INFORMATION AND COMPUTATION 106, 159–202 (1993)

Asynchronous Mappings and Asynchronous Cellular Automata

ROBERT CORI*

Laboratoire Bordelais de Recherche en Informatique,[†] *Université Bordeaux-1, 351, cours de la Libération, 33405 Talence Cedex, France*

YVES MÉTIVIER*

Laboratoire Bordelais de Recherche en Informatique,[†] *ENSERB, 351, cours de la Libération, 33405 Talence Cedex, France*

AND

WIESŁAW ZIELONKA

Laboratoire Bordelais de Recherche en Informatique,[†] *Université Bordeaux-1, 351, cours de la Libération, 33405 Talence Cedex, France*

The aim of this paper is the study of asynchronous automata, a special kind of automata which encode the independency relation between actions and which enable their concurrent execution. These automata, introduced by Zielonka (*RAIRO Inform. Theor. Appl.* **21**, 99–135 (1987)), constitute a natural extension of finite automata to the case of asynchronous parallelism. Their behaviour is described by trace languages, subsets of partially commutative monoids. The main result concerning this class of automata states that they accept exactly all recognizable trace languages. In this paper we give new improved constructions of asynchronous automata. In the final part of the paper we present a distributed system of messages with bounded time-stamps based on asynchronous automata.

INTRODUCTION

Many different mathematical models have been introduced to express concurrency; let us mention Petri nets (Reisig, 1982), CSP (Brookes *et al.*, 1984), and CCS (Milner, 1983) as the most popular examples. Among these models we can distinguish machine-oriented models, where

* With the support of the P.R.C. Mathématiques et Informatique and the European project EBRA 3166—ASMICS.
[†] URA CNRS 1304.

0890-5401/93 $5.00

a concurrent system is represented by a device with a possibility of concurrent execution of actions. Finally, the machine-oriented models of concurrency can be divided into two classes, synchronous and asynchronous machines; in the first class we can find for instance systolic systems (Conway and Mead, 1980) and cellular automata (von Neumann, 1966), in the second Petri nets.

There are also many different semantics of concurrency. But in the simplest non-interleaving setting the behaviour of concurrent asynchronous systems is given by a partial order of actions, where two instances of actions are incomparable if they are causally independent. Partially ordered sets and operations over them are rather cumbersome to handle algebraically. This obstacle was circumvented to some extent in Mazurkiewicz's paper (1977). He proposed to describe the behaviour of concurrent systems with a fixed independency relation between actions by means of partially commutative monoids.

Although elements of these monoids, called traces, can be interpreted as labelled partial orders, this representation is almost never used in practice, because traces have also other interpretations much more convenient in use. Let us notice that the concept of partially commutative monoid was introduced earlier by Cartier and Foata (1969) for purely combinatorial purposes. Now the theory of partially commutative monoids is growing rapidly mainly in relationship to various aspects of concurrency: Aalbersberg and Rozenberg (1986), Bertoni *et al.* (1989), Choffrut (1986), Cori and Perrin (1985), Flé and Roucairol (1985), Gastin and Rozoy (1991), Mazurkiewicz (1984), Métivier (1986, 1987), Ochmański (1985), Perrin (1989), Sakarovitch (1987), Viennot (1987), Zielonka (1987), and Diekert (1990). In this paper we examine these monoids in connection with automata theory.

As it turns out, nets used in Mazurkiewicz's original paper (1977) to model concurrent systems suffer from a weakness; they are able to accept only a proper subclass of recognizable trace languages.

A new model, asynchronous finite automata, more suited for recognition of traces was introduced by Zielonka (1987). Although asynchronous automata relate closely to other machine-like models such as Petri nets or a model presented by Karp and Miller (1969), they fit much better the framework of classical automata theory. In fact, they may be viewed as finite automata with a decentralized control structure that allows concurrent execution of some actions. They also provide a remedy for the deficiency of Mazurkiewicz's model since the main result of Zielonka (1987) shows that they accept exactly all recognizable trace languages. Various aspects of asynchronous automata were next examined in a series of papers: Bruschi *et al.* (1988), Cori *et al.* (1988), Cori and Métivier (1988), Métivier (1987), Perrin (1989), and Zielonka (1989). As a recent interesting application of asynchronous automata, let us mention also a paper of Thomas (1990) where old results due to Büchi and Elgot concerning definability of words in monadic second order logic are extended to traces.

The construction of asynchronous automata presented in Zielonka (1987) is quite opaque and difficult to follow. The aim of this paper is to reconsider it in a more systematic way using an algebraic framework. Exploiting some new ideas, partly introduced by Cori and Métivier (1988), we are able to simplify considerably both the construction and its presentation, reducing also the number of states of the resulting automaton.

In fact, we even give two partly independent constructions. We have chosen to construct cellular asynchronous automata defined in Zielonka (1989) rather than the original asynchronous automata of Zielonka (1987). The reason is that the cellar asynchronous automata seem to constitute a more fundamental object, in fact they can be transformed immediately to asynchronous automata, while the inverse transformation, although possible, is not so trivial.

As an intermediate step of the construction of asynchronous cellular automata, we use so-called asynchronous mappings. The concept of asynchronous mapping turns out to be very useful. This fundamental notion captures in an algebraic and machine-independent way the idea of distributed recognizability of traces languages.

Our paper is organized as follows. In Section 1 we recall the basic notions concerning partially commutative monoids, recognizable sets, and we define the main object of the paper: asynchronous cellular automaton. In Sections 2 and 3 we study combinatorial properties of the prefix order for traces. In Section 4 we introduce the new fundamental notion of asynchronous mapping. We show that, given an asynchronous mapping, a corresponding asynchronous cellular automaton can easily be constructed. Thus in the remainder of the paper we concentrate our efforts on the construction of asynchronous mappings.

First nontrivial asynchronous mapping is built in Section 4. It has very nice properties and gives rise to a cellular asynchronous automaton which, intuitively, "recognizes" the prefix order of actions in traces. In Section 5 we achieve the construction of an asynchronous mapping recognizing a given trace language T; in fact we give two independent constructions, both of them based on the mapping built in Section 4.

In Section 6 we present an interesting application of the asynchronous mapping of Section 4. We consider there a distributed system with agents communicating by means of messages left in boxes and show how to construct a bounded time-stamp system in this case. Note that the problem of constructing bounded time-stamp systems is usually very difficult and solutions need ingenious ideas; see, for example, Li and Vitanyi (1989) for another such system.

1. Preliminaries

In this section we introduce the basic notions of traces, recognizable subsets of traces, and asynchronous cellular automata that will be used in this paper.

We use the standard mathematical notations. In particular, for any sets X, Y, by $\mathscr{P}(X)$ we denote the family of all subsets of X and by $F(X; Y)$ we denote the family of all partial mappings from X to Y. For the notions of formal language theory we follow in general Eilenberg (1974).

Let A be a finite alphabet, its elements are called letters or actions. Then A^* is the set of all words over A. Formally, A^* with the concatenation operation forms the free monoid with the set of generators A and where the empty word ε plays the role of the unit element. For any word x of A^*, $|x|$ denotes the length of x, $|x|_a$ denotes the number of occurrences of a letter a in x, and for any subset α of A, $|x|_\alpha$ denotes the length of the word obtained from x by deleting all the letters which are not in α ($|x|_\alpha = \sum_{a \in \alpha} |x|_a$). The notation $\text{alph}(x) = \{a \in A \mid |x|_a \neq 0\}$ is used to denote the set of letters of A actually appearing in the word x.

A. *Traces*

Let A be a finite alphabet. Throughout the paper Θ is a symmetric and irreflexive relation over the alphabet A, called the independency relation. Two actions a and b such that $(a, b) \in \Theta$ are considered independent. Intuitively, this means that a and b act on disjoint sets of resources and the order in which they are performed does not matter; they can even be performed simultaneously. The complement $\bar{\Theta} = A \times A \setminus \Theta$ of Θ is the dependency relation: two actions a and b such that $(a, b) \notin \Theta$ are dependent and cannot be executed simultaneously. For any letter a of A, $\bar{\Theta}(a)$ denotes the set of letters which depend on a:

$$\bar{\Theta}(a) = \{b \in A \mid (a, b) \notin \Theta\}.$$

Note that since Θ is irreflexive, $a \in \bar{\Theta}(a)$. We extend this notation by setting, for any $\alpha \subseteq A$,

$$\bar{\Theta}(\alpha) = \{b \in A \mid \exists a \in \alpha \text{ such that } (a, b) \in \bar{\Theta}\}.$$

The relation Θ induces an equivalence relation $\sim_\Theta$ over A^*: two words x and y are equivalent under $\sim_\Theta$, denoted by $x \sim_\Theta y$, if there exists a sequence $z_1, z_2, ..., z_k$ of words such that $x = z_1$, $y = z_k$, and for all i, $1 \leqslant i < k$, there exist words u_i, v_i, and letters a_i, b_i satisfying

$$z_i = u_i a_i b_i v_i, \qquad z_{i+1} = u_i b_i a_i v_i, \qquad \text{and} \qquad (a_i, b_i) \in \Theta.$$

Thus two words are equivalent by $\sim_\Theta$ if one can be obtained from the other by successive transpositions of neighbouring independent actions. It is easy to verify that $\sim_\Theta$ is the least congruence over A^* such that $ab \sim_\Theta ba$ for each pair $(a, b) \in \Theta$. The quotient of A^* by the congruence $\sim_\Theta$ is the free partially commutative monoid induced by the relation Θ, it is denoted by $M(A, \Theta)$. The elements of $M(A, \Theta)$, which are equivalence classes of words of A^* under the relation $\sim_\Theta$, are called traces. For a word x of A^*, $[x]_\Theta$ denotes the equivalence class of x under $\sim_\Theta$. Since the relation $\sim_\Theta$ is a congruence, the composition operation in $M(A, \Theta)$ is given by

$$\forall x, y \in A^* \qquad [x]_\Theta [y]_\Theta = [xy]_\Theta.$$

For the sake of simplicity, the equivalence class $[a]_\Theta$ of a letter a of A will be denoted by a in the sequel. Similarly, the unit element of $M(A, \Theta)$ (i.e., the equivalence class of the empty word, $[\varepsilon]_\Theta$) will be denoted by ε. It is obvious that any two elements x and y of A^* such that $[x]_\Theta = [y]_\Theta$ differ only by the order in which the letters appear; therefore it is possible to define for a trace $t = [x]_\Theta$ of $M(A, \Theta)$ the length $|t| = |x|$ of t, the number $|t|_a = |x|_a$ of occurrences of the letter a in t, the set $\text{alph}(t) = \text{alph}(x)$ of letters appearing in t, and if $\alpha \subseteq A$, then $|t|_\alpha = \sum_{a \in \alpha} |t|_a$.

Two traces u and v of $M(A, \Theta)$ are said to be independent and this fact is denoted by $u \Theta v$ if

$$\text{alph}(u) \times \text{alph}(v) \subseteq \Theta.$$

Note that this is equivalent to the following conditions:

$$uv = vu \qquad \text{and} \qquad \text{alph}(u) \cap \text{alph}(v) = \varnothing.$$

A trace p is a prefix of a trace t if $t = pr$ for some trace r; in such a case r is a suffix of t. The prefix relation is an order relation and it will be denoted by $\preccurlyeq$; $u \preccurlyeq v$ means that u is a prefix of v, and $u \prec v$ means that u is a proper prefix of v (i.e., u is a prefix of v different from v).

The following important result characterizing factorizations of traces was proved in Mazurkiewicz (1984) and in Cori and Perrin (1985). It will often be used in Section 2.

