θ, from 10 and 11 we get $\sigma(u'a) \sim_\theta \sigma(v'b)$.

Now we prove the following property, from which the first point and the second part of the second point of the lemma can be immediately deduced:

if $\sigma(u) \sim_\theta w'$ then $w' = \sigma(\pi(w'))$ and $u \sim \pi(w')$,

which is proved by induction on the number of permutations of letters needed to go from $\sigma(u)$ to w'.

If $\sigma(u) = w'$ there is nothing to prove. Let us assume that $w' = w'_1 a' b' w'_2$ with $\langle a', b' \rangle \in \theta$ and that the induction hypothesis applies to $\sigma(u) \sim_\theta w'_1 b' a' w'_2$. Thus, we have $\pi(w'_1 b' a' w'_2) = w_1 baw_2$, $\sigma(w_1 baw_2) = w'_1 b' a' w'_2$, and $u \sim w_1 baw_2$. We want to prove that $w'_1 a' b' w'_2 = \sigma(w_1 abw_2)$ and $u \sim w_1 abw_2$.

By definition of σ, we have

$$w'_1 = \sigma(w_1)$$

$$b' = \langle b, \Gamma_{w_1}(b) \rangle$$

$$a' = \langle a, \Gamma_{w_1 b}(a) \rangle.$$

Thus, $w_1 ab \sim w_1 ba$ and, by P2, $w_1 abw_2 \sim w_1 baw_2 \sim u$. Also, we have $b \in \Gamma_{w_1}(a)$, $a \in \Gamma_{w_1}(b)$, and, by (jjj), we get

$$\Gamma_{w_1}(a) = \Gamma_{w_1 b}(a)$$

$$\Gamma_{w_1}(b) = \Gamma_{w_1 a}(b).$$

Therefore, $\sigma(w_1 ab) = \sigma(w_1) \langle a, \Gamma_{w_1}(a) \rangle \langle b, \Gamma_{w_1 a}(b) \rangle = w'_1 a' b'$, and, by (jj), $\sigma(w_1 abw_2) = w'_1 a' b' w'_2$. $\square$

5. P-ASYNCHRONOUS AUTOMATA

The notion of an *asynchronous automaton* has been introduced by Zielonka [18] to characterize the recognizable subsets of a partially commutative monoid. An asynchronous automaton reads words, but, since "independent" letters have independent effects on the state of the automaton, one can consider that it reads the "pomsets" associated with the congruence classes of the words. We extend this notion to a notion of automata reading P-traces.

Before defining the P-asynchronous automata, let us recall the definition of an asynchronous automaton.

DEFINITION 5.1: An *asynchronous automaton* over an alphabet A is an automaton $\langle Q, \delta, q_*, F, I, D \rangle$ defined in the following way.

- I is a set of *indices*.
- For each index $i \in I$, there is a finite set of states Q_i and $Q = \prod_{i \in I} Q_i$.
- If J is a subset of I, we denote by Q_J the product $\prod_{j \in J} Q_j$, and if q is an element of Q, q_J will be the element of Q_J consisting of the components of q having their index in J.
- D is a family of nonempty subsets $D(a)$ of I, for each letter a.
- For each letter a there is a mapping δ_a from $Q_{D(a)}$ into $Q_{D(a)}$.
- The transition function $\delta : Q \times A \to Q$ is defined as follows: $\delta(q, a)$ is the unique state q' such that:

  – $q'_{D(a)} = \delta_a(q_{D(a)})$,

  – $\forall j \notin D(a)$, $q'_j = q_j$.
- F is a subset of Q, the set of final states, and q_* is an element of Q, the initial state.

Now we extend this definition the following way. □

DEFINITION 5.2: A *P-asynchronous automaton* over an alphabet A is an automaton $\langle Q, \delta, q_*, F, I, D \rangle$ such that:

- I is a set of *indices* equal to the union of A and $\{1, \ldots, n\}$ for some n.
- For each index $i \in I$, there is a finite set of states Q_i and $Q = \prod_{i \in I} Q_i$.
- If J is a subset of I, we denote by Q_J the product $\prod_{j \in J} Q_j$, and if q is an element of Q, q_J will be the element of Q_J consisting of the components of q having their index in J.
- For each nonempty subset J of I and for each letter a there is a mapping δ_a^J from Q_J into Q_J.
- D is a family of mappings D_a from Q_a into $\mathscr{P}(I)$, for each letter a (remember that a is also an index), such that for each $q \in Q_a$, $D_a(q)$ contains the index a.
- The transition function $\delta : Q \times A \to Q$ is defined as follows: $\delta(q, a)$ is the unique state q' such that:

  – $q'_J = \delta_a^J(q_J)$;

  – $\forall j \notin J$, $q'_j = q_j$,

where $J = D_a(q_a)$.

• F is a subset of Q, the set of final states, and q_* is an element of Q, the initial state. □

Let us remark that if the function D_a is a constant function, for each a, we get the definition of an asynchronous automaton of Zielonka, setting $D(a)$ equal to the constant value of $D_a(q_a)$.

5.1. Zielonka's theorem

Let us recall here the statement of Zielonka's theorem [18]. The theorem 5.2, proved below, can be considered as an extension of this theorem to CCI sets.

Let L be the set of words accepted by an asynchronous automaton $\mathcal{A}=\langle Q, \delta, q_*, F, I, D\rangle$, and let $\theta_{\mathcal{A}}$ be the commutation relation defined by $\langle a, b\rangle \in \theta_{\mathcal{A}}$ if and only if $D(a) \cap D(b)=\varnothing$. Obviously, L is a recognizable language closed under the equivalence relation $\sim_{\theta_{\mathcal{A}}}$. Zielonka has also proved the converse of this property.

If L is a recognizable subset of A^* closed under the equivalence relation $\sim_\theta$ associated with a commutation relation θ over A, then there exists an asynchronous automaton $\mathcal{A}$ which accepts L and such that $\theta=\theta_{\mathcal{A}}$.

5.2. A-regular CCI sets of P-traces

In the same way as an asynchronous automaton defines a commutation relation and, therefore, a partially commutative monoid, a P-asynchronous automaton allows some letters to commute, but this commutation relation depends on the state of the automaton. Thus, intuitively, a P-asynchronous automaton defines a partial order on the occurrences of letters in the word it is reading. Hence, every P-asynchronous automaton $\mathcal{A}$ defines a CCI set of P-traces in a way which we shall make precise below. Indeed, as it will appear, this set does not depend on the set F of final states of $\mathcal{A}$.

Let $\mathcal{A}=\langle Q, \delta, q_*, F, I, D\rangle$ be a P-asynchronous automaton over an alphabet A. Let us consider the alphabet $B=A\times\mathcal{P}(I)$ and the canonical projection $\pi: B\rightarrow A$. We define over B the commutation relation $\theta=\{\langle\langle a, J\rangle, \langle b, J'\rangle\rangle \mid a\neq b, J\cap J'=\varnothing\}$. For each $q\in Q$, let us define the sequential mapping $\sigma_q: A^*\rightarrow B^*$ by

- $\sigma_q(\varepsilon)=\varepsilon$;
- $\sigma_q(ua)=\sigma_q(u).\langle a, D_a(q'_a)\rangle$, where $q'=\delta(q, u)$.

It is easy to see, from this definition, that

$$\sigma_q(uv)=\sigma_q(u)\,\sigma_{q'}(v)$$

where $q'=\delta(q, u)$.

Now let us set $\sigma=\sigma_{q_*}$, and, for each $q\in Q$, $L_q=\{\sigma(u) \mid \delta(q_*, u)=q\}$. Obviously, L_q is a recognizable subset of B^*, for each q.

Finally, let $L=\bigcup_{q\in Q} L_q$ which is also a recognizable subset of B^*. By definition, π is a bijection between L and A^* and L is closed under prefix. In order to apply Proposition 4.1, it remains to prove that L is closed under $\sim_\theta$. Indeed, we prove that every L_q is closed under this congruence.

LEMMA 5.1: *If* $\sigma(w_1\,abw_2)=w'_1\,a'\,b'\,w'_2\in L_{\bar{q}}$ *and if* $\langle a', b'\rangle\in\theta$, *then* $\sigma(w_1\,baw_2)=w'_1\,b'\,a'\,w'_2\in L_{\bar{q}}$.

Proof: Let us set

$$q=\delta(q_*, w_1)$$
$$q'=\delta(q_*, w_1\,a)$$
$$q''=\delta(q_*, w_1\,b)$$
$$\bar{q}'=\delta(q_*, w_1\,ab)$$
$$\bar{q}''=\delta(q_*, w_1\,ba).$$

We have, by definition of σ,

$$\sigma(w_1\,abw_2)=w'_1\,\langle a, D_a(q_a)\rangle\,\langle b, D_b(q'_b)\rangle\,\sigma_{\bar{q}'}(w_2)$$
$$\sigma(w_1\,baw_2)=w'_1\,\langle b, D_b(q_b)\rangle\,\langle a, D_a(q''_a)\rangle\,\sigma_{\bar{q}''}(w_2),$$

and it suffices to show that $q_a=q''_a$, $q_b=q'_b$, and $\bar{q}'=\bar{q}''$.

Since $\langle a', b'\rangle\in\theta$, with $a'=\langle a, D_a(q_a)\rangle$ and $a'=\langle b, D_b(q_b)\rangle$, the intersection of $D_a(q_a)$ and $D_b(q_b)$ is empty. Since $b\in D_b(q_b)$, $b\notin D_a(q_a)$ and from the definition of $q'=\delta(q_*, w_1\,a)=\delta(q, a)$ we get $q'_b=q_b$. For symmetric reasons, $q''_a=q_a$.

Now let us set

$$J_a=D_a(q_a)=D_a(q''_a)$$
$$J_b=D_b(q_b)=D_b(q'_b)$$
$$J=I-(J_a\cup J_b).$$

From the definition of δ we get $\bar{q}'=\bar{q}''$, considering the three cases where an index i belongs to J_a, J_b, or J. □

Now we define the equivalence relation $\sim$ over A^* by $u \sim v$ if and only if $\sigma(u) \sim_\theta \sigma(v)$, and, by Proposition 4.1, this equivalence defines an r-projective CCI set denoted by $P(\mathscr{A})$.

DEFINITION 5.3: A CCI set P of P-traces is said to be *a-regular* if it is equal to $P(\mathscr{A})$ for some P-asynchronous automaton $\mathscr{A}$. □

As an example of this definition, let us remark that any partially commutative monoid is an a-regular CCI set, since the asynchronous automaton recognizing the whole monoid is also a P-asynchronous automaton, and the sequential mapping σ, associated with it, is the identity. Another example of an a-regular CCI set is P_{ex} of Examples 3.1 and 4.1.

Example 5.1: Let $\mathscr{A}$ be the P-asynchronous automaton defined by

- the set of indices I is equal to $A=\{a, b, c\}$;
- $Q_a=\{q_a, q'_a\}$, $Q_b=\{q_b, q'_b\}$, $Q_c=\{q_c, q'_c\}$,
- $q_*=\langle q_a, q_b, q_c\rangle$,
- $D_a(q_a)=\{a\}$, $D_b(q_b)=\{b\}$; $D_a(q'_a)=D_b(q'_b)=\{a, b\}$;

$D_c(q_c)=D_c(q'_c)=\{a, b, c\}$;

- for any subset J of I, δ^J_a and δ^J_b are identities;
- $\delta^I_c(\langle q_1, q_2, q_3\rangle)=\begin{cases} \langle q'_a, q'_b, q'_c\rangle & \text{if } q_3=q_c \\ \langle q_a, q_b, q_c\rangle & \text{if } q_3=q'_c. \end{cases}$

This automaton can take only two states when reading a word, q_* and $q'=\langle q'_a, q'_b, q'_c\rangle$. The first one is reached when reading a word with an even number of c's, and the second one is reached when reading a word with an odd number of c's. Moreover, a and b commute only in state q_*. Hence, the mapping σ defined above, applied to A^*, gives exactly the language introduced in Example 4.1 to show that P_{ex} is r-projective. □

5.3. Recognizable subsets of an a-regular CCI set

Let P be an a-regular CCI set of P-traces over an alphabet A.

DEFINITION 5.4: A subset K of P is said to be *recognizable* if the set $[K]=\bigcup_{t \in K} LE(t)$ is a recognizable subset of A^*. □

The following theorem explains in which sense recognizable subsets of an a-regular CCI set are recognized by P-asynchronous automata.

THEOREM 5.2: *If R is a subset of A^* recognized by a P-asynchronous automaton $\mathcal{A}$, then $R=[K]$ for some recognizable subset K of the* CCI *set* $P(\mathcal{A})$.

If a subset K of an a-regular CCI *set P of P-traces is recognizable, then there exists a P-asynchronous automaton $\mathcal{A}$ which recognizes $[K]$ (as a set of words) and such that $P=P(\mathcal{A})$.*

Proof: Let $\mathcal{A}=\langle Q, \delta, q_*, F, D, I\rangle$ be a P-asynchronous automaton which recognizes a subset R of A^* and such that $P=P(\mathcal{A})$. Thus $\sigma(R)=\bigcup_{q\in F} L_q$, and, by Lemma 5.1, $\sigma(R)$ is closed under $\sim_\theta$; hence, $R=\pi(\sigma(R))$ is closed under $\sim$. Thus $R=[K]$ for some subset K of P.

Let $\mathcal{A}=\langle Q, \delta, q_*, F, D, I\rangle$ be a P-asynchronous automaton such that $P=P(\mathcal{A})$ and let K be a subset of P such that $[K]$ is a recognizable subset of A^*. Let us consider the sequential mapping $\sigma: A^*\to B^*$ previously defined. The subset $L=\sigma([K])$ of B^* is recognizable and is closed under $\sim_\theta$ by Lemma 5.1 and by definition of $\sim$. Thus, by Zielonka's theorem, there exists an asynchronous automaton $\mathcal{B}=\langle S, \delta', s_*, G, D', I'\rangle$ which recognizes L.

Let us define the P-asynchronous automaton $\mathcal{C}$ over A as a kind of product of $\mathcal{A}$ and B:

- The set of indices of $\mathcal{C}$ is the disjoint union of I and I'.
- The set of states of $\mathcal{C}$ is thus $Q\times S$; the initial state is $\langle q_*, s_*\rangle$ and the set of final states is the set of all states r such that $r_{I'}$ is in G.
- The mapping D''_a is defined by $D''_a(q)=D_a(q)\cup D'(\langle a, D_a(q)\rangle)$.
- Let a be a letter, J be a subset of the disjoint union $I\cup I'$; let us set $J_1=J\cap I$ and $J_2=J\cap I'$. We define $\delta''^{J}_a(r_J)$ by:

  – $\langle \delta^{J_1}_a(r_{J_1}), \delta'_{\langle a, J_1\rangle}(r_{J_2})\rangle$ if $J_2=D'(\langle a, J_1\rangle)$;

  – r_J otherwise.

The idea behind the construction of this product is the following: the first component of this product is $\mathcal{A}$ and works on an input word u of A^* exactly as $\mathcal{A}$ does whilst, simultaneously, the second part works on $\sigma(u)$ exactly as $\mathcal{B}$ does. Thus it is not difficult to prove that $\mathcal{C}$ accepts the words u such that $\sigma(u)$ is accepted by $\mathcal{B}$. Hence $\mathcal{C}$ recognizes $\pi(L)=[K]$. Moreover, the sequential maping σ'' associated with $\mathcal{C}$ is from A^* into C^* where C is the set of all pairs $\langle a, J_1\cup J_2\rangle$ such that $J_2=D'(\langle a, J_1\rangle)$ which is in bijection with B. It is not difficult to see that σ and σ'' are identical up to this bijection. Finally let us consider two elements $a'=\langle a, J\rangle$ and $b'=\langle b, J'\rangle$ of B and their corresponding elements $a''=\langle a, J\cup D'(a')\rangle$ and $b''=\langle b, J'\cup D'(b')\rangle$ of C. If a' and b' commute (*i.e.*, if $J\cap J'=\varnothing$) then $D'(a')\cap D'(b')=\varnothing$;

hence, a'' and b'' commute. Conversely, if a'' and b'' commute, $J \cap J' = 0$, and a' and b' commute. Thus, $\mathscr{A}$ and $\mathscr{C}$ define one and the same a-regular CCI set. □

Let us remark that we cannot simply define a recognizable subset K of a CCI set P of P-traces by the condition: $[K]$ is a recognizable subset of A^*, as it is the case for a partially commutative monoid. With this definition, every CCI set P would be recognizable, since $[P] = A^*$, and we will see in the next section that there are CCI sets which are not accepted by P-asynchronous automata. Indeed, our definition is consistent with the definiton of a recognizable subset of a partially commutative monoid, since every partially commutative monoid is an a-regular CCI set.

6. REGULAR CCI SETS AND THEIR CHARACTERIZATION

In the next section we shall exhibit necessary and sufficient conditions on the equivalence $\sim$ associated with a CCI set of P-traces for this set to be a-regular.

The first condition is that the least right semi-congruence containing $\sim$ is of finite index.