PROPOSITION 1.1. *Let t, u, v, w be traces of $M(A, \Theta)$ such that $tu = vw$. Then there exist unique traces t_1, t_2, t_3, t_4 such that*

$$t = t_1 t_2, \qquad u = t_3 t_4, \qquad v = t_1 t_3, \qquad w = t_2 t_4, \qquad \textit{and} \qquad t_2 \Theta t_3.$$

COROLLARY 1.2. *The monoid $M(A, \Theta)$ is cancellative*; *i.e.*,

$$\forall u, v, w \in M(A, \Theta) \qquad uv = uw \Rightarrow v = w,$$
$$vu = wu \Rightarrow v = w.$$

Traces admit numerous different representations. We present here two of them which, although not used explicitly in our paper, may provide the reader with useful intuitions.

A dependency graph over (A, Θ) is a triple (V, E, λ), where (V, E) is a finite acyclic graph and $\lambda: V \to A$ is a vertex labelling verifying the following condition: for all $v_1, v_2 \in V$,

$$v_1 \neq v_2 \quad \text{and} \quad (\lambda(v_1), \lambda(v_2)) \notin \Theta \Leftrightarrow (v_1, v_2) \in E \text{ or } (v_2, v_1) \in E.$$

Traces can be identified with (classes of isomorphisms of) dependence graphs. If (V, E, λ) is a dependence graph then the corresponding trace t of $M(A, \Theta)$ is given by

$$t = \{\lambda(x) \mid x \text{ is a list of the vertices of the graph } (V, E) \text{ sorted topologically}\}.$$

On the other hand, with each word $w \in A^*$ we can associate a dependence graph $D(w)$ defined by

$D(\varepsilon) = (\emptyset, \emptyset, \emptyset)$;
if $w \neq \varepsilon$ then $D(w) = (V, E, \lambda)$, where $V = \{1, ..., |w|\}$, $\forall i \in V$, $\lambda(i) = i$th letter of w, $E = \{(i, j) \in V^2 \mid i < j$ and $(\lambda(i), \lambda(j)) \notin \Theta\}$.

Then for any words $x, y \in A^*$, $x \sim_\Theta y$ if and only if $D(x)$ and $D(y)$ are isomorphic. Moreover, the set of dependence graphs can be equipped with a suitable composition operation.

Note that in this representation, if we have a dependence graph $D(t) = (V, E, \lambda)$ of a trace t of $M(A, \Theta)$ then a trace r is a prefix of t if and only if there exists a subset V' of V that is backward closed under E (i.e., such that whenever $(v_1, v_2) \in E$ and $v_2 \in V'$ then $v_1 \in V'$) and such that the subgraph $(V', E \cap (V' \times V'), \lambda_{|V'})$ of $D(t)$ induced by V' is (isomorphic with) a dependence graph of r.

The second representation of a trace t by a labelled partial order is obtained from the dependence graph $D(t) = (V, E, \lambda)$ of t by taking the transitive closure E^+ of E. Intuitively two distinct events $v_1, v_2 \in V$ are interpreted as causally independent if neither $(v_1, v_2) \in E^+$ nor $(v_2, v_1) \in E^+$, while $(v_1, v_2) \in E^+$ means that the event v_1 precedes the event v_2. Moreover, each event $v \in V$ is an occurrence of the action $\lambda(v)$.

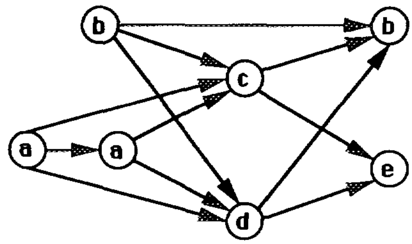

FIGURE 1.

Figure 1 gives the dependence graph $D(t)$ of the trace $t = [abadceb]_\Theta$, where $\Theta = \{(a, b), (b, a), (d, c), (c, d), (e, a), (a, e), (b, e), (e, b)\}$. We see that, for instance, the second occurrence of a is independent of the first occurrence of b but it precedes the second occurrence of b.

B. *Recognizable Trace Languages*

By analogy with the theory of formal languages the subsets of $M(A, \Theta)$ are called trace languages. We recall the following basic facts from Eilenberg (1974).

Let M be a monoid with the unit element ε, a subset T of M is said to be recognizable if there exists a homomorphism from M into a finite monoid H such that $T = f^{-1}(G)$ for some subset G of H.

For a monoid M, an M-automaton $\mathscr{A} = (M, Q, \delta, q_0, F)$ consists of a finite set Q of states, an initial state $q_0 \in Q$, a subset F of Q of final states and a transition function δ from $Q \times M$ into Q satisfying the following conditions:

$$\forall q \in Q \qquad \delta(q, \varepsilon) = q,$$

$$\forall q \in Q, \forall m_1, m_2 \in M \qquad \delta(q, m_1 m_2) = \delta(\delta(q, m_1), m_2).$$

The subset T of M recognized by the automaton $\mathscr{A}$ is defined by

$$T = \{m \in M \mid \delta(q_0, m) \in F\}.$$

The following result is classical (Eilenberg, 1974) and easy to prove.

PROPOSITION 1.3. *A subset T of M is recognizable if and only if there exists an M-automaton which recognizes it.*

From now on we assume that the monoid M is the trace monoid $M(A, \Theta)$. For any subset L of A^*, $[L]_\Theta$ is the subset of $M(A, \Theta)$ consisting of the traces generated by elements of L:

$$[L]_\Theta = \{t \in M(A, \Theta) \mid \exists x \in L \text{ such that } [x]_\Theta = t\}.$$

Conversely, for any subset T of $M(A, \Theta)$ the set $[T]_\Theta^{-1} = \{x \in A^* \mid [x]_\Theta \in T\}$ consists of the words which represent traces from T. The following easy proposition characterizes recognizable subsets of $M(A, \Theta)$:

PROPOSITION 1.4. *A subset T of $M(A, \Theta)$ is recognizable if and only if $[T]_\Theta^{-1}$ is a recognizable subset of A^*.*

Let $\mathscr{A} = (A, Q, \delta, q_0, F)$ be an A^*-automaton and let Θ be an independency relation. $\mathscr{A}$ is said to be Θ-compatible if for any words x and y,

$$x \sim_\Theta y \Rightarrow \forall q \in Q \;\; \delta(q, x) = \delta(q, y).$$

Note that it suffices to verify the preceding property for words of length 2; namely, $\mathscr{A}$ is Θ-compatible if and only if for any $a, b \in A$,

$$(a, b) \in \Theta \Rightarrow \forall q \in Q \;\; \delta(q, ab) = \delta(q, ba).$$

A Θ-compatible automaton $\mathscr{A}$ may be considered as an $M(A, \Theta)$-automaton, since for any two equivalent words x and y, and for any state q we have $\delta(q, x) = \delta(q, y)$. This makes it possible to extend δ for traces by setting $\delta(q, t) = \delta(q, x)$, for all $t \in M(A, \Theta)$ and $x \in A^*$ such that $[x]_\Theta = t$. As it is easy to verify we then have $\delta(q, st) = \delta(\delta(q, s)\, t)$ for all $q \in Q$ and $s, t \in M(A, \Theta)$.

C. *Asynchronous Cellular Automata*

Asynchronous cellular automata, which we introduce in this section, may be viewed as ordinary finite automata but with an internal structure of states resembling the structure of the cellular automata (von Neumann, 1966). The property making them different from the usual cellular automata is that they have decentralized control structure and they perform actions asynchronously. We begin with an informal introduction. Let A be a finite alphabet and Θ an independency relation. Let $\mathscr{G}$ be the dependency graph associated with the corresponding dependency relation $\bar{\Theta}$ whose vertices are letters of A and whose edges are the pairs of distinct non-commuting letters. With every vertex $a \in A$ of $\mathscr{G}$ there are associated an agent, a transition mapping δ_a, and a value, called a state, from a finite set X. If the agent a makes an action then it examines the states of all its neighbours from $\bar{\Theta}(a)$ and changes its own state (i.e., value) in accordance with its transition mapping δ_a. This single action is considered as atomic, which implies that two neighbouring agents cannot act simultaneously. Recall that $F(X; Y)$ denotes the set of all mappings from X into Y.

DEFINITION. A deterministic asynchronous cellular automaton A over $M(A, \Theta)$ is given by a finite set X of basic states, a family $\{\delta_a \mid a \in A\}$ of transitions mappings, where for every $a \in A$, δ_a is a partial mapping from $F(\bar{\Theta}(a); X)$ into X, an initial state $s_0 \in F(A; X)$, a set of final states $F \subseteq F(A; X)$.

Let S denote the set $F(A; X)$. The elements of S are called global states of A. For any $s \in S$ and for any $\alpha \subseteq A$, by $s|_\alpha$ we denote the restriction of s to the set α (i.e., $s|_\alpha \in F(\alpha; X)$ is such that $\forall a \in \alpha$, $s|_\alpha(a) = s(a)$).

To describe the sequential behaviour of the asynchronous cellular automaton, we introduce the global transition mapping Δ from $S \times A$ into S.

DEFINITION. Let $s, s' \in S$ and $a \in A$, then $\Delta(s, a) = s'$ if $s'(a) = \delta_a(s|_{\bar{\Theta}(a)})$ and $\forall b \in A \setminus \{a\}$, $s'(b) = s(b)$.

The mapping Δ is extended to $S \times A^*$ in the standard way. Note that $\mathscr{A} = (A, S, \Delta, s_0, F)$ is an ordinary A^*-automaton, and

$$\forall (a, b) \in \Theta, \forall s \in S \qquad \Delta(s, ab) = \Delta(s, ba).$$

Thus $\mathscr{A}$ is Θ-compatible and we can extend the global transition mapping to traces by setting $\Delta(s, t) = \Delta(s, x)$, where $x \in A^*$ is such that $[x]_\Theta = t$, and the trace language recognized by the cellular asynchronous automaton A is

$$T(A) = \{t \in M(A, \Theta) \mid \Delta(s_0, t) \in F\}.$$

It is clear directly from the definition that the trace language $T(A)$ recognized by an asynchronous cellular automaton A is a recognizable subset of $M(A, \Theta)$. The aim of this paper is to prove that also the inverse holds:

THEOREM. *For every recognizable subset T of $M(A, \Theta)$ there exists a deterministic asynchronous cellular automaton A over $M(A, \Theta)$ recognizing T.*

The theorem presented above seems to be very natural but it is by no means trivial. The reader may try to construct an asynchronous cellular automaton recognizing the following trace language:

$$T = [((a+c)(b+d))^*]_\Theta, \qquad \text{where} \quad \Theta = \{(a, c), (c, a), (b, d), (d, b)\}.$$

The usual minimal $M(A, \Theta)$-automaton recognizing T has only two states but the least known asynchronous cellular automaton recognizing T has more than a hundred of global states.

Asynchronous cellular auomata relate closely to the usual cellular automata. The set A of vertices of the graph $\mathscr{G}$ corresponds to the set of cells. In the cellular automata the connection pattern is usually regular,

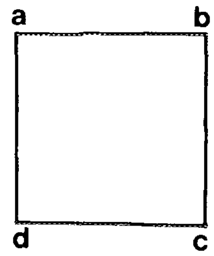

FIGURE 2.

(e.g., this can be a grid and they can work in the synchronized mode), while in the case of the asynchronous cellular automata the connection pattern is given by the graph $\mathscr{G}$ of the dependency relation $\bar{\Theta}$ and actions are executed asynchronously.

EXAMPLE. Let $A = \{a, b, c, d\}$, $\Theta = \{(a, c), (c, a), (b, d), (d, b)\}$. Figure 2 shows the corresponding dependency graph $\mathscr{G}$.