Let us consider a CCI set P over an alphabet A, $\sim$ its associated equivalence, and let us define the equivalence relation $\equiv$ over A^* by $u \equiv v$ if and only if

$$\forall w, w', \quad uw \sim uw' \qquad \text{if and only if} \quad vw \sim vw'.$$

The equivalence $\equiv$ is obviously a right semi-congruence greater than $\sim$.

Definition 6.1: A CCI set P is said to be *regular* if the right semi-congruence $\equiv$ is of finite index. □

In fact, $\equiv$ can be considered as the syntactic right semi-congruence of P. This semi-congruence can be used to characterize CCI sets which are partially commutative monoids.

Proposition 6.1: *A* CCI *set is a partially commutative monoid if and only if the equivalence* $\equiv$ *has only one equivalence class.*

Proof: If P is a partially commutative monoid (*i.e.*, $\sim \, = \, \sim_\theta$), then $uw \sim uw'$ if and only if $w \sim w'$; hence, every word u is $\equiv$-equivalent to the empty word.

Conversely, if every word u is $\equiv$-equivalent to the empty word then we have $uw \sim uw'$ if and only if $w \sim w'$. In particular, for every pair (a, b) of distinct letters, $uab \sim uba$ if and only if $ab \sim ba$. Let us define the commutation relation θ by $\langle a, b \rangle \in \theta$ if and only if $ab \sim ba$. It remains to prove that $\sim = \sim_\theta$. This is proved by induction on the length of the words.

– $\varepsilon \sim v$ if and only if $\varepsilon = v$ if and only if $\varepsilon \sim_\theta v$;

– $ua \sim va$ if and only if $u \sim v$ if and only if $u \sim_\theta v$ if and only if $ua \sim_\theta va$;

– $ua \sim vb$ if and only if $u \sim wb$, $v \sim wa$, and $wab \sim wba$ if and only if $u \sim_\theta wb$, $v \sim_\theta wa$, and $ab \sim_\theta ba$ if and only if $ua \sim_\theta vb$. □

Since every a-regular set is r-projective, the following proposition proves that every a-regular CCI set is regular.

PROPOSITION 6.2: *Every r-projective* CCI *set of P-traces is regular.*

Proof: Let L be the recognizable subset of which P is the projection under π. Let us denote by $\sigma(u)$ the unique word in L such that $u = \pi(\sigma(u))$. Let $\equiv_L$ be the syntactic right semi-congruence of L defined by $u \equiv_L v$ if and only if $\forall w,\ uw \in L \Leftrightarrow vw \in L$, which is of finite index. We show that if $\sigma(u) \equiv_L \sigma(v)$ then $u \equiv v$, from which it follows that $\equiv$ is of finite index.

Let us assume that $\sigma(u) \equiv_L \sigma(v)$ and that $uw_1 \sim uw_2$. Then $\sigma(uw_1) \sim_\theta \sigma(uw_2)$ with $\sigma(uw_1) = \sigma(u) w'_1$, $\sigma(uw_2) = \sigma(u) w'_2$, $w_1 = \pi(w'_1)$, and $w_2 = \pi(w'_2)$. It follows that $w'_1 \sim_\theta w'_2$. Since $\sigma(u) w'_1$ and $\sigma(u) w'_2$ are both in L, and since $\sigma(u) \equiv_L \sigma(v)$, $\sigma(v) w'_1$ and $\sigma(v) w'_2$ are also both in L. Moreover, $\sigma(v) w'_1 \sim_\theta \sigma(v) w'_2$. Thus, $vw_1 = \pi(\sigma(v) w'_1) \sim \pi(\sigma(v) w'_2) = vw_2$, and the result is proved. □

Let us give an example of CCI set which is not regular.

Example 6.1: Let us consider the alphabets $A = \{a, b, a', b', c\}$ and $B = \{a, b, c\}$, the mapping $\pi: A \to B$ defined by

$$\pi(x) = \begin{cases} a & \text{if } x = a' \\ b & \text{if } x = b' \\ x & \text{otherwise.} \end{cases}$$

Let us define the commutation relation $\theta = \{\langle a, b \rangle\}$ over A^*. Let D be the Dyck language over $\{a, b\}^*$ and $\bar{D}$ its complement, *i.e.*, $D = \{u \in \{a, b\}^* \mid |u|_a = |u|_b\}$ and $\bar{D} = \{u \in \{a, b\}^* \mid |u|_a \neq |u|_b\}$. Let L be equal to the set of prefixes of

$$\bigcup_{u \in \bar{D}} (D c)^* u c \{a', b', c\}^*.$$

It is easy to see that L is closed under prefix and closed under $\sim_\theta$. Moreover, $\pi(L)=B^*$ and π is a bijection between $\pi(L)$ and B^*. Hence, the language L defines a CCI set over B.

The $\equiv$-equivalence class of ε, with respect to this CCI set, is $(Dc)^*$. This set is not recognizable; hence, $\equiv$ cannot be of finite index. □

Now we shall prove that every regular CCI set is r-projective. We shall proceed as in Section 4, by constructing a sequential mapping σ which will be regular. In order to define σ, we need to associate with every word u and every letter a some information ranging over a finite domain, which will play the role of $\Gamma_u(a)$ in the construction of σ in section 4. The definition of this information involves techniques used by Zielonka in his proof, but some of the notions introduced by Zielonka [18] and also by Cori and Métivier [2] have to be adapted to the case of CCI sets.

From now on, let P be a CCI set of P-traces such that the semi-congruence $\equiv$ is of finite index.

6.1. The syntactic congruence of P

Firstly, we define another equivalence relation over words, related to a CCI set P, which can be considered as the syntactic congruence of P. The congruence class of a word u will play the same role as does the set of letters occurring in u in Zielonka's proof. Given a CCI set of P-traces over A characterized by the equivalence $\sim$, and the associated right semi-congruence defined above, we define the following equivalence relation $\approx$ over A*.

DEFINITION 6.2: Given two words u and v of A^*, $u \approx v$ if and only if

1. $\mathrm{alph}(u)=\mathrm{alph}(v)$;
2. $\forall w,\ wu \equiv wv$,
3. $\forall w, w', w''$ such that $\mathrm{alph}(w'') \cap \mathrm{alph}(uw')=\varnothing$, $wuw'w'' \sim ww''uw'$ if and only if $wvw'w'' \sim ww''vw'$. □

PROPOSITION 6.3: *The equivalence $\approx$ is a congruence. If $\equiv$ is of finite index, so is $\approx$.*

Proof: The fact that $\approx$ is a congruence is an immediate consequence of its definition.

The equivalence defined by the point 1 of the definition of $\approx$ is obviously of finite index. The equivalence defined by the point 2 is also of finite index when $\equiv$ is of finite index. Thus we consider only the point 3 of the definition.

Let us set, for two words w and u,

$$\mathrm{K}(w, u)=\{\langle w', w'' \rangle \mid \mathrm{alph}(w'') \cap \mathrm{alph}(uw')=\varnothing,\ wuw'w'' \sim ww''uw'\}.$$

Then the point 3 of the definition of $\approx$ is equivalent to

$$\forall w, \quad K(w, u)=K(w, v)$$

and thus we need to prove that $K(w, u)$ can take only a finite number of different values when w and u range over A^*. Firstly, it follows from the definition of K that $w \equiv w' \Rightarrow K(w, u)=K(w', u)$ and since $\equiv$ is of finite index, it remains to show that, for a fixed w, $K(w, u)$ takes a finite number of values when u ranges over A^*.

Let us also define $G(w, u)=\{w' \mid \langle \varepsilon, w' \rangle \in K(w, u)\}=\{w' \mid wuw' \sim ww'u\}$. We prove that every set $K(w, u)$ is the intersection of a finite number of sets of the form $K(w, u_i)$ or $A^* \times G(w, u_i)$ where the words u_i have their length bounded by some integer N we are going to define. Since the number of words of length less than N is finite, $K(w, u)$ can take only a finite number of values.

In all the sequel, w will be a fixed word. Since $\equiv$ is of finite index, there exists an integer N such that any word u of length greater than N can be written $u=u_1 u_2 u_3$ with $u_2 \neq \varepsilon$, $u_3 \neq \varepsilon$, and $wu_1 \equiv wu_1 u_2$.

If u has this form we will prove that

$$K(w, u_1 u_2 u_3)=K(w, u_1 u_3) \cap A^* \times G(w, u_1 u_2) \tag{13}$$

from which we deduce

$$G(w, u_1 u_2 u_3)=G(w, u_1 u_3) \cap G(w, u_1 u_2) \tag{14}$$

and we get the previously claimed property.

Proof of 13: If $\langle w', w'' \rangle \in K(w, u_1 u_2 u_3)$, then

$$wu_1 u_2 u_3 w' w'' \sim ww'' u_1 u_2 u_3 w' \tag{15}$$

with $\mathrm{alph}(w'') \cap \mathrm{alph}(u_1 u_2 u_3 w')=\varnothing$. By Lemma 3.2 we get

$$wu_1 u_2 w'' \sim ww'' u_1 u_2, \tag{16}$$

that is to say, $w'' \in G(w, u_1 u_2)$. From 16, we also get

$$wu_1 u_2 w'' u_3 w' \sim ww'' u_1 u_2 u_3 w' \sim wu_1 u_2 u_3 w' w''$$

and, since $wu_1 \equiv wu_1 u_2$, this implies

$$wu_1 w'' u_3 w' \sim wu_1 u_3 w' w''. \tag{17}$$

Applying again Lemma 3.2 to 16, we get

$$wu_1 w'' \sim ww'' u_1$$

and 17 becomes

$$ww'' u_1 u_3 w' \sim wu_1 u_3 w' w'',$$

that is to say, $\langle w', w'' \rangle \in K(w, u_1 u_3)$.

Conversely, let us assume that $\langle w', w'' \rangle \in K(w, u_1 u_3)$ and $w'' \in G(w, u_1 u_2)$; we get

$$wu_1 u_3 w' w'' \sim ww'' u_1 u_3 w' \tag{18}$$

$$wu_1 u_2 w'' \sim ww'' u_1 u_2. \tag{19}$$

From 18, by Lemma 3.2, we get

$$wu_1 w'' \sim ww'' u_1$$

hence, from 18

$$wu_1 u_3 w' w'' \sim wu_1 w'' u_3 w'.$$

Since $wu_1 \equiv wu_1 u_2$, this is equivalent to

$$wu_1 u_2 u_3 w' w'' \sim wu_1 u_2 w'' u_3 w'$$

and, using 19, we get

$$wu_1 u_2 u_3 w' w'' \sim ww'' u_1 u_2 u_3 w'$$

which means that $\langle w', w'' \rangle \in K(w, u_1 u_2 u_3)$.

Proof of 14: By definition, $w' \in G(w, u_1 u_2 u_3)$ if and only if $\langle \varepsilon, w' \rangle \in K(w, u_1 u_2 u_3) = K(w, u_1 u_3) \cap A^* \times G(w, u_1 u_2)$ (because of 13), if and only if $w' \in G(w, u_1 u_3)$ and $w' \in G(w, u_1 u_2)$. □

In the sequel, we denote by M the quotient monoid $A^*/\approx$, and by $u/\approx$ the congruence class of u, but in some cases, when the context makes it unambiguous, we will write u instead of $u/\approx$.

6.2. Some useful notions

Here we extend to the case of a regular CCI set of P-traces some notions introduced by Cori and Métivier [2] and Zielonka [18] in the case of a partially commutative monoid.

From now on we assume that a regular CCI set P is given.

DEFINITION 6.3: Let t be a P-trace and w be a word. A w-factorization of t is a pair $\langle t', v \rangle$ such that

- $t=t'.v$ and $\text{alph}(v) \cap \text{alph}(w)=\varnothing$;
- $t'.vw=t'.wv$. $\square$

Let us remark that, if $w \approx w'$ then, for every linear extension u of t', $uvw \sim uwv$ if and only if $uvw' \sim uw'v$; hence, $t'.vw=t'.wv$ if and only if $t'.vw'=t'.w'v$; thus a w-factorization of t is also a w'-factorization of t. Since this notion is independent of the choice of a representative in a congruence class m of M, we can as well define an m-factorization of t. And, by abuse of notation, we shall also write $t'.mv=t'.vm$, the second condition of this definition.

LEMMA 6.4: *If $\langle t', v \rangle$ is an m-factorization of t, and if $t=t'.v'$, then $\langle t', v' \rangle$ is also an m-factorization of t.*

Proof: We have $t=t'.v=t'.v'$, $t'.vm=t'.mv$, and $\text{alph}(v)=\text{alph}(v')$. We have to prove $t'.v'm=t'.mv'$. If it is not the case they have disjoint sets of linear extensions. Let u be a linear extension of t' and w an element of m. Then uwv' is a linear extension of $t'.mv'$. Let us assume that it is not a linear extension of $t''=t'.v'm=t'.vm=t'.mv$. Then there exist $x \neq y$ such that $x \leqq_{t''} y$ and $y \leqq_{uwv'} x$. We cannot have x and y both in E_{uw} or both in $E_{uv}=E_{uv'}$. Hence, $y \in E_{uw}-E_u$ and $x \in E_{uv'}-E_u=E_{uv}-E_u$. But, in this case, $y \leqq_{uwv} x$ and, since uwv is a linear extension of t'', we cannot have $x \leqq_{t''} y$, a contradiction. $\square$

LEMMA 6.5: *If $\langle t_1, v_1 \rangle$ and $\langle t_2, v_2 \rangle$ are two m-factorizations of t, then there exists a word v such that $\langle t_1 \sqcap t_2, v \rangle$ is also an m-factorization of t.*

Proof: Let $t'=t_1 \sqcap t_2$ and $t''=t_1 \sqcup t_2$. By Lemma 3.4, $t_1=t'.w_1$, $t_2=t'.w_2$, $t''=t'.w_1 w_2=t'.w_2 w_1$ and $\text{alph}(w_1) \cap \text{alph}(w_2)=\varnothing$. And, since $t'' \sqsubseteq t$, there exists v such that $t=t''.v$. By Lemma 6.4, $\langle t_1, w_2 v \rangle$ and $\langle t_2, w_1 v \rangle$

are two m-factorizations of t. Let u be a linear extension of t'. We get

$$uw_1 w_2 \sim uw_2 w_1$$
$$uw_1 w_2 vm \sim uw_1 mw_2 v$$
$$uw_2 w_1 vm \sim uw_2 mw_1 v.$$

Hence, by P2,

$$uw_1 mw_2 \sim uw_2 mw_1.$$

By applying Lemma 3.2, we get

$$uw_1 m \sim umw_1$$
$$uw_2 m \sim umw_2$$

and, thus,

$$uw_1 w_2 vm \sim uw_2 w_1 vm \sim umw_1 w_2 v \sim umw_2 w_1 v. \quad \square$$

Due to this lemma we can define the "least" m-factorization of t.

DEFINITION 6.4: If m is an element of M, and if t is a P-trace, we denote by $\partial_m(t)$ the least prefix t' of t such that there exists v such that $\langle t', v \rangle$ is an m-factorization of t. $\square$

In particular, if a is a letter, $\partial_a(t)$ is the prefix t' of t having as domain $E_{t'} = \{x \in E_t \mid x \leqq_{t.a} z\}$, where z is the last occurence of a in $t.a$.

The following properties will be very useful later on.

PROPOSITION 6.6: *Let $m, n \in M$ and let t be a P-trace such that $t.mn = t.nm$. Then $\partial_m(t.n) = \partial_m(t)$.*

For $m \in M$, $a \in A$, and $t \in P$, we have

$$\partial_m(t.a) = \begin{cases} \partial_m(t) & \text{if } \; t.am = t.ma \\ \partial_{am}(t).a & \text{otherwise.} \end{cases}$$

Proof: Let us assume that $t.mn = t.nm$. If $\langle t', v \rangle$ is an m-factorization of t, then $t = t'.v$ and $t'.vm = t'.mv$. Hence, $t.n = t'.vn$ and $t'.vnm = t'.vmn = t'.mvn$.

Thus, $\langle t', vn \rangle$ is an m-factorization of $t.n$. It follows

$$\partial_m(t.n) \sqsubseteq \partial_m(t).$$

Let $t_1=\partial_m(t.n)$ and $t_2=\partial_m(t)$. There exist w, v, and v', such that $t_2=t_1.w$, $\langle t_1, v\rangle$ is an m-factorization of $t.n$, and $\langle t_2, v'\rangle$ is an m-factorization of t. Then, $t=t_1.wv'$, $t.n=t_1.wv'n$, and, by Lemma 6.4, $t_1.mwv'n=t_1.wv'nm$. But $t.mn=t.nm$ and $t=t_1.wv'$ implies $t_1.wv'mn=t_1.wv'nm$. It follows, by P2, that $t_1.mwv'=t_1wv'm$; hence, $\langle t_1, wv'\rangle$ is an m-factorization of $t_1.wv'=t$ and $t_2 \sqsubseteq t_1$.