We present an asynchronous automaton $\mathcal{A}$ recognizing the trace language $T = [(acbd)^* (b \cup d)]_\Theta$. As the set of basic states we take $X = \{0, 1, 2\}$. Now note that all transition mappings δ_u, $u \in A$, have different domains. For convenience we introduce notation making it possible to present them in a uniform way. Let $\sharp$ be a special symbol not in X, and let $Y = X \cup \{\sharp\}$. Let $u \in A$, $r \in F(\bar{\Theta}(u); X)$ and $x \in X$. Then the fact that $\delta_u(r) = x$ will be denoted as $s \xrightarrow{u} s'$, where $s, s' \in F(A; Y)$ are defined in the following way: for all $v \in A$,

$$s(v) = \begin{cases} \sharp & \text{if } v \notin \bar{\Theta}(u), \\ r(v) & \text{if } v \in \bar{\Theta}(u), \end{cases}$$

$$s'(v) = \begin{cases} \sharp & \text{if } v \neq u, \\ x & \text{if } v = u. \end{cases}$$

Moreover, each element $s \in F(A; Y)$ will be identified with the quadruple

$$(s(a), s(b), s(c), s(d))$$

of values of Y (in particular each global state $s \in F(A; X)$ will be represented by a quadruple with all components in X).

For example, if $r \in F(\{a, b, d\}; X)$ is such that $r(a) = r(b) = r(d) = 0$ and $\delta_a(r) = 1$ then this fact is represented by

$$(0, 0, \sharp, 0) \xrightarrow{a} (1, \sharp, \sharp, \sharp).$$

According to the intuitive ideas that were introduced earlier this rule means that if the basic states of all the agents a, b, d are equal to 0 then a can change its own basic state to 1. The sharp symbol at the third component of the first quadruple of the rule means that the basic state of c is irrelevant (a has no access to this state upon reading). The sharp symbols in the second quadruple of the rule mean that a has no access to the local states of b, c, and d upon writing. In this notation, transition mappings of $\mathcal{A}$ are given by the following set of rules:

$$
\begin{array}{ll}
(0, 0, \sharp, 0) \xrightarrow{a} (1, \sharp, \sharp, \sharp) & (\sharp, 0, 0, 0) \xrightarrow{c} (\sharp, \sharp, 1, \sharp), \\
(1, 1, \sharp, 1) \xrightarrow{a} (0, \sharp, \sharp, \sharp), & (\sharp, 1, 1, 1) \xrightarrow{c} (\sharp, \sharp, 0, \sharp), \\
(0, 0, 0, \sharp) \xrightarrow{b} (\sharp, 2, \sharp, \sharp), & (0, \sharp, 0, 0) \xrightarrow{d} (\sharp, \sharp, \sharp, 2), \\
(1, 0, 1, \sharp) \xrightarrow{b} (\sharp, 1, \sharp, \sharp), & (1, \sharp, 1, 0) \xrightarrow{d} (\sharp, \sharp, \sharp, 1), \\
(0, 1, 0, \sharp) \xrightarrow{b} (\sharp, 0, \sharp, \sharp), & (0, \sharp, 0, 1) \xrightarrow{d} (\sharp, \sharp, \sharp, 0), \\
(1, 1, 1, \sharp) \xrightarrow{b} (\sharp, 2, \sharp, \sharp), & (1, \sharp, 1, 1) \xrightarrow{d} (\sharp, \sharp, \sharp, 2).
\end{array}
$$

The initial state of $\mathcal{A}$ is equal $s_0 = (0, 0, 0, 0)$ and we have four final states $F = \{(0, 2, 0, 0), (0, 0, 0, 2), (1, 2, 1, 1), (1, 1, 1, 2)\}$.

Figure 3 gives the transition diagram of the underlying finite automaton (only the states accessible from the initial state are presented). Let us note the presence of two states $(0, 2, 0, 2)$ and $(1, 2, 1, 2)$ which are accessible but not co-accessible.

2. Prefix Order for Traces

Let us recall that a trace u is a prefix of a trace v if $v = uu'$ for some trace u', and this is denoted by $u \preccurlyeq v$. Obviously, $\preccurlyeq$ defines a partial order among traces. Properties of the prefix order are more complicated for traces than for words. The main reasons are that two prefixes of a given word are always comparable by the prefix order while this is not always true for traces and that a given trace has in general many more prefixes than a word of the same length. For instance let $t = [a^m b^k]_\Theta$, where a and b are independent. Let $n < m + k$ be an integer; then for every trace of the form $[a^{n_1} b^{n_2}]_\Theta$, where $n_1 \leqslant m$, $n_2 \leqslant k$, $n_1 + n_2 = n$ is a prefix of t, all these prefixes have the same length and are pairwise incomparable by the prefix order.

The aim of this section is to present basic properties of the prefix relation for traces. Although the results given here are not new, they are presented with proofs for the sake of completeness. We introduce and examine several important notions: α-prefixes, prime traces, maximal letters of a trace.

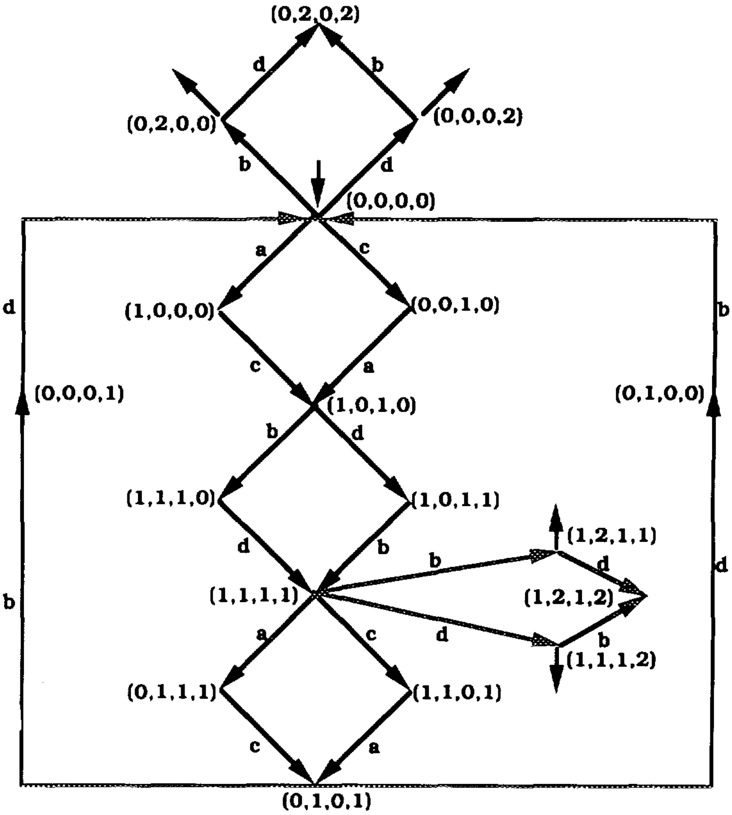

FIGURE 3.

Roughly speaking, for $\alpha \subseteq A$, the α-prefix of a trace t, denoted $\partial_\alpha(t)$, is the shortest prefix of t containing all letters from α occurring, i.e., such that $|\partial_\alpha(t)|_\alpha = |t|_\alpha$. A letter a of A is maximal in t if $t = ra$ for some trace r, and finally t is prime if either t is empty or it has exactly one maximal letter. Prime elements are of great importance since each trace is the least upper bound w.r.t. $\preccurlyeq$ of prime elements. For a detailed study of the poset $(M(A, \Theta), \preccurlyeq)$ we refer to Gastin and Rozoy (1991).

A. *Basic Results*

As in the case of words the least common upper bound $u \vee v$ w.r.t. $\preccurlyeq$ of two traces u and v of $M(A, \Theta)$ may not exist. The following proposition gives some necessary and sufficient conditions ensuring the existence of $u \vee v$.

PROPOSITION 2.1. *For any pair u, v of traces, the following statements are equivalent*:

(1) *there exists a trace t such that u and v are prefixes of t,*

(2) *there exist unique traces t_0, u', and v' such that $u = t_0 u'$, $v = t_0 v'$, and $u' \Theta v'$,*

(3) *there exists the least common upper bound $u \vee v$ of u and v w.r.t. the prefix order $\preccurlyeq$.*

Moreover, if (2) is satisfied then $u \vee v = t_0 u' v'$ and t_0 is the greatest common lower bound $u \wedge v$ of u and v w.r.t. $\preccurlyeq$.

Proof. (1) $\Rightarrow$ (2). Let $t = ut_1 = vt_2$. Then by Proposition 1.1 there exist unique traces t_0, u', v', and r such that

$$u = t_0 u', \qquad v = t_0 v', \qquad t_1 = v'r, \qquad t_2 = u'r, \qquad \text{and} \qquad u' \Theta v'.$$

Hence (2) holds.

(2) $\Rightarrow$ (3). We are going to prove that $w = t_0 u' v'$ is the least common upper bound of u and v. Clearly $w = uv' = vu'$ and w is an upper bound of u and v. To prove that it is the least upper bound suppose that w' is a trace such that $w' = uv'' = vu''$ for some u'', v'' in $M(A, \Theta)$. By the cancellative property of $M(A, \Theta)$ we get $u'v'' = v'u''$. Again by Proposition 1.1 there exist unique traces u'_1, v'_1, w'_1, and w'_2 such that

$$u' = u'_1 w'_1, \qquad v'' = v'_1 w'_2, \qquad v' = u'_1 v'_1, \qquad u'' = w'_1 w'_2, \qquad \text{and} \qquad w'_1 \Theta v'_1.$$

Since $u' \Theta v'$ we get $u'_1 = \varepsilon$ and $v'' = v' w'_2$ giving $w' = uv'w'_2 = ww'_2$. Hence w is a prefix of w' and we are done.

(3) $\Rightarrow$ (1). Of course u nd v are prefixes of $u \vee v$.

It remains to prove that t_0 is the greatest common lower bound of u and v whenever $u = t_0 u'$, $v = t_0 v'$, and $u' \Theta v'$. Obviously t_0 is a lower bound for u and v. Assume that r is also a lower bound of u and v, then we get $u = t_0 u' = rr'$ and $v = t_0 v' = rr''$, for some traces r', r''. Hence $t_0 u' v' = rr'v' = rr''u'$ and by the cancellative property $r'v' = r''u'$. Now Proposition 1.1 applied in the particular case $u' \Theta v'$ gives $r' = wu'$, $r'' = wv'$, for some trace w. Hence $u = rr' = rwu' = t_0 u'$, and $t_0 = rw$ entailing $r \preccurlyeq t_0$. And this ends the proof. ∎

COROLLARY 2.2. *For any set U of traces there always exists the greatest lower bound of U w.r.t. $\preccurlyeq$.*

Proof. Let Pref(U) be the set of all common prefixes for traces from U. This set is non-empty since it contains at least the empty trace ε and it is finite since for each trace u the set $\mathrm{Pref}(u)$ of prefixes of u is finite and $\mathrm{Pref}(U)=\bigcap_{u\in U}\mathrm{Pref}(u)$. Now note that, since for each trace u of U elements of $\mathrm{Pref}(u)$ are prefixes of u, Proposition 2.1 ensures that the least upper bound of the set $\mathrm{Pref}(U)$ exists and it is obviously the greatest lower bound of U. ∎

B. α-Prefixes of Traces

In this subsection we define α-prefixes of traces, where α is a subset of A. They play a crucial role, in fact they constitute the main tool in formulations and proofs of basic properties of the prefix order.

For any trace t and for any letter a we denote by $\mathrm{Pref}_a(t)$ the set of prefixes of t having as many occurrences of the letter a as t has:

$$\mathrm{Pref}_a(t)=\{u\in M(A,\Theta)\mid u\preccurlyeq t \text{ and } |u|_a=|t|_a\}.$$

LEMMA 2.3. *The greatest common lower bound of the elements of* $\mathrm{Pref}_a(t)$ *is itself an element of* $\mathrm{Pref}_a(t)$.