As a corollary, if $t.am=t.ma$, then $\partial_m(t.a)=\partial_m(t)$. Let us consider the case where $t.am \neq t.ma$. Let $\langle t', v\rangle$ be an am-factorization of t. We have $t=t'.v$ and $t'.amv=t'.vam$. By Lemma 3.2, $t'.av=t'.va$. Hence, $t'.amv=t'.avm$, and $\langle t'.a, v\rangle$ is a m-factorization of $t'.av=t'.va=t.a$. Thus,

$$\partial_m(t.a) \sqsubseteq \partial_{am}(t).a.$$

Now, let us prove that, if $\langle t', v\rangle$ is an m-factorization of $t.a$, then $t'=t''.a$ and $a\notin \mathrm{alph}(v)$.

Let us assume that $a\in \mathrm{alph}(v)$, that is to say, $v=v_1av_2$ with $a\notin\mathrm{alph}(v_2)$. Thus, $t.a=t'.v_1av2$, and, by Lemma 3.2, $t=t'.v_1v_2$, which implies $t.a=t'.v_1v_2a$. By Lemma 6.4, $t'.v_1v_2am=t'.mv_1v_2a$, thus, by Lemma 3.2, $t'.v_1v_2m=t'.mv_1v_2$, and then, $t.am=t'.v_1v_2am=t'.v_1v_2ma=t.ma$, a contradiction. Let u' be a linear extension of t'. Then $u'v$ is a linear extension of $t.a=t'.v$; hence $u'=u_1'au_2'$ with $a\notin\mathrm{alph}(u_2'v)$. But, if u is a linear extension of t, ua is a linear extension of $t.a$, thus, $ua\sim u_1'au_2'v$, and, by Lemma 3.2, $u_1'au_2'\sim u''a$. Hence, $u''a$ is a linear extension of t' which can, thus, be written as $t''.a$.

Now, let $\langle t''.a, v\rangle$ be an m-factorization of $t.a$. We have $t''.av=t.a$ with $a\notin\mathrm{alph}(v)$, and, by Lemma 3.2, $t''.v=t$, hence, $t''.va=t.a=t''.av$. Moreover, $t''.amv=t''.avm$. Hence, $t''.vam=t''.avm=t''.amv$, and $\langle t'', v\rangle$ is an am-factorization of $t''.v=t$.

Hence,

$$\partial_{am}(t) \sqsubseteq \partial_m(t.a). \quad \square$$

Proposition 6.7: *Let t be a P-trace and t' be a prefix of t. Let $t_0=t'\sqcap\partial_b(t)$ with $t'=t_0.u$ and $\partial_b(t)=t_0.v$. Then $t_0=\partial_{vb}(t')$.*

Proof: Let us set $t_1=\partial_{vb}(t')$ and let us prove that $t_0=t_1$. By Lemma 3.4, $t_0.uv=t_0.vu$. Since t' and $\partial_b(t)$ are both prefixes of t, their union is also a prefix of t. Thus, there exists w' such that $t=t_0.uvw'$. By definition of $t_0.v=\partial_b(t)$, we have

$$t_0.vuw'b=t_0.vbuw' \tag{20}$$

hence, by Lemma 3.2,

$$t_0.uvb=t_0.vub=t_0.vbu.$$

It follows, from this last equality, that $t_1 \sqsubseteq t_0$. Thus, $t_0=t_1.w$. Hence, $t'=t_1.wu$, and equality 20 becomes

$$t_1.wvuw'b=t_1.wvbuw'. \tag{21}$$

By definition of t_1,

$$t_1.wvub=t_1.wuvb=t_1.vbwu$$

and then

$$t_1.wvubw'=t_1.vbwuw'. \tag{22}$$

Applying Lemma 3.2 to equality 21, it becomes

$$t_1.wvub=t_1.wvbu$$

and, thus,

$$t_1.wvubw'=t_1.wvbuw'. \tag{23}$$

From equalities 21, 23, and 22, we get

$$t_1.wvuw'b=t_1.vbwuw' \tag{24}$$

and, applying Lemma 3.2 twice,

$$t_1.wvu=t_1.vwu.$$

Thus, equality 24 becomes

$$t_1.vwuw'b=t_1.vbwuw'.$$

Hence, $\langle t_1.v, wuw' \rangle$ is a b-factorization of t, and $t_1.wv \sqsubseteq t_1.v$. This implies $w=\varepsilon$, and thus, $t_0=t_1$. □

Now, we extend the definition of ∂ to subsets of M.

DEFINITION 6.5: Let L be a subset of M. For a P-trace t, we define $\partial_L(t)$ as the least prefix t' of $t=t'.v$ such that, for all $m \in L$, $\langle t', v \rangle$ is an m-factorization of t. $\square$

We have the following property.

LEMMA 6.8: $\partial_{L \cup L'}(t)=\partial_L(t) \sqcup \partial_{L'}(t)$.

Proof: Obviously, $\partial_L(t) \sqcup \partial_{L'}(t) \sqsubseteq \partial_{L \cup L'} L'(t)$. Conversely, let $t_1=\partial_L(t)$, $t_2=\partial_{L'}(t)$, $t_3=t_1 \sqcap t_2$, $t_4=t_1 \sqcup t_2$. By Lemma 3.4, $t_1=t_3.v_1$, $t_2=t_3.v_2$, $t_4=t_3.v_1 v_2=t_3.v_2 v_1$, and $\text{alph}(v_1) \cap \text{alph}(v_2)=0$. Now, $\partial_{L \cup L'}(t)=t_4.w$ and $t=\partial_{L \cup L'}(t).v=t_4.wv$. Since $t_1=\partial_L(t)$, we have

$$\forall m \in L, \quad t_3.v_1 v_2 wvm=t_3.v_1 mv_2 wv.$$

By Lemma 3.2, and, since $t_3.v_1 v_2=t_3.v_2 v_1$, we get

$$\forall m \in L, \quad t_3.v_1 v_2 m=t_3.v_1 mv_2$$

hence,

$$\forall m \in L, \quad t_3.v_1 v_2 wvm=t_3.v_1 v_2 mwv.$$

For similar reasons

$$\forall m \in L', \quad t_3.v_1 v_2 wvm=t_3.v_1 v_2 mwv.$$

It follows that

$$\forall m \in L \cup L', \quad t_4.wvm=t_4.mwv=t.m$$

and$\partial_{L \cup L'}(t) \sqsubseteq t_4=\partial_L(t) \cup \partial_{L'}(t)$. $\square$

Now, we are going to define the numbering of the occurrences of a P-trace, which plays a major role in Zielonka's proof. Here too, this numbering is the key tool of our proofs.

DEFINITION 6.6: For a P-trace t and a letter a, $\gamma_a(t)$ is the last occurrence of a in t if $a \in \text{alph}(t)$; otherwise, it is undefined. $\square$

Here are some properties of γ_a.

LEMMA 6.9: *If* $t \sqsubseteq t'$ *and* $\gamma_a(t') \in E_t$, *then* $\gamma_a(t)=\gamma_a(t')$.

For all $m \in M$ *and for any letter* a, $\gamma_a(\partial_{am}(t).a)=\gamma_a(t.a)$.

Proof: The first point of the lemma is an immediate consequence of the definition.

From the definition of $\partial_{am}(t)$, we get $t=\partial_{am}(t).v$ with

$$\partial_{am}(t).amv=\partial_{am}(t).vam.$$

Thus v does not contain the letter a, and the number of a's in $\partial_{am}(t)$ and in t are equal. □

DEFINITION 6.7: For each $t\in P$, and for each linear extension u of t, we define the mapping ν_t^u: $E_t \to \{1, \ldots, N+1\}$ (remember that N is the cardinal of M), by induction on the length of u.

- If t is the empty P-trace, and u the empty word, ν_t^u is the empty mapping.
- Let $u=u'a$, $t'=\varphi(u')$, so that $t=t'.a$, and let us assume that $\nu_{t'}^{u'}$ is already defined. Let $x\in E_t$; then $\nu_t^u(x)$ is defined as follows:

– $\nu_t^u(x)=\nu_{t'}^{u'}(x)$ if $x\in E_{t'}$;

– if $x\notin E_{t'}$, *i.e.*, $x=\gamma_a(t)$, let us consider the set $\{\nu_{t'}^{u'}(\gamma_a(\partial_m(\partial_a(t'))) \mid m\in M$ and $\gamma_a(\partial_m(\partial_a(t')))$ defined$\}$; the cardinality of this set is obviously less than or equal to N and $\nu_t^u(x)$ is the least element of $\{1, \ldots, N+1\}$ not belonging to this set. □

Indeed, the mapping ν_t^u does not depend on the choice of u, as shown by the following lemma, thus we shall denote simply by ν_t the mapping previously defined.