Proof. First note that $\mathrm{Pref}_a(t)$ is not empty since it contains the trace t itself. Then by Corollary 2.2 it has the greatest common lower bound. We prove the following assertion which implies the thesis directly:

$$t_1, t_2\preccurlyeq t \qquad \text{and} \qquad |t|_a=|t_1|_a=|t_2|_a\Rightarrow |t_1\wedge t_2|_a=|t|_a. \tag{1}$$

Indeed by Proposition 2.1 we get $t_1=t_0t'_1$, $t_2=t_0t'_2$, and $t'_1\Theta t'_2$. Note that

$$|t_0|_a+|t'_1|_a=|t_0t'_1|_a=|t_1|_a=|t_2|_a=|t_0t'_2|_a=|t_0|_a+|t'_2|_a.$$

Thus $|t'_1|_a=|t'_2|_a$, but $t'_1\Theta t'_2$ implies that the alphabets of t'_1 and t'_2 are disjoint and hence $|t'_1|_a=|t'_2|_a=0$. This yields $|t_1\wedge t_2|_a=|t_0|_a=|t_1|_a=|t_2|_a$ which proves the assertion (1). ∎

The previous lemma justifies the following definition.

DEFINITION. Let t be a trace and $a\in A$; then $\partial_a(t)$ denotes the least prefix of t such that $|t|_a=|\partial_a(t)|_a$. For any nonempty subset α of A we set $\partial_\alpha(t)=\bigvee_{a\in\alpha}\partial_a(t)$. If $\alpha=\varnothing$ then $\partial_\alpha(t)=\varepsilon$. The trace $\partial_\alpha(t)$ is called the α-prefix of t.

Note that $\partial_\alpha(t)$ is always defined since the traces $\partial_a(t)$ for $a\in\alpha$ are prefixes of the trace t. Moreover, it is easy to verify that $\partial_\alpha(t)$ is the least prefix of t such that $|\partial_\alpha(t)|_\alpha=|t|_\alpha$.

EXAMPLE. Let $A=\{a,b,c,d\}$, $\Theta=\{(a,c),(c,a),(b,d),(d,b)\}$, and $t=[acabdb]_\Theta$. Then $\partial_a(t)=[aa]_\Theta$, $\partial_b(t)=[acabb]_\Theta$, $\partial_c(t)=c$, $\partial_d(t)=[acad]_\Theta$, $\partial_{\{a,c\}}(t)=[aca]_\Theta$.

We begin with a proposition which summarizes some obvious but useful properties of $\partial_\alpha(t)$.

PROPOSITION 2.4. *Let α and β be nonempty subsets of A and let t and r be traces of $M(A, \Theta)$. Then*:

(1) $\partial_\alpha(\partial_\alpha(t)) = \partial_\alpha(t)$,

(2) $\alpha \subseteq \beta \Rightarrow \partial_\alpha(t) = \partial_\alpha(\partial_\beta(t)) = \partial_\beta(\partial_\alpha(t))$ *and* $\partial_\alpha(t) \leqslant \partial_\beta(t)$,

(3) $\partial_{\alpha\cup\beta}(t) = \partial_\alpha(t) \vee \partial_\beta(t)$,

(4) $t = \partial_\alpha(t)\, r \Rightarrow \alpha \cap \mathrm{alph}(r) = \varnothing$.

(5) $r \leqslant t \Rightarrow \partial_\alpha(r) \leqslant \partial_\alpha(t)$,

(6) $(r \leqslant t$ *and* $|t|_\alpha = |r|_\alpha) \Rightarrow \partial_\alpha(t) = \partial_\alpha(r) \leqslant r$,

(7) $|\partial_\alpha(t)|_\alpha = |t|_\alpha$.

The next proposition gives a formula allowing us to compute $\partial_\alpha(uv)$ for any traces u and v.

PROPOSITION 2.5. *For any traces u and v and for any subset α of A we have*

(1) $\alpha \cap \mathrm{alph}(v) = \varnothing \Leftrightarrow \partial_\alpha(uv) = \partial_\alpha(u)$,

(2) $\partial_\alpha(uv) = \partial_{\alpha\cup\beta}(u)\, \partial_\alpha(v)$ *where* $\beta = \bar{\Theta}(\mathrm{alph}(\partial_\alpha(v)))$.

Proof. (1) Let $\alpha \cap \mathrm{alph}(v) = \varnothing$; then we get

$$|\partial_\alpha(uv)|_\alpha = |uv|_\alpha = |u|_\alpha = |\partial_\alpha(u)|_\alpha.$$

Since $\partial_\alpha(u) \leqslant uv$, Proposition 2.4 (6) yields $\partial_\alpha(uv) = \partial_\alpha(u)$. Conversely, $\partial_\alpha(uv) = \partial_\alpha(u)$ implies $|uv|_\alpha = |u|_\alpha$ and $\alpha \cap \mathrm{alph}(v) = \varnothing$.

(2) Let $\beta = \bar{\Theta}(\mathrm{alph}(\partial_\alpha(v)))$ and $t = \partial_{\alpha\cup\beta}(u)$, $r = \partial_\alpha(v)$. Then $u = tt'$ for some trace t' such that $|t'|_{\alpha\cup\beta} = 0$. Hence

$$r\Theta t', \qquad ur = tt'r = trt', \qquad \text{and} \qquad tr \leqslant ur \leqslant uv.$$

From $\partial_\alpha(u) \leqslant \partial_{\alpha\cup\beta}(u) \leqslant u$, we get $|u|_\alpha = |\partial_\alpha(u)|_\alpha = |\partial_{\alpha\cup\beta}(u)|_\alpha$ and hence

$$(3) \qquad |uv|_\alpha = |u|_\alpha + |v|_\alpha = |\partial_{\alpha\cup\beta}(u)|_\alpha + |\partial_\alpha(v)|_\alpha = |tr|_\alpha.$$

As tr is a prefix of uv, by (3) and Proposition 2.4 (6) $\partial_\alpha(uv)$ is a prefix of tr.

Now let w be a trace such that

$$(4) \qquad \partial_\alpha(uv)\, w = tr.$$

Since $|\partial_\alpha(uv)|_\alpha = |uv|_\alpha$ and $\partial_\alpha(uv)\,w$ is a prefix of uv we have $|w|_\alpha = 0$. Applying Proposition 1.1 to (4) we get

$$\partial_\alpha(uv) = t_1 t_2, \qquad w = t_3 t_4, \qquad t = t_1 t_3, \qquad r = t_2 t_4, \qquad \text{and} \qquad t_2 \Theta t_3.$$

Clearly, by $|w|_\alpha = 0$, we get $|t_4|_\alpha = 0$. Thus $|r|_\alpha = |t_2|_\alpha$ and since r is of the form $\partial_\alpha(v)$ we have $t_4 = \varepsilon$ and $r = t_2$. Hence we get $r\Theta w$ and w is a suffix of $t = \partial_{\alpha \cup \beta}(u)$. Since $\beta = \bar{\Theta}(\text{alph}(r))$ and $r\Theta w$ we get $|w|_\beta = 0$ which together with $|w|_\alpha = 0$ implies $|w|_{\alpha \cup \beta} = 0$. But a suffix of $\partial_{\alpha \cup \beta}(u)$ with this property is empty; thus $w = \varepsilon$ and (4) gives the required formula. ▮

PROPOSITION 2.6. *For any traces u and v such that the upper bound $u \vee v$ exists and for any subset α of A, $\partial_\alpha(u \vee v) = \partial_\alpha(u) \vee \partial_\alpha(v)$. Moreover, for any trace u and any letter a, the set $\{\partial_a(t) \mid t \preccurlyeq u\}$ is totally ordered by the prefix relation.*

Proof. If $u \vee v$ exists then by Proposition 2.1

$$u = t_0 t_1, \qquad v = t_0 t_2, \qquad \text{and} \qquad t_1 \Theta t_2.$$

Thus, for any $a \in A$ either $|t_1|_a = 0$ or $|t_2|_a = 0$. In the first case $\partial_a(u) = \partial_a(t_0) \preccurlyeq \partial_a(v)$ and $\partial_a(u \vee v) = \partial_a(v)$, in the second case $\partial_a(v) \preccurlyeq \partial_a(u)$ and $\partial_a(u \vee v) = \partial_a(u)$. This proves the second part of Proposition 2.6, and the first part in the particular case when α contains only one letter.

Now let us suppose that $\alpha = \{a_1, a_2, ..., a_k\}$, then $\partial_\alpha(u \vee v) = \bigvee_{i=1,k} \partial_{a_i}(u \vee v)$ but by the previous result we have $\partial_{a_i}(u \vee v) = \partial_{a_i}(u) \vee \partial_{a_i}(v)$ and the result follows from the commutativity of $\vee$. ▮

Remark. The trace $\partial_\alpha(u \wedge v)$ is not necessarily equal to $\partial_\alpha(u) \wedge \partial_\alpha(v)$ as the following example shows:

$$A = \{a, b, c, d\}, \qquad \Theta = \varnothing, \qquad u = abca, \qquad v = abda, \qquad u \wedge v = ab,$$

$$\partial_a(u \wedge v) = a, \qquad \partial_a(u) = u, \qquad \partial_a(v) = v, \qquad \text{and} \qquad \partial_a(u) \wedge \partial_a(v) = ab.$$

COROLLARY 2.7. *For any trace t, any letter a, and any subset α of A we have*

$$\partial_a(\partial_\alpha(t)) = \text{Max}\{\partial_a(\partial_b(t)) \mid b \in \alpha\},$$

where this maximum is taken for the prefix order $\preccurlyeq$.

Moreover, for any letter c such that $\partial_c(t) \preccurlyeq \partial_\alpha(t)$ there exists a letter b of α such that $\partial_c(t) = \partial_c(\partial_b(t))$.

Proof. We have $\partial_\alpha(t) = \bigvee_{b \in \alpha} \partial_b(t)$; thus, by Proposition 2.6 $\partial_a(\partial_\alpha(t)) = \bigvee_{b \in \alpha} \partial_a(\partial_b(t))$ and as the traces $\partial_b(t)$, $b \in \alpha$, are prefixes of t, also by Proposition 2.6 the set $\{\partial_a(\partial_b(t)) \mid b \in \alpha\}$ is totally ordered by the prefix order and its least upper bound is its maximal element. Now let $\partial_c(t) \leqslant \partial_\alpha(t) \leqslant t$; applying Proposition 2.4 (5) we get $\partial_c(t) \leqslant \partial_c(\partial_\alpha(t)) \leqslant \partial_c(t)$, hence $\partial_c(t) = \partial_c(\partial_\alpha(t))$. Using the first part of this corollary we finish the proof. ∎

C. *Prime Elements of $M(A, \Theta)$ and Maximal Letters of a Trace*

Now we define the subset $Pr(A, \Theta)$ of prime elements of $M(A, \Theta)$.

DEFINITION. $Pr(A, \Theta) = \{\partial_a(t) \mid t \in M(A, \Theta) \text{ and } a \in A\}$.

Note that $r \in Pr(A, \Theta)$ if and only if there exists a letter a such that $r = \partial_a(r)$ and if $r \neq \varepsilon$ then this letter a is unique. Since for any trace t we have $t = \partial_A(t) = \bigvee_{a \in A} \partial_a(t)$, the set $Pr(A, \Theta)$ generates $M(A, \Theta)$ by means of the operation $\vee$. Moreover, this is the least set with this property since it can be proved (Gastin and Rozoy, 1991) that for any $r \in Pr(A, \Theta)$ if $r = t_1 \vee t_2 \vee \cdots \vee t_k$ then for at least one i, $1 \leqslant i \leqslant k$, $r = t_i$.

For every trace t we distinguish the set $\mathrm{Max}(t)$ consisting of the letters a for which there exists the factorization $t = t'a$,

$$\mathrm{Max}(t) = \{a \in A \mid \exists t' \in M(A, \Theta) \text{ such that } t = t'a\}.$$

The following proposition gives a list of basic properties of $\mathrm{Max}(t)$.

PROPOSITION 2.8. *Let t be a trace and α a nonempty set of letters; then*

(1) $\partial_\alpha(t) = t \Leftrightarrow \mathrm{Max}(t) \subseteq \alpha$,

(2) *if* $\beta = \mathrm{Max}(t)$ *then* $t = \partial_\beta(t)$,

(3) *for every letter b there exists $a \in \mathrm{Max}(t)$ such that* $\partial_b(t) \leqslant \partial_a(t)$,

(4) $t \in Pr(A, \Theta) \setminus \{\varepsilon\} \Leftrightarrow \mathrm{Card}(\mathrm{Max}(t)) = 1$.

Proof. (1) '$\Leftarrow$' suppose that $t = \partial_\alpha(t)\, w$ for some trace $w \neq \varepsilon$. Then $w = w'a$ for some letter a and thus $a \in \mathrm{Max}(t)$. But $\alpha \cap \mathrm{alph}(w) = \varnothing$, i.e., $a \notin \alpha$, and we get a contradiction.

'$\Rightarrow$' If there exists a letter a such that $a \in \mathrm{Max}(t)$ but $a \notin \alpha$ then $t = t'a$ and $\partial_\alpha(t) = \partial_\alpha(t') \leqslant t' \prec t$.

(2) This is an immediate consequence of (1).

(3) Let $\beta = \mathrm{Max}(t)$. Then by Corollary 2.7 $\partial_b(t) = \partial_b(\partial_\beta(t)) = \partial_b(\partial_a(t))$ for some $a \in \beta$. But $\partial_b(\partial_a(t)) \leqslant \partial_a(t)$, which accomplishes the proof.

(4) Trivial. ∎

EXAMPLE. Let $A = \{a, b, c\}$, $\Theta = \{(a, c), (c, a)\}$, then $\mathrm{Max}([abcac]_\Theta) = \{a, c\}$, $\mathrm{Max}([acbc]_\Theta) = \{c\}$, $\mathrm{Max}([acab]_\Theta) = \{b\}$, and the last two traces are prime.

Note that al the definitions given here admit simple intuitive interpretations in terms of dependency graphs introduced in Section 1. Let t be a trace and $D(t)=(V, E, \lambda)$. Then for a subset α of A we set $V_\alpha=\{v\in V \mid \exists v'\in V,\ \lambda(v')\in\alpha \text{ and } (v, v')\in E^*\}$, where E^* is the transitive and reflexive closure of E. Then $D(\partial_\alpha(t))=(V_\alpha, (V_\alpha\times V_\alpha)\cap E, \lambda|_{V_\alpha})$. Similarly, $\text{Max}(t)$ corresponds to the set of labels of maximal elements of $D(t)$, $\text{Max}(t)=\{\lambda(v) \mid v\in V \text{ and } \neg(\exists v'\in V,\ (v, v')\in E)\}$.

3. Computing with ∂_α

In this section we continue to examine properties of the operator ∂_α. In particular, we investigate closely how the prefix $\partial_{\alpha\cup\beta}(t)$ of t is constructed from $\partial_\alpha(t)$ and $\partial_\beta(t)$. We begin with an auxiliary lemma.

LEMMA 3.1. *Let t be a trace, t_0t_1 and t_0t_2 be prefixes of t such that*

$$\exists\alpha\subseteq A, \qquad \partial_\alpha(t)=t_0t_1 \textit{ and } t_1\Theta t_2.$$

Then $\text{Max}(t_0)\cap\text{alph}(t_2)=\varnothing$.

Proof. Assume that a is an element of $\text{alph}(t_2)$ and of $\text{Max}(t_0)$. Then we get

$$|t_0|_a<|t_0t_2|_a\leqslant|t|_a. \tag{1}$$

Since $t_1\Theta t_2$ and $a\in\text{alph}(t_2)$, we have $at_1=t_1a$. On the other hand, $a\in\text{Max}(t_0)$ implies that $t_0=t_0'a$ for some trace t_0'. Thus

$$\partial_\alpha(t)=t_0t_1=t_0'at_1=t_0't_1a.$$

The last equality implies $a\in\alpha$. Hence

$$|t|_a=|\partial_\alpha(t)|_a=|t_0t_1|_a=|t_0|_a. \tag{2}$$

But (1) and (2) are in contradiction, proving the lemma. ∎

DEFINITION. For any two traces u and v, we define the following sets:

$$E(u, v)=\{a\in A \mid \partial_a(u)=\partial_a(v)\},$$

$$NE(u, v)=\{a\in A \mid \partial_a(u)\neq\partial_a(v)\},$$

$$G(u, v)=\{a\in A \mid \partial_a(u)\prec\partial_a(v)\}.$$

Note that $NE(u, v)=A\setminus E(u, v)$, and if u and v are prefixes of the same trace then by Proposition 2.6 $NE(u, v)=G(u, v)\cup G(v, u)$.

PROPOSITION 3.2. *Let t be a trace and $u=\partial_\alpha(t)$, $v=\partial_\beta(t)$, $w=\partial_{\alpha\cup\beta}(t)$ for some α, $\beta\subseteq A$. Then there exist traces t_0, t_1, t_2 such that*

$$u=t_0t_1, \qquad v=t_0t_2, \qquad w=t_0t_1t_2, \qquad t_1\Theta t_2.$$

Furthermore, let $\tau=E(\partial_\alpha(t), \partial_\beta(t))$; then $t_0=\partial_\tau(u)=\partial_\tau(v)=\partial_\tau(w)$ and $alph(t_1)\cap\tau=alph(t_2)\cap\tau=\varnothing$.

Proof. Since $\partial_\alpha(t)$ and $\partial_\beta(t)$ are prefixes of the same trace t, we get by Proposition 2.1 and by Proposition 2.4 (3)

$$u=t_0t_1, \qquad v=t_0t_2, \qquad t_1\Theta t_2 \qquad \partial_{\alpha\cup\beta}(t)=u\vee v=t_0t_1t_2=w,$$

for some traces t_0, t_1, t_2.

For any letter a, one of the following disjoint possibilities holds:

(1) $\quad a\in \mathrm{alph}(t_1)\Leftrightarrow\partial_a(t_0)=\partial_a(v)\prec\partial_a(u)\Leftrightarrow a\in G(v, u)$,

(2) $\quad a\in \mathrm{alph}(t_2)\Leftrightarrow\partial_a(t_0)=\partial_a(u)\prec\partial_a(v)\Leftrightarrow a\in G(u, v)$,

(3) $\quad a\notin(\mathrm{alph}(t_1)\cup\mathrm{alph}(t_2))\Leftrightarrow\partial_a(t_0)=\partial_a(u)=\partial_a(v)\Leftrightarrow a\in E(u, v)$.

If $a\in\mathrm{Max}(t_0)$ then by Lemma 3.1 $a\notin\mathrm{alph}(t_1)\cup\mathrm{alph}(t_2)$ implying immediately $a\in E(u, v)=\tau$. Thus $\mathrm{Max}(t_0)\subseteq\tau$, which entails, by Proposition 2.8 (1) and by Proposition 2.5 (1), the thesis. ∎

In the rest of this section we investigate some properties of the sets $E(u, v)$ and $G(u, v)$ when $u=\partial_\alpha(t)$ and $v=\partial_\beta(t)$. Note that in this case, since u and v are prefixes of the same trace t, we have

$$NE(u, v)=G(u, v)\cup G(v, u).$$

First we give a technical lemma.

LEMMA 3.3. *Let t be a trace of $M(A, \Theta)$; let $\partial_\alpha(t)=uv$ and $a\in alph(v)$. Then there exists a sequence of letters $a_1, a_2, ..., a_k$ of $alph(v)$ such that:*

$$a_1=a,\ a_k\in\alpha,\ \forall i=1, ..., k-1,\ (a_i, a_{i+1})\in\bar{\Theta}.$$

Proof. We proceed by induction on the length of v. If $|v|=1$ then $v=a$, thus we can take $k=1$ since $a\in\alpha$. Let $v=v'av''$. If $a\Theta v''$ then we get $\partial_\alpha(t)=uv'av''=av'v''a$, hence $a\in\alpha$ and we can take $k=1$.

Now suppose that there exists $b\in\mathrm{alph}(v'')$ such that $(a, b)\in\bar{\Theta}$. Since $|v''|<|v|$, we can apply the inductive hypothesis to v'' and b to obtain a sequence $a_1, a_2, ..., a_k\in\mathrm{alph}(v'')$ such that:

$$a_1=b,\ a_k\in\alpha,\ \forall i=1, ..., k-1,\ (a_i, a_{i+1})\in\bar{\Theta}.$$

Now the sequence $a, a_1, a_2, ..., a_k$ fulfils the required condition for v. ∎

The next proposition states that if the sets α, β, and $E(\partial_\alpha(t), \partial_\beta(t))$ are given then we are also able to calculate the sets $G(\partial_\alpha(t), \partial_\beta(t))$ and $G(\partial_\beta(t), \partial_\alpha(t))$.

PROPOSITION 3.4. *Let t be a trace of $M(A, \Theta)$. Let α, β be subsets of A, and $\eta = A \setminus E(\partial_\alpha(t), \partial_\beta(t))$. Then $c \in G(\partial_\alpha(t), \partial_\beta(t))$ if and only if there exists a sequence of letters $c_1, c_2, ..., c_n$ such that:*

$$c_1 = c, \qquad c_n \in \beta, \qquad \forall i,\ 1 \leqslant i \leqslant n-1, \qquad (c_i, c_{i+1}) \in \bar{\Theta} \cap (\eta \times \eta).$$

Proof. From Proposition 3.2 we have $\partial_\alpha(t) = t_0 t_1$, $\partial_\beta(t) = t_0 t_2$, $t_1 \Theta t_2$, $\eta = \operatorname{alph}(t_1) \cup \operatorname{alph}(t_2)$, and $G(\partial_\alpha(t), \partial_\beta(t)) = \operatorname{alph}(t_2)$. Let $c \in G(\partial_\alpha(t), \partial_\beta(t))$. From Lemma 3.3 it follows that there exists a sequence $c_1, ..., c_n$ of letters in $\operatorname{alph}(t_2)$ such that $c_1 = c$, $c_n \in \beta$, and $\forall i$, $1 \leqslant i \leqslant n-1$, $(c_i, c_{i+1}) \in \bar{\Theta}$, and we are done.