LEMMA 6.10: *If u and v are two linear extensions of t, then $\nu_t^u=\nu_t^v$.*

Proof: This result is proved by induction on the length of u.

If u is empty, there is nothing to prove.

If $u=u'a$ and $v=v'a$, then $t'=\varphi(u')=\varphi(v')$, and, by induction hypothesis, $\nu_{t'}^{u'}=\nu_{t'}^{v'}$; it follows immediately from the definition of ν_t^u that $\nu_t^u=\nu_t^v$.

If $u=u'a$ and $v=v'b$, with $a\neq b$, then $u'\sim wb$ and $v'\sim wa$. Let $t=\varphi(u)=\varphi(v)$; let $x_a=\gamma_a(t)$, $x_b=\gamma_b(t)$. Let $x\in E_t$. If $x\in E_w$, *i.e.* $x\neq x_a$ and $x\neq x_b$, $\nu_t^u(x)=\nu_{\varphi(u')}^{u'}(x)$ and $\nu_t^v(x)=\nu_{\varphi(v')}^{v'}(x)$. By induction hypothesis; $\nu_{\varphi(u')}^{u'}(x)=\nu_{\varphi(w)}^{w}=\nu_{\varphi(v')}^{v'}(x)$, hence, $\nu_t^u(x)=\nu_t^v(x)$. Now, $\nu_t^u(x_a)$ is the least element not in $\{\nu_{\varphi(u')}^{u'}(\gamma_a(\partial_m(\partial_a(\varphi(u'))))) \mid m\in M\}$. But $u'\sim wb$ with $wab\sim wba$, thus $\varphi(u')=\varphi(w).b$, $\varphi(w).ab=\varphi(w).ba$, and, by Proposition 6.6, $\partial_a(\varphi(u'))=\partial_a(\varphi(w))$. Since, for every $x\in E_w$, $\nu_{\varphi(u')}^{u'}(x)=\nu_{\varphi(w)}^{w}(x)$, $\nu_t^u(x_a)$ is the least element not in $\{\nu_{\varphi(w)}^{w}(\gamma_a(\partial_m(\partial_a(\varphi(w))))) \mid m\in M\}$. On the other hand, by definition, $\nu_t^v(x_a)=\nu_{\varphi(v')}^{v'}(x_a)$, which is equal, by induction hypothesis, to $\nu_{\varphi(wa)}^{wa}(x_a)$, which is, by definition, the least element not in

$\{\nu^w_{\varphi(w)}(\gamma_a(\partial_m(\partial_a(\varphi(w))))) \mid m \in M\}$. Hence, $\nu^u_t(x_a) = \nu^v_t(x_a)$. By similar reasoning, $\nu^u_t(x_b) = \nu^v_t(x_b)$, which completes the proof. □

The following property is an immediate consequence of the definition of ν_t.

PROPOSITION 6.11: *If $t \sqsubseteq t'$ and if $x \in E_t$, then* $\nu_t(x) = \nu_{t'}(x)$.

Now, we define the equivalence relation $\equiv_1$ over P.

DEFINITION 6.8: Let t and t' be two P-traces in P. We say that $t \equiv_1 t'$ if

(i) $t \equiv t'$,

(ii) $\forall m \in M$, $\partial_m(t) \equiv \partial_m(t')$,

(iii) $\forall a \in A$, $\nu_t(\gamma_a(t)) = \nu_{t'}(\gamma_a(t'))$,

(iv) $\forall a \in A$, $\forall m, n \in M$, $\nu_t(\gamma_a(\partial_m(\partial_n(t)))) = \nu_{t'}(\gamma_a(\partial_m(\partial_n(t'))))$. □

DEFINITION 6.9: By analogy with the case of words, we say that an equivalence relation R over a set of P-traces is a right semi-congruence if $t R t' \Rightarrow t.a R t'.a$, for any letter a. □

PROPOSITION 6.12: *The equivalence relation $\equiv_1$ is a right semi-congruence of finite index.*

Proof: Since $\equiv$ is of finite index, and since M is finite and ν_t has a finite domain, $\equiv_1$ is of finite index.

Let us assume that $t \equiv_1 t'$ and let us show that $t.a \equiv_1 t'.a$.

1. Since $t \equiv t'$, $t.a \equiv t'.a$.

2. Since $t \equiv t'$, $t.am = t.ma$ if and only if $t'.am = t'.ma$. Hence, by Proposition 6.6, either

$$\partial_m(t.a) = \partial_m(t) \quad \text{and} \quad \partial_m(t'.a) = \partial_m(t')$$

or

$$\partial_m(t.a) = \partial_{am}(t).a \quad \text{and} \quad \partial_m(t'.a) = \partial_{am}(t').a.$$

Since $\partial_n(t) \equiv \partial_n(t')$ for all $n \in M$, in both cases, $\partial_m(t.a) \equiv \partial_m(t'.a)$.

3. If $b \neq a$, then $\gamma_b(t.a) = \gamma_b(t)$; thus, $\nu_{t.a}(\gamma_b(t.a)) = \nu_{t.a}(\gamma_b(t)) = \nu_t(\gamma_b(t))$, and, similarly, $\nu_{t'.a}(\gamma_b(t'.a)) = \nu_{t'}(\gamma_b(t'))$.

Now, if $b = a$, $\nu_{t.a}(\gamma_a(t.a))$ is equal, by definition, to the least element not in the set $\{\nu_t(\gamma_a(\partial_m(\partial_a(t)))) \mid m \in M\}$. Since $t \equiv_1 t'$, by (iv), this set is equal to $\{\nu_{t'}(\gamma_a(\partial_m(\partial_a(t')))) \mid m \in M\}$, and $\nu_{t.a}(\gamma_a(t.a)) = \nu_{t'.a}(\gamma_a(t'.a))$.

4. Since $t \equiv_1 t'$, $t.an = t.na$ if and only if $t'.an = t'.na$. Hence, either

$$\partial_n(t.a) = \partial_n(t) \quad \text{and} \quad \partial_n(t'.a) = \partial_n(t')$$

or

$$\partial_n(t.a) = \partial_{an}(t).a \quad \text{and} \quad \partial_n(t'.a) = \partial_{an}(t').a.$$

In the first case, we get, for any b, possibly equal to a,

$$\begin{aligned}\nu_{t.a}(\gamma_b(\partial_m(\partial_n(t.a)))) &= \nu_{t.a}(\gamma_b(\partial_m(\partial_n(t)))) = \nu_t(\gamma_b(\partial_m(\partial_n(t))))\\ &= \nu_{t'}(\gamma_b(\partial_m(\partial_n(t')))) = \nu_{t'.a}(\gamma_b(\partial_m(\partial_n(t')))) = \nu_{t'.a}(\gamma_b(\partial_m(\partial_n(t'.a)))),\end{aligned}$$

and the result is obtained. In the second case,

$$\nu_{t.a}(\gamma_b(\partial_m(\partial_n(t.a)))) = \nu_{t.a}(\gamma_b(\partial_m(\partial_{an}(t).a)))$$

and

$$\nu_{t'.a}(\gamma_b(\partial_m(\partial_n(t'.a)))) = \nu_{t'.a}(\gamma_b(\partial_m(\partial_{an}(t').a))).$$

Since, by (ii), $\partial_{an}(t) \equiv \partial_{an}(t')$, we have either

$$\partial_m(\partial_{an}(t).a) = \partial_m(\partial_{an}(t)) \quad \text{and} \quad \partial_m(\partial_{an}(t').a) = \partial_m(\partial_{an}(t'))$$

and, in this case, we get the result, because of (iv), or

$$\partial_m(\partial_{an}(t).a) = \partial_{am}(\partial_{an}(t)).a \quad \text{and} \quad \partial_m(\partial_{an}(t').a) = \partial_{am}(\partial_{an}(t')).a.$$

In this case, if $b \neq a$, then $\gamma_b(\partial_{am}(\partial_{an}(t)).a) \in E_t$ and $\gamma_b(\partial_{am}(\partial_{an}(t')).a) \in E_{t'}$, and the result follows, by (iv). If $b = a$, then, by Lemma 6.9 $\gamma_a(\partial_{am}(\partial_{an}(t)).a) = \gamma_a(t.a)$ and $\gamma_a(\partial_{am}(\partial_{an}(t')).a) = \gamma_a(t'.a)$, and the result has been already proved in point 3 above. □

Now, we can prove an important result of this section.

PROPOSITION 6.13: *Let t and t' be such that*

$$\partial_a(t) \equiv_1 \partial_a(t')$$
$$\partial_b(t) \equiv_1 \partial_b(t'.a).$$

Then $t'.ab = t'.ba$.