Conversely, assume that there exists a sequence $c_1, ..., c_n$ of letters satisfying the required condition. Note that

$$\partial_\beta(t) = \partial_\beta(\partial_\beta(t)) \preccurlyeq \partial_\beta(\partial_\beta(t)\, t_1) = \partial_\beta(t_0 t_1 t_2) \preccurlyeq \partial_\beta(t).$$

Thus $\beta \cap \operatorname{alph}(t_1) = \varnothing$ and, since $c_n \in \beta$, we get $c_n \notin \operatorname{alph}(t_1)$. As $c_n \in \eta$, this implies that $c_n \in \operatorname{alph}(t_2)$. We prove the following assertion: for all i, $i = 1, ..., n$, $c_i \in \operatorname{alph}(t_2)$. Suppose that the assertion does not hold and that i is the greatest index such that $c_i \notin \operatorname{alph}(t_2)$. Note that $i < n$ since $c_n \in \operatorname{alph}(t_2)$. Thus $c_i \in \operatorname{alph}(t_1) = \eta \setminus \operatorname{alph}(t_2)$ and $c_{i+1} \in \operatorname{alph}(t_2)$. But $(c_i, c_{i+1}) \in \bar{\Theta}$ is in contradiction with the fact that $t_1 \Theta t_2$ and the proof of the assertion is accomplished. Hence we get $c = c_1 \in \operatorname{alph}(t_2) = G(\partial_\alpha(t), \partial_\beta(t))$. ∎

PROPOSITION 3.5. *Suppose that the sets $E(\partial_a(t), \partial_b(t))$ are known for all $a, b \in A$. Then we can determine for all subsets α, β of A the set $G(\partial_\alpha(t), \partial_\beta(t))$.*

Proof. Let $c \in A$. First note that for all $a \in \alpha$, $\partial_a(t) \preccurlyeq \partial_\alpha(t)$, which implies $\partial_c(\partial_a(t)) \preccurlyeq \partial_c(\partial_\alpha(t))$. Similarly, for all $b \in \beta$, $\partial_c(\partial_b(t)) \preccurlyeq \partial_c(\partial_\beta(t))$. On the other hand, by Corollary 2.7,

$$\partial_c(\partial_\alpha(t)) = \operatorname{Max}\{\partial_c(\partial_d(t)) \mid d \in \alpha\}$$

and

$$\partial_c(\partial_\beta(t)) = \operatorname{Max}\{\partial_c(\partial_e(t)) \mid e \in \beta\}.$$

Therefore $\partial_c(\partial_\alpha(t)) \prec \partial_c(\partial_\beta(t))$ if and only if

$$\exists e \in \beta,\ \forall a \in \alpha, \qquad \partial_c(\partial_a(t)) \prec \partial_c(\partial_e(t)).$$

Thus

$$G(\partial_\alpha(t), \partial_\beta(t)) = \{c \in A \mid \exists e \in \beta,\ \forall a \in \alpha,\ c \in G(\partial_a(t), \partial_e(t))\}.$$

But by Proposition 3.4, for all $a, e \in A$, the set $G(\partial_a(t), \partial_e(t))$ can be determined if we know the set $E(\partial_a(t), \partial_e(t))$. ∎

4. Asynchronous Mappings

In this section we introduce the notion of asynchronous mappings from $M(A, \Theta)$ into an arbitrary set X. We show that an asynchronous mapping φ into a finite set X makes it possible to construct easily, for any subset Y of X, a cellular asynchronous automaton recognizing the inverse image $\varphi^{-1}(Y)$ of Y. Thus the problem of constructing an asynchronous cellular automaton is reduced to the problem of constructing an appropriate asynchronous mapping φ. The construction of φ represents the difficult part of the paper and is fully achieved in the next section.

In the present section we build a special auxiliary asynchronous mapping v which codes some structural properties of traces. For instance, knowing the value v_t of v for a trace t, we are able to verify for any $a, b \in A$ if $\partial_a(t) \prec \partial_b(t)$ holds when t itself is not known. In other words we are able to reconstruct the prefix order of the traces $\partial_a(t)$, $a \in A$, by means of v_t. The mapping v will also be used in Section 6 as a bounded time-stamp system of messages in a special distributed system.

Definition. A mapping φ from $M(A, \Theta)$ into an arbitrary set X is uniform if for any two subsets α, β of A and for all traces u and v the following condition holds:

$$[\varphi(\partial_\alpha(u)) = \varphi(\partial_\alpha(v)) \text{ and } \varphi(\partial_\beta(u)) = \varphi(\partial_\beta(v))]$$
$$\Rightarrow \varphi(\partial_{\alpha \cup \beta}(u)) = \varphi(\partial_{\alpha \cup \beta}(v)).$$

A mapping φ is locally right regular if for all traces u and v such that ua, $va \in Pr(A, \Theta)$ the following implication holds:

$$\varphi(u) = \varphi(v) \Rightarrow \varphi(ua) = \varphi(va).$$

A mapping φ which is uniform and locally right regular is called an asynchronous mapping.

Definition. We say that an asynchronous mapping φ from $M(A, \Theta)$ into an arbitrary set X recognizes a subset T of $M(A, \Theta)$ if $T = \varphi^{-1}(Y)$ for some subset Y of X.

Note that the uniformity condition can be coded by means of a family of tables $\{U_{\alpha,\beta} \mid \alpha, \beta \subseteq A\}$. The rows and the columns of every table $U_{\alpha,\beta}$ are indexed by elements of the set X and if $x_1, x_2 \in X$ are such that $\varphi(\partial_\alpha(t)) = x_1$, $\varphi(\partial_\beta(t)) = x_2$ for some trace t then $U_{\alpha,\beta}[x_1, x_2] = x_3$, where $x_3 = \varphi(\partial_{\alpha \cup \beta}(t))$. Uniformity of φ ensures the consistency of this coding.

Similarly, the local right regularity can be coded by means of a single table R indexed by elements of X and A. The table R is such that if $x_1 = \varphi(u)$ and $r = ua \in Pr(A, \Theta)$ for some letter a and traces u, r then $R[x_1, a] = x_2$, where $x_2 = \varphi(r)$. Given $\varphi(\varepsilon)$, $\{U_{\alpha,\beta} \mid \alpha, \beta \subseteq A\}$ and R it is possible to compute by induction on the prefix order the value $\varphi(t)$ for any trace t. To see this, suppose that $t \neq \varepsilon$ and the value of t is known for all proper prefixes r of t, $r \prec t$. We consider two cases.

If $t \in Pr(A, \Theta)$ then $t = ua$ for some trace u and a letter a. Using the table R we get $\varphi(t) = R[\varphi(u), a]$.

Now suppose that t is not prime. Then $\mathrm{Card}(\mathrm{Max}(t)) > 1$ and there exist nonempty disjoint sets α and β such that $\alpha \cup \beta = \mathrm{Max}(t)$. Since α and β are proper subsets of $\mathrm{Max}(t)$ by Proposition 2.8 (1) $\partial_\alpha(t)$ and $\partial_\beta(t)$ are proper prefixes of t and $t = \partial_{\alpha \cup \beta}(t)$. By the inductive hypothesis $x_1 = \varphi(\partial_\alpha(t))$ and $x_2 = \varphi(\partial_\beta(t))$ are given and, using the table $U_{\alpha,\beta}$, we get $\varphi(t) = U_{\alpha,\beta}[x_1, x_2]$.

The crucial role of asynchronous mappings is shown by the following theorem which states the principal result of this paper.

THEOREM 4.1. *For any subset T of $M(A, \Theta)$ the following statements are equivalent*:

(1) *T is recognizable,*

(2) *there exists an asynchronous mapping φ from $M(A, \Theta)$ into a finite set X recognizing T,*

(3) *there exists a deterministic asynchronous cellular automaton recognizing T.*

Proof. An asynchronous cellular automaton is also an $M(A, \Theta)$-automaton; thus the implication $(3) \Rightarrow 1$ holds.

The implication $(1) \Rightarrow (2)$ needs further developments and is delayed for Section 5.

To prove $(2) \Rightarrow (3)$ we construct an asynchronous cellular automaton from the asynchronous mapping φ. We choose X as the set of basic states. For every trace u we define a global state $s^u \in S = F(A; X)$ in the following way:

$$\forall a \in A, \qquad s^u(a) = \varphi(\partial_a(u)).$$

We construct the automaton $\mathcal{A}$ in such a way that s^u will be the global state reached in $\mathcal{A}$ after the execution of the trace u.

First note that if u is a trace and a a letter then

$$\forall b \in A \setminus \{a\}, \qquad s^{ua}(b) = s^u(b). \tag{1}$$

Indeed, $s^{ua}(b) = \varphi(\partial_b(ua)) = \varphi(\partial_b(u)) = s^u(b)$. Let us recall that $s^u|_\alpha$ denotes the restriction of the mapping s^u to the set $\alpha \subseteq A$. We prove that

$$\forall a \in A,\ \forall u, v \in M(A, \Theta), \qquad s^u|_{\bar{\Theta}(a)} = s^v|_{\bar{\Theta}(a)} \Rightarrow s^{ua}(a) = s^{va}(a). \tag{2}$$

The equality

$$s^u|_{\bar{\Theta}(a)} = s^v|_{\bar{\Theta}(a)}$$

implies that

$$\forall b \in \bar{\Theta}(a),\ s^u(b) = \varphi(\partial_b(u)) = \varphi(\partial_b(v)) = s^v(b).$$

By uniformity of φ we get

$$\varphi(\partial_{\bar{\Theta}(a)}(u)) = \varphi(\partial_{\bar{\Theta}(a)}(v)).$$

Now note that $\partial_a(ua) = \partial_{\bar{\Theta}(a)}(u)\, a$ and similarly $\partial_a(va) = \partial_{\bar{\Theta}(a)}(v)\, a$, thus by the previous equality and the local right regularity of φ we obtain:

$$\varphi(\partial_a(ua)) = \varphi(\partial_a(va)),$$

i.e.,

$$s^{ua}(a) = s^{va}(a).$$

Now we are able to define the transition mappings of $\mathcal{A}$.

If $a \in A$, $\alpha = \bar{\Theta}(a)$, $s_\alpha \in F(\alpha; X)$, $s_a \in X$ then $s_a = \delta_a(s_\alpha)$ if and only if there exists a trace $u \in M(A, \Theta)$ such that

$$s^{ua}(a) = s_a \qquad \text{and} \qquad s^u|_\alpha = s_\alpha.$$

Note that (1) and (2) ensure the correctness of this definition. Moreover if Δ is the global transition mapping induced by the local transition mappings defined above then we have for every trace u

$$s^u = \Delta(s^\varepsilon, u). \tag{3}$$

Finally we choose s^ε as the initial state and $F = \{s^u \mid u \in T\}$ as the set of final states. From (3) it is clear that $T \subseteq T(\mathcal{A})$. To prove that $\mathcal{A}$ recognizes T, i.e.,

$T = T(A)$, suppose that $s^u = s^v$. Thus $\forall a \in A$, $\varphi(\partial_a(u)) = \varphi(\partial_a(v))$ and by uniformity of φ, $\varphi(u) = \varphi(\partial_A(u)) = \varphi(\partial_A(v)) = \varphi(v)$. This proves that

$$u \in T = \varphi^{-1}(Y) \Leftrightarrow v \in T = \varphi^{-1}(Y). \quad \blacksquare$$

Note that for any two asynchronous mappings φ_1 and φ_2 from $M(A, \Theta)$ into X_1 and X_2 respectively the mapping $\varphi = \varphi_1 \times \varphi_2$ from $M(A, \Theta)$ into $X_1 \times X_2$ defined by $\varphi(t) = (\varphi_1(t), \varphi_2(t))$ is again an asynchronous mapping.

Let us give here some examples of asynchronous mappings. Of course any constant mapping and the identity mapping are asynchronous mappings. Note that the last one maps $M(A, \Theta)$ into itself, that is, into an infinite set. By the result obtained in the preceding section the operator ∂_a, for any $a \in A$, is uniform and as it is easy to observe it is also locally right regular, thus giving a third example of such a mapping.

Another asynchronous mapping, called App_1, is obtained by associating with any trace t the set $\{\partial_a(t) \mid a \in A\}$. Similarly the mapping App_2 associating with any trace t the set $\{\partial_a(\partial_b(t)) \mid a, b \in A\}$ is asynchronous.

For any mapping φ we are interested in the equivalence relation $\sim_\varphi$ over $M(A, \Theta)$ associated with φ and defined by

$$u \sim_\varphi v \qquad \text{if} \quad \varphi(u) = \varphi(v).$$

Note that the identity mapping and the mappings App_1 and App_2 induce the same equivalence relation which is equality in $M(A, \Theta)$, having of course an infinite number of equivalence classes. We wish to define here a mapping carrying enough information on a trace t and having a finite image, thus inducing an equivalence of a finite index. Information which seems structurally essential in any trace t is the prefix order on the traces $\partial_a(t)$ for $a \in A$. This idea suggests to define for every trace t the set

$$\mathrm{front}_1(t) = \{(a, b) \in A \mid \partial_a(t) \prec \partial_b(t)\}.$$

The mapping front_1, which is a natural candidate for "approximation" of a trace, is not uniform, as the following example shows.

EXAMPLE. Let $A = \{a, b, c\}$ and $\Theta = \{(a, b), (b, a)\}$, $u = abcabca$, $v = abcabcab$. Then $\partial_a(u) = \partial_a(v) = abcabca$, $\partial_b(u) = abcb$, $\partial_b(v) = abcabcb$.

We get

$$\mathrm{front}_1(\partial_a(u)) = \mathrm{front}_1(\partial_a(v)), \ \mathrm{front}_1(\partial_b(u)) = \mathrm{front}_1(\partial_b(v))$$

but

$$\mathrm{front}_1(\partial_{\{a,b\}}(u)) \neq \mathrm{front}_1(\partial_{\{a,b\}}(v)).$$

It seems that the ambiguity that does not allow determination of $\mathrm{front}_1(\partial_{\alpha \cup \beta}(u))$ from $\mathrm{front}_1(\partial_\alpha(u))$ and $\mathrm{front}_1(\partial_\beta(u))$ is caused by the deficiency in the information coded (provided) by front_1 and thus we repeat our attempt with a more precise mapping front_2, given by the following definition:

$$\mathrm{front}_2(t) = \{((a, b), (c, d)) \mid \partial_a(\partial_b(t)) \prec \partial_c(\partial_d(t))\}.$$

It is obvious that front_2 codes simply the prefix relation of elements of $\mathrm{App}_2(t)$ and it is a refinement of front_1. But another example analogous to the preceding shows that front_2 is not uniform. Of course, it is possible to generalize these constructions by defining front_k for any integer k but it turns out that none of these mappings is uniform. Examining carefully the example given above we observe that the difficulty is due to the impossibility of determining which letters match in $\mathrm{front}_1(\partial_\alpha(u))$ and $\mathrm{front}_1(\partial_\beta(u))$ if u is unknown. In other words we cannot verify if $\partial_c(\partial_\alpha(u)) = \partial_c(\partial_\beta(u))$ if we only know $\mathrm{front}_1(\partial_\alpha(u))$ and $\mathrm{front}_1(\partial_\beta(u))$.

The main idea making it possible to eliminate this weakness is labelling the elements of $Pr(A, \Theta)$ in such a way that for any trace t, any subsets α, β of A, and any letter c the equality $\partial_c(\partial_\alpha(t)) = \partial_c(\partial_\beta(t))$ holds if and only if the labels of these two traces are equal. These labels will provide the information needed for the reconstruction of $\mathrm{front}_1(\partial_{\alpha \cup \beta}(u))$ from $\mathrm{front}_1(\partial_\alpha(u))$ and $\mathrm{front}_1(\partial_\beta(u))$. We thus define a labelling λ from $Pr(A, \Theta)$ into the set of positive integers, $\mathbb{N}_+$, and show that it has the desired properties.

Definition. The label mapping λ from $Pr(A, \Theta)$ into the set of positive integers is inductively defined by

$$\lambda(\varepsilon) = \mathrm{card}(A)$$

for $a \in A$, $t \in M(A, \Theta)$ such that $ta \in Pr(A, \Theta)$ $\lambda(ta) = \mathrm{Min}\{i \in \mathbb{N}_+ \mid \forall b \in A \setminus \{a\}\ i \neq \lambda(\partial_a(\partial_b(t)))\}$.

From this mapping λ we construct for every trace t a mapping v_t from $A \times A$ into $\mathbb{N}_+$:

Definition. For any trace t in $M(A, \Theta)$ and any pair of letters a, b in A we set

$$v_t(a, b) = \lambda(\partial_a(\partial_b(t))).$$

Intuitively, for each trace t of $M(A, \Theta)$ the mapping v_t gives the labels of elements of $\mathrm{App}_2(t)$. In the sequel we prove that for any prime trace u of $Pr(A, \Theta)$ we have $1 \leqslant \lambda(u) \leqslant \mathrm{card}(A)$. This implies that for each t

of $M(A, \Theta)$ the mapping ν_t belongs in fact to the finite set $F(A \times A; \{1, ..., \mathrm{card}(A)\})$ of mappings. Moreover, it turns out that the mapping ν associating with a trace t of $M(A, \Theta)$ the element ν_t of $F(A \times A; \{1, ..., \mathrm{card}(A)\})$ is asynchronous. This is the first non-trivial example of an asynchronous mapping with a finite codomain. Its importance is emphasized by the fact that all asynchronous mappings recognizing a given set T of traces that are constructed in the next section will be obtained by refinements of this basic mapping ν. The following technical lemma is crucial for establishing the basic properties of the mapping ν_t. It states that if $v \preccurlyeq u$ and r is an element of $\mathrm{App}_2(u)$ such that $r \preccurlyeq v$ then r belongs to $\mathrm{App}_2(v)$.

LEMMA 4.2. *Let $u, v \in M(A, \Theta)$ and $a, b \in A$ be such that*

$$\partial_a(\partial_b(u)) \preccurlyeq v \preccurlyeq u.$$

Then there exists a letter c of A such that $\partial_a(\partial_c(v)) = \partial_a(\partial_b(u))$.

Proof. Let us first prove the following assertion: for any trace t and any letter a there exists a letter b of $\mathrm{Max}(t)$ such that $\partial_a(t) = \partial_a(\partial_b(t))$.

If we let $\alpha = \mathrm{Max}(t)$, then $t = \partial_\alpha(t)$ and by Corollary 2.7 $\partial_a(t) = \partial_a(\partial_\alpha(t)) = \partial_a(\partial_b(t))$ for some $b \in \alpha$.

We now return to the lemma's proof. Since $\partial_b(u)$ and v are prefixes of u, there exist, by Proposition 2.1, traces t_0, t_1, t_2 such that $\partial_b(u) = t_0 t_1$, $v = t_0 t_2$ and $t_1 \Theta t_2$. As $\partial_a(\partial_b(u)) \preccurlyeq v$, we get $a \notin \mathrm{alph}(t_1)$ and $\partial_a(\partial_b(u)) = \partial_a(t_0)$. By the assertion, there exists a letter c of $\mathrm{Max}(t_0)$ such that $\partial_a(t_0) = \partial_a(\partial_c(t_0))$. By Lemma 3.1 $\mathrm{alph}(t_2) \cap \mathrm{Max}(t_0) = \varnothing$, which implies that $\partial_c(t_0) = \partial_c(t_0 t_2) = \partial_c(v)$ and $\partial_a(\partial_b(u)) = \partial_a(t_0) = \partial_a(\partial_c(t_0)) = \partial_a(\partial_c(v))$ as required. ∎

Note that in general the letters b and c used in the statement of Lemma 4.2 are different. Let us take for instance $u = acb$, $v = a$, $\Theta = \varnothing$. Then $\partial_a(\partial_b(u)) = a \preccurlyeq v$ and $\partial_a(\partial_b(u)) = \partial_a(\partial_a(v))$; i.e., we can take $c = a$, while $\varepsilon = \partial_a(\partial_b(v)) \prec \partial_a(\partial_b(u))$, and $c = b$ does not fulfil the condition.

PROPOSITION 4.3. *The mapping ν from $M(A, \Theta)$ into $F(A \times A; \mathbb{N}_+)$ associating with every trace t the function ν_t satisfies the following conditions*: $\forall a, b, c \in A$, $\forall t, r \in M(A, \Theta)$,

(ν1) $1 \leqslant \nu_t(a, b) \leqslant \mathrm{card}(A)$,

(ν2) $\partial_a(\partial_b(t)) = \partial_a(\partial_c(r)) \Rightarrow \nu_t(a, b) = \nu_r(a, c)$,

(ν3) $\partial_a(\partial_b(t)) = \partial_a(\partial_c(t)) \Leftrightarrow \nu_t(a, b) = \nu_t(a, c)$,

(ν4) $(ta, ra \in Pr(A, \Theta)$ *and* $\nu_t = \nu_r) \Rightarrow \nu_{ta} = \nu_{ra}$.

Proof. ($\nu 1$) We return to the definition of λ and proceed by induction on the length of $t \in Pr(A, \Theta)$ to prove that $\lambda(t) \leqslant \text{card}(A)$.

The set $\{\lambda(\partial_a(\partial_b(t))) \mid b \in A \setminus \{a\}\}$ contains at most $\text{card}(A) - 1$ distinct integers, and by the inductive hypothesis they are all not greater than $\text{card}(A)$. Hence the minimal element of $\mathbb{N}_+$ that does not belong to this set is also not greater than $\text{card}(A)$.

($\nu 2$) is an immediate consequence of the definition of ν.

($\nu 3$) It suffices to prove that $\nu_t(a, b) = \nu_t(a, c) \Rightarrow \partial_a(\partial_b(t)) = \partial_a(\partial_c(t))$ as the converse is a restricted form of ($\nu 2$).

Let $a, b, c \in A$ be such that

$$\lambda(\partial_a(\partial_b(t))) = \lambda(\partial_a(\partial_c(t))). \tag{1}$$

Because of the symmetric role of b and c and by Proposition 2.6 we may assume without loss of generality that

$$\partial_a(\partial_b(t)) \preccurlyeq \partial_a(\partial_c(t)).$$

Setting $r = \partial_a(\partial_c(t))$ we have

$$\partial_a(\partial_b(t)) \preccurlyeq r \preccurlyeq t.$$

The case $r = \varepsilon$ being trivial, we assume that $r \neq \varepsilon$. By Lemma 4.2 there exists a letter d of A such that

$$\partial_a(\partial_b(t)) = \partial_a(\partial_d(r)). \tag{2}$$

Thus

$$\lambda(\partial_a(\partial_b(t))) = \lambda(\partial_a(\partial_d(r))),$$

and as a consequence of (1) we obtain

$$\lambda(r) = \lambda(\partial_a(\partial_d(r))).$$

Since $r = \partial_a(\partial_c(t))$ and $r \neq \varepsilon$, there exists a trace u such that $r = ua$. Suppose that $d \neq a$, then $\partial_d(r) = \partial_d(ua) = \partial_d(u)$, yielding $\lambda(r) = \lambda(\partial_a(\partial_d(u)))$, which is in contradiction with the definition of λ. Thus we see that $d = a$ which implies by (2)

$$\partial_a(\partial_b(t)) = \partial_a(\partial_a(r)) = \partial_a(\partial_a(\partial_a(\partial_c(t)))) = \partial_a(\partial_c(t)).$$

($\nu 4$) Let $ta \in Pr(A, \Theta)$. We shall show that ν_{ta} is determined by ν_t which obviously implies ($\nu 4$). Let $c \in A$, $c \neq a$. Then $\partial_c(ta) = \partial_c(t)$ and thus for all $b \in A$ $\nu_{ta}(b, c) = \nu_t(b, c)$. Similarly, since $\partial_a(ta) = ta$, we get

$$\nu_{ta}(c, a) = \lambda(\partial_c(\partial_a(ta))) = \lambda(\partial_c(ta)) = \lambda(\partial_c(t)) = \lambda(\partial_c(\partial_c(t))) = \nu_t(c, c).$$

Finally the last case to examine is

$v_{ta}(a, a) = \lambda(\partial_a(\partial_a(ta))) = \lambda(ta) = \mathrm{Min}\{i \in \mathbb{N}_+ \mid \forall b \in A \setminus \{a\}\ i \neq v_t(a, b)\}.$ ▮

Let us recall that $E(u, v) = \{c \in A \mid \partial_c(u) = \partial_c(v)\}$. The following proposition shows how to compute the set $E(\partial_\alpha(t), \partial_\beta(t))$ if the mappings $v_{\partial_\alpha(t)}$ and $v_{\partial_\beta(t)}$ are given.