Proof: Let us assume that $t'.ab \neq t'.ba$.

By Proposition 6.6,

$$\partial_b(t'.a) = \partial_{ab}(t').a.$$

Thus, by Proposition 6.9,

$$\gamma_a(\partial_b(t'.a))=\gamma_a(\partial_{ab}(t').a)=\gamma_a(t'.a)=\gamma_a(\partial_a(t').a). \tag{25}$$

The second hypothesis of this proposition implies, by definition of $\equiv_1$,

$$\nu_{\partial_b(t)}(\gamma_a(\partial_b(t)))=\nu_{\partial_b(t'.a)}(\gamma_a(\partial_b(t'.a)))$$

and, from equality 25,

$$\nu_{\partial_b(t)}(\gamma_a(\partial_b(t)))=\nu_{\partial_b(t'.a)}(\gamma_a(\partial_a(t').a))=\nu_{\partial_b(t'.a)}(\gamma_a(t'.a)).$$

Thus, using Proposition 6.9,

$$\nu_t(\gamma_a(\partial_b(t)))=\nu_{t'.a}(\gamma_a(\partial_a(t').a))=\nu_{t'.a}(\gamma_a(t'.a)). \tag{26}$$

Since $\equiv_1$ is a right semi-congruence, the first hypothesis of the proposition implies $\partial_a(t).a\equiv_1\partial_a(t').a$, hence, by definition of $\equiv_1$, and using Proposition 6.9,

$$\nu_{t.a}(\gamma_a(\partial_a(t).a))=\nu_{t'.a}(\gamma_a(\partial_a(t').a))=\nu_{t.a}(\gamma_a(t.a))=\nu_{t'.a}(\gamma_a(t'.a)). \tag{27}$$

From equalities 26 and 27, we get

$$\nu_t(\gamma_a(\partial_b(t)))=\nu_{t.a}(\gamma_a(t.a)). \tag{28}$$

Now, let $t_0=\partial_a(t)\sqcap\partial_b(t)$; thus,

$$\partial_a(t)=t_0.u,\ \partial_b(t)=t_0.v, \qquad \text{and} \qquad t=t_0.uvw=t_0.vuw.$$

By definition of $\partial_a(t)$, we have $t_0.uvwa=t_0.uavw$. Thus, $a\notin \text{alph}(v)$. It follows that $\gamma_a(t_0.v)\in E_{t_0}$, and, by Proposition 6.9, $\gamma_a(\partial_b(t))=\gamma_a(t_0.v)=\gamma_a(t_0)$. By Proposition 6.7, $t_0=\partial_{vb}(\partial_a(t))$, and equality 28 becomes

$$\nu_t(\gamma_a(\partial_{vb}(\partial_a(t))))=\nu_{t.a}(\gamma_a(t.a))$$

which is in contradiction with the definition of $\nu_{t.a}$. □

6.3. The construction of σ

The construction of the sequential mapping σ proceeds exactly as in section 4.

Let G be the finite set of equivalence classes of A^* for the semi-congruence $\equiv_1$. Let B be equal to $A\times G$. Let us define $\sigma: A^*\to B^*$ by

- $\sigma(\varepsilon)=\varepsilon$,

- $\sigma(ua)=\sigma(u)\langle a, \Gamma_u(a)\rangle$,

where $\Gamma_u(a)$ is the $\equiv_1$-equivalence class of $\partial_a(\varphi(u))$. Let us remark that if $u \sim v$, then $\varphi(uw)=\varphi(vw)$, and $\Gamma_{uw}(a)=\Gamma_{vw}(a)$.

Now, we define the commmutation relation θ over B by $\langle\langle a, g_a\rangle, \langle b, g_b\rangle\rangle \in \theta$ if and only if $a \neq b$ and there exists u such that $g_a=\Gamma_u(a)$ and $g_b=\Gamma_u(b)$.

It is clear that the projection $\pi: B^* \to A^*$ is a bijection between $\sigma(A^*)$ and A^* and that $\sigma(A^*)$ is closed under prefix. In order to prove that $\sigma(A^*)$ is closed under $\sim_\theta$ and that $u \sim v$ if and only if $\sigma(u) \sim_\theta \sigma(v)$, it is sufficient, for the same reasons as in the proof of Lemma 4.2, to prove the following lemma.

LEMMA 6.14: *If* $u \sim v$ *then* $\sigma(u) \sim_\theta \sigma(v)$. *If* $\sigma(u) \sim_\theta w$ *then* $w=\sigma(\pi(w))$ *and* $u \sim \pi(w)$.

Proof: Let us prove by induction on the length of u that $u \sim v \Rightarrow \sigma(u) \sim_\theta \sigma(v)$. If u is the empty word, this is obvious.

If $u=u'a$ and $v=v'a$, then $u' \sim v'$, $\Gamma_{u'}(a)=\Gamma_{v'}(a)=g$, and, by induction hypothesis, $\sigma(u') \sim_\theta \sigma(v')$. Hence, $\sigma(u)=\sigma(u')\langle a, g\rangle \sim_\theta \sigma(v')\langle a, g\rangle$.

If $u=u'a$ and $v=v'b$ with $a \neq b$, then there exists w such that $u' \sim wb$, $v' \sim wa$ and, by induction hypothesis, $\sigma(u') \sim_\theta \sigma(w)\langle b, \Gamma_w(b)\rangle$ and $\sigma(v') \sim_\theta \sigma(w)\langle a, \Gamma_w(a)\rangle$. Since $wab \sim wba$, by Proposition 6.6, we get $\partial_a(\varphi(wb))=\partial_a(\varphi(w))$ and $\partial_b(\varphi(wa))=\partial_b(\varphi(w))$. Thus $\Gamma_{wb}(a)=\Gamma_w(a)$ and $\Gamma_{wa}(b)=\Gamma_w(b)$. Moreover, $\langle\langle a, \Gamma_w(a)\rangle, \langle b, \Gamma_w(b)\rangle\rangle \in \theta$, thus, $\sigma(u) \sim_\theta (w)$

$$\langle b, \Gamma_w(b)\rangle\langle a, \Gamma_w(a)\rangle \sim_\theta \sigma(w)\langle a, \Gamma_w(a)\rangle\langle b, \Gamma_w(b)\rangle \sim_\theta \sigma(v).$$

To prove the second point of this proposition, we need only to prove, as in Lemma 4.2, that if $\sigma(u)=w\langle a, g_a\rangle\langle b, g_b\rangle$, with $\langle\langle a, g_a\rangle, \langle b, g_b\rangle\rangle \in \theta$, (and thus $a \neq b$), then $u \sim \pi(w)ba$ and $\sigma(\pi(w)ba)=w\langle b, g_b\rangle\langle a, g_a\rangle$. Indeed $u=w'ab$ with $w'=\pi(w)$. Thus, $g_a=\Gamma_{w'}(a)$, $g_b=\Gamma_{w'a}(b)$. By definition of θ, we have $g_a=\Gamma_v(a)$ and $g_b=\Gamma_v(b)$. Hence, $\partial_a(\varphi(v)) \equiv_1 \partial_a(\varphi(w'))$ and $\partial_b(\varphi(v)) \equiv_1 \partial_b(\varphi(w').a)$. Thus, by Proposition 6.13, $\varphi(w').ab=\varphi(w').ba$; hence, $u=w'ab \sim w'ba$. Finally $\sigma(w'ba)=\sigma(w')\langle b, \Gamma_{w'}(b)\rangle\langle a, \Gamma_{w'b}(a)\rangle$. Since $\pi(\sigma(w'))=w'=\pi(w)$, we have $w=\sigma(w')$. Since $w'ba=w'ab$, by Proposition 6.6, $\partial_a(\varphi(w'a))=\partial_a(\varphi(w').a)=\partial_a(\varphi(w'))$; hence, $\Gamma_{w'b}(a)=g_a$. □

Moreover, due to the following property, σ is a regular sequential mapping and therefore, $\sigma(A^*)$ is a recognizable language. This language can be recognized by an asynchronous automaton and we will exploit this fact in the construction of a P-asynchronous automaton defining P.

PROPOSITION 6.15: *The equivalence relation* $\equiv_2$ *defined by* $t \equiv_2 t'$ *if and only if*

- $t \equiv t'$;
- *For any* $m \in M$, $\partial_m(t) \equiv_1 \partial_m(t')$

is a right semi-congruence of finite index.

Proof: Since $\equiv$ and $\equiv_1$ are of finite index, and since M is finite, $\equiv_2$ is of finite index.

We already know that $\equiv$ is a right semi-congruence. Thus, we have only to prove $t \equiv_2 t' \Rightarrow \forall m \in M$, $\partial_m(t.a) \equiv_1 \partial_m(t'.a)$ for any letter a. Since $t \equiv t'$, $t.am \sim t.ma$ if and only if $t'.am \sim t'.ma$. Thus, by Proposition 6.6, either $\partial_m(t.a) = \partial_m(t)$ and $\partial_m(t'.a) = \partial_m(t')$, in which case the result is true, since $\partial_m(t) \equiv_1 \partial_m(t')$, or $\partial_m(t.a) = \partial_{am}(t).a$ and $\partial_m(t'.a) = \partial_{am}(t').a$, in which case the result is true too, since $\partial_{am}(t) \equiv_1 \partial_{am}(t')$ and since $\equiv_1$ is a right semi-congruence. □

As a conclusion of this section we can state the following result.

THEOREM 6.16: *A* CCI *set* P *of* P*-traces is regular if and only if it is* r*-projective.*

Proof: We have just shown that a regular set is r-projective. The other implication was proved in Proposition 6.2. □

7. A CHARACTERIZATION OF A-REGULAR CCI SETS

An a-regular CCI set is r-projective, and thus, by Proposition 6.2, it is also regular. We show that an a-regular CCI set also has the property Q defined below.

(Q) For all words u and v, and for all letters a and b, if there exist two words w and w' such that

$$uvwa \sim uwav \quad \text{and} \quad uvw'b \sim uw'bv$$

then, for all words w such that $uvw \sim uwv$, we have

$$uwab \sim uwba \quad \text{if and only if} \quad uvwab \sim uvwba. \quad \Box$$

It is left as an exercise for the reader to prove that the property M defined in section 4 is stronger than this property Q.

PROPOSITION 7.1: *Every a-regular* CCI *set has the property* Q.

Proof: Let $\mathscr{A}$ be a P-asynchronous automaton such that $P=P(\mathscr{A})$.

Let us establish a preliminary property. Let q, q', and q'' be the states reached by $\mathscr{A}$ after reading u, uv, and uw, with $uvw \sim uwv$. Let I' and I'' be the sets of components that $\mathscr{A}$ accesses when reading v and w from state q, and let J' and J'' be their complements. Thus, $I' \cap I'' = \varnothing$, and $q_{J'} = q'_{J'}$, $q_{J''} = q''_{J''}$. Therefore I' and I'' are also the sets of components accessed by $\mathscr{A}$ when reading v from state q'' and w from state q'.

Let u, v, w_1, and w_2 be such that $uvw_1 a \sim uw_1 av$ and $uvw_2 b \sim uw_2 bv$.

Let q and q' be the states reached by $\mathscr{A}$ after reading u and uv. Let us denote by I the set of indices of the components that $\mathscr{A}$ accesses when it reads v starting in state q, and by J its complement, so that $q_J = q'_J$. Since $uvw_1 a \sim uw_1 av$, a is not in I. For similar reasons, b is not in I.

Let s and s' be the states reached by $\mathscr{A}$ after reading uw and uvw. Let us assume that $uvw \sim uwv$. By the previous remark, I is also the set of components accessed by $\mathscr{A}$ reading v from state s and $s_J = s'_J$.

Since a and b are both in J, we have, in particular, $s_a = s'_a$ and $s_b = s'_b$. Thus, $uvab \sim uvba$ if and only if $D_a(s_a) \cap D_b(s_b) = \varnothing$ if and only if $D_a(s'_a) \cap D_b(s'_b) = \varnothing$ if and only if $uvwab \sim uvwba$. □

Now we can state the following conjecture.

CONJECTURE: *A* CCI *set is a-regular if and only if it is a regular set having the property Q.*

The reason why the property Q could allow us to prove that a regular set is also a-regular is the following. Since any regular set is r-projective, there exists a recognizable subset of some partially commutative monoid, closed under prefix, such that, roughly speaking, $P = \pi(L)$. Therefore, we can construct an asynchronous automaton $\mathscr{B}$ recognizing L, and we have to transform this automaton into a P-asynchronous automaton $\mathscr{A}$ recognizing $\pi(L)$. In order to do that, we have to guess, when $\mathscr{A}$ has read some word u and when a letter a has to be read, which is the letter a' such that $\sigma(ua) = \sigma(u) a'$. The problem is that this guess has to be made only from partial information about the word u, namely the prefix of u which has modified the component indexed by a of the state of the automaton $\mathscr{A}$. In some sense, the property Q amounts to saying that this a' is indeed not dependent on the whole word u but only on some of its prefixes.

As an example, let us consider the case where P satisfies the stronger property M, and let us consider the construction given in section 4 to prove that a CCI set which satisfies M is projective. We have to guess $\Gamma_u(a)$ for some word u. But it is easy to show that property M implies that

$\Gamma_u(a)=\Gamma_{\partial_a(u)}(a)$. Therefore, we can transform an asynchronous automaton recognizing $\sigma(A^*)$ into a P-asynchronous automaton by simply adding as component indexed by a, for any letter a, the congruence class, for $\equiv$, of $\partial_a(u)$. If b commutes with a in u, which can be decided knowing the congruence class of $\partial_a(u)$, the a-component of the state of the P-asynchronous automaton is not modified, since in this case $\partial_a(u)=\partial_a(ub)$. If b does not commute with a, it is possible to retrieve the new value of the congruence class of $\partial_a(ub)$ from the part of the state of the asynchronous automaton which has been modified in executing the transition associated with the b' corresponding to this b.

REFERENCES

1. J. J. Aalsberg, G. Rozenberg, Theory of traces. Technical Report 86-16, *Institute of Applied Mathematics and Computer Science*, University of Leiden, 1986.
2. R. Cori, Y. Métivier, Approximation of a trace, asynchronous automata, and the ordering of events in a distributed system. In Lepistö and Salomaa, editors, *15th I.C.A.L.P.*, pp. 147-161, *Lecture Notes Comput. Sci.*, 317, 1988.
3. R. Cori, D. Perrin, Automates et commutations partielles, *R.A.I.R.O. Inform. Théor. Appl.*, 1985, **19**, pp. 21-32.
4. P. Degano, U. Montanari, Specification languages for distributed systems. *In* H. Ehrig, C. Floyd, M. Nivat, J. Thatcher, Eds., *Mathematical Foundations of Software Development* (*TAPSOFT*'85), pp. 29-51, *Lecture Notes Comput. Sci.*, 185, 1985.
5. U. Goltz, W. Reisig, The non-sequential behaviour of Petri nets, *Inform. and Control*, 1983, **57**, pp. 125-147.
6. J. Grabowski, On partial languages, *Fundamenta Informaticae*, 1981, IV, 2, pp. 427-498.
7. A. Mazurkiewicz, Traces, histories, and graphs: Instances of process monoids, *In* M. Chytil, V. Koubek Eds., *MFCS*'84, pp. 115-133, *Lecture Notes Comput. Sci.*, 176, 1984.
8. A. Mazurkiewicz, Compositional semantics of pure place/transition systems, *Fund. Inform.*, XI, 1988.
9. A. Mazurkiewicz, Concurrency, modularity, and synchronization. *In* A. Kreczmar, G. Mirkowska Eds., *MFCS*'89, pp. 577-598, *Lecture Notes Comput. Sci.*, 379, 1989.
10. Y. Métivier, Une condition suffisante de reconnaissabilité dans un monoïde partiellement commutatif. *R.A.I.R.O. Inform. Théor. Appl.*, 1985, **19**, pp. 121-127.
11. M. Nielsen, G. Plotkin, G. Winskel, Petri nets, event structures and domains, *Theoret. Comput. Science*, 1981, **13**, pp. 85-108.
12. E. Ochmanski, Regular behaviour of concurrent systems, *Bulletin EATCS*, 1985, **27**, pp. 56-67.
13. A. Rabinovich, B. A. Trakhtenbrot, Behavior structures and nets, *Fund. Inform.*, **XI**, 1988, pp. 357-404.

14. B. Rozoy, *Un modèle de parallélisme : le monoïde distribué*, Thèse d'État, Université de Caen, 1987.
15. E. Szpilrajn, Sur l'extension de l'ordre partiel, *Fund. Math.*, 1930, **16**, pp. 386-389.
16. J. Winkowski, Behaviours of concurrent systems, *Theoret. Comput. Sci.*, 1980, **12**, pp. 39-60.
17. J. Winkowski, An algebraic description of system behaviours, *Theoret. Comput. Sci.*, 1982, **21**, pp. 315-340.
18. W. Zielonka, Notes on finite asynchronous automata, *R.A.I.R.O. Inform. Théor. Appl.*, 1987, **21**, pp. 99-135.