PROPOSITION 4.4. *Let t be a trace, and $u = \partial_\alpha(t)$, $v = \partial_\beta(t)$. Moreover, let v be a mapping verifying conditions* (v2), (v3), (v4) *of Proposition* 4.3. *Then*

$$E(u, v) = \{a \in A \mid v_u(a, a) = v_v(a, a)\}.$$

Proof. Let $a \in A$. If $a \in E(u, v)$ then $\partial_a(\partial_\alpha(t)) = \partial_a(\partial_\beta(t))$. Hence $\partial_a(\partial_a(\partial_\alpha(t))) = \partial_a(\partial_a(\partial_\beta(t)))$ and by (v2) $v_u(a, a) = v_v(a, a)$.

Conversely, let $v_u(a, a) = v_v(a, a)$. This implies that $\lambda(\partial_a(\partial_\alpha(t))) = \lambda(\partial_a(\partial_\beta(t)))$. Now note that by Corollary 2.7 there exist letters $c \in \alpha$ and $d \in \beta$ such that

$$\partial_a(\partial_\alpha(t)) = \partial_a(\partial_c(t)) \qquad \text{and} \qquad \partial_a(\partial_\beta(t)) = \partial_a(\partial_d(t)),$$

and by the previous equality

$$v_t(a, c) = \lambda(\partial_a(\partial_c(t))) = \lambda(\partial_a(\partial_d(t))) = v_t(a, d).$$

Applying (v3) to this formula we get

$$\partial_a(\partial_c(t)) = \partial_a(\partial_d(t)),$$

i.e.,

$$\partial_a(\partial_\alpha(t)) = \partial_a(\partial_\beta(t)),$$

which means that $a \in E(u, v)$. ▮

Now we are able to prove that the mapping v is asynchronous.

THEOREM 4.5. *Any mapping v from $M(A, \Theta)$ into $F(A \times A$; $\{1, ..., \mathrm{card}(A)\})$ associating with every trace t a function v_t satisfying conditions* (v1), (v2), (v3), (v4) *is asynchronous.*

Proof. Since (v4) states local right regularity of v, we only have to prove uniformity. Let $u = \partial_\alpha(t)$, $v = \partial_\beta(t)$, and $w = \partial_{\alpha \cup \beta}(t)$ for some subsets α, β of A and a trace t. By Proposition 3.2 we have $u = t_0 t_1$, $v = t_0 t_2$, $w = t_0 t_1 t_2$, $t_1 \Theta t_2$ and setting $\tau = E(u, v)$, $\mathrm{alph}(t_1) \cap \tau = \mathrm{alph}(t_2) \cap \tau = \varnothing$.

We should prove that ν_w is entirely determined by ν_u and ν_v. Observe now that for any $b \in A$ we have

$$\partial_b(w) = \begin{cases} \partial_b(u) & \text{if} \quad b \in \tau \cup G(v, u), \\ \partial_b(v) & \text{if} \quad b \in \tau \cup G(u, v). \end{cases}$$

Hence for all $a, b \in A$,

$$\nu_w(a, b) = \begin{cases} \nu_u(a, b) & \text{if} \quad b \in \tau \cup G(v, u), \\ \nu_v(a, b) & \text{if} \quad b \in \tau \cup G(u, v). \end{cases}$$

Now note that by Proposition 4.4 the set $\tau = E(u, v)$ can be calculated by means of ν_u and ν_v and, in turn, by Proposition 3.4 the sets $G(u, v)$ and $G(v, u)$ can be calculated from τ, α and β. ∎

We end this section with some remarks concerning the mapping ν. First of all, note that ν_t makes it possible to reconstruct the prefix order of the elements of $\mathrm{App}_1(t) = \{\partial_a(t) \mid a \in A\}$. To see this we present the equivalences

$$\partial_a(t) \preccurlyeq \partial_b(t) \Leftrightarrow \partial_a(t) = \partial_a(\partial_b(t)) \Leftrightarrow \nu_t(a, a) = \nu_t(a, b),$$

for any $a, b \in A$, $t \in M(A, \Theta)$. Thus ν_t is a refinement of the mapping front_1 considered previously, but in contrast to front_1 the mapping ν is asynchronous.

The asynchronous cellular automaton $\mathcal{A}$ obtained from ν by the canonical construction given in the proof of Theorem 4.1 also exhibits some remarkable features.

Let s^u be the global state reached after the execution of a trace u. By the construction,

$$\forall a \in A, \qquad s^u(a) = \nu_{\partial_a(u)}.$$

Now observe that the following equivalences hold for all $a, b \in A$:

$$\begin{aligned} &\partial_a(u) \preccurlyeq \partial_b(u) \Leftrightarrow \partial_a(u) = \partial_a(\partial_b(u)) \Leftrightarrow \partial_a(\partial_a(\partial_a(u))) = \partial_a(\partial_a(\partial_b(u))) \\ &\quad \Leftrightarrow \nu_{\partial_a(u)}(a, a) = \nu_{\partial_b(u)}(a, a). \end{aligned}$$

The last equality can obviously be tested if the local states of the agents a and b, $s^u(a)$ and $s^u(b)$, are given. By a symmetric condition we can check if $\partial_b(u) \preccurlyeq \partial_a(u)$ and if neither of these two conditions holds then $\partial_a(u)$ and $\partial_b(u)$ are incomparable.

Recapitulating, we have obtained an asynchronous cellular automaton such that for any trace u, for any agents a, b we can check which holds of the three conditions:

(1) $\partial_a(u) \preccurlyeq \partial_b(u)$,

(2) $\partial_b(u) \preccurlyeq \partial_a(u)$,

(3) $\partial_a(u)$ and $\partial_b(u)$ are incomparable,

by means of the local states $s^u(a)$ and $s^u(b)$ of these agents. It is not an easy task to construct an asynchronous cellular automaton with the property described above even for very simple dependency graphs. To appreciate the problem the reader may try to do it in the case $A = \{a, b, c, d\}$, and $\Theta = \{(a, c), (c, a), (b, d), (d, b)\}$.

5. Asynchronous Mappings for Recognizable Sets of Traces

In this section we accomplish the proof of the main theorem. For a given recognizable trace language T we construct asynchronous mapping with a finite codomain recognizing T. In fact we present two different constructions of asynchronous mappings. Both of them are obtained by augmenting the basic asynchronous mapping ν constructed in the previous section. We begin with a small technical subsection, where some class of suffixes of traces is examined.

A. *α-Suffixes of Traces*

Up to now, we were interested only in prefixes of traces. In this subsection we investigate suffixes of a special form. First a notational remark. For any subset α of A, by $\bar{\alpha}$ we denote the complement of α, $\bar{\alpha} = A \setminus \alpha$.

Definition. Let t be a trace, and $\alpha \subseteq A$. By $\nabla_\alpha(t)$ we denote the suffix of t corresponding to the prefix $\partial_{\bar{\alpha}}(t)$, i.e., the suffix such that

$$t = \partial_{\bar{\alpha}}(t)\, \nabla_\alpha(t).$$

Note that $\mathrm{alph}(\nabla_\alpha(t)) \subseteq \alpha$ and it is the longest suffix with this property. Therefore any suffix u of t such that $\mathrm{alph}(u) \subseteq \alpha$ is also a suffix of $\nabla_\alpha(t)$. This fact will be used in the proof of the following proposition which gives a simple formula for $\nabla_\alpha(\nabla_\beta(t))$.

Proposition 5.1. *For any trace t and any subsets α, β of A,*

$$\nabla_\alpha(\nabla_\beta(t)) = \nabla_{\alpha \cap \beta}(t).$$

Proof. The trace $\nabla_{\alpha \cap \beta}(t)$ is a suffix of $\nabla_\beta(t)$ and $\mathrm{alph}(\nabla_{\alpha \cap \beta}(t)) \subseteq \alpha \cap \beta \subseteq \alpha$. Since $\nabla_\alpha(\nabla_\beta(t))$ is the longest suffix of $\nabla_\beta(t)$ containing only letters from α, this implies that $\nabla_{\alpha \cap \beta}(t)$ is a suffix of $\nabla_\alpha(\nabla_\beta(t))$.

Conversely, $\nabla_\alpha(\nabla_\beta(t))$ is a suffix of t and it contains only letters from $\alpha \cap \beta$. But $\nabla_{\alpha\cap\beta}(t)$ is the longest suffix of t with this property. Therefore $\nabla_\alpha(\nabla_\beta(t))$ is a suffix of $\nabla_{\alpha\cap\beta}(t)$. ∎

PROPOSITION 5.2. *For any traces u and v and any subset α of A,*

$$\nabla_\alpha(uv) = \nabla_{\alpha\setminus\beta}(u)\,\nabla_\alpha(v),$$

where β is given by $\beta = \bar{\Theta}(\mathrm{alph}(\partial_{\bar{\alpha}}(v)))$.

Proof. By Proposition 2.5 we obtain

$$\partial_{\bar{\alpha}}(uv) = \partial_{\bar{\alpha}\cup\beta}(u)\,\partial_{\bar{\alpha}}(v), \tag{1}$$

where $\beta = \bar{\Theta}(\mathrm{alph}(\partial_{\bar{\alpha}}(v)))$. Since

$$\begin{aligned} u &= \partial_{\bar{\alpha}\cup\beta}(u)\,\nabla_{\bar{\beta}\cap\alpha}(u),\\ v &= \partial_{\bar{\alpha}}(v)\,\nabla_\alpha(v),\\ uv &= \partial_{\bar{\alpha}}(uv)\,\nabla_\alpha(uv), \end{aligned}$$

we get

$$uv = \partial_{\bar{\alpha}\cup\beta}(u)\,\nabla_{\bar{\beta}\cap\alpha}(u)\,\partial_{\bar{\alpha}}(v)\,\nabla_\alpha(v) = \partial_{\bar{\alpha}}(uv)\,\nabla_\alpha(uv).$$

This fact and (1) imply by the cancellative property of $M(A, \Theta)$ that

$$\nabla_{\bar{\beta}\cap\alpha}(u)\,\partial_{\bar{\alpha}}(v)\,\nabla_\alpha(v) = \partial_{\bar{\alpha}}(v)\,\nabla_\alpha(uv). \tag{2}$$

As $\mathrm{alph}(\nabla_{\bar{\beta}\cap\alpha}(t)) \subseteq \bar{\beta}$, for any trace t, by the definition of β the two traces $\nabla_{\bar{\beta}\cap\alpha}(u)$ and $\partial_{\bar{\alpha}}(v)$ are independent and commute, and applying the cancellative property to (2) we get the result. ∎

COROLLARY 5.3. *For any trace u and any subset α of A and a letter $a \in A$*

$$\begin{aligned} a \in \alpha &\Rightarrow \nabla_\alpha(ua) = \nabla_\alpha(u)\,a,\\ a \notin \alpha &\Rightarrow \nabla_\alpha(ua) = \nabla_{\alpha\setminus\Theta(a)}(u). \end{aligned}$$

Proof. Choose $v = a$ in Proposition 5.2 and observe that $\partial_{\bar{\alpha}}(a) = a$ if $a \notin \alpha$ and $\partial_{\bar{\alpha}}(a) = \varepsilon$ if $a \in \alpha$. ∎

Now we prove a useful property linking the mappings ∇ and ν:

LEMMA 5.4. *For any trace t of $M(A, \Theta)$ and any subsets α and β of A, a mapping ν_t verifying conditions* (ν1), (ν2), (ν3), (ν4) *of Proposition* 4.3 *determines uniquely the set* $\mathrm{alph}(\partial_\alpha(\nabla_\beta(t)))$.

Proof. Observe that by (ν3), for all $a, b \in A$,

$$E(\partial_a(t), \partial_b(t)) = \{c \in A \mid \nu_t(c, a) = \nu_t(c, b)\}.$$

Hence by Proposition 3.5, if ν_t is given we can find the set $G(\partial_{\bar\beta}(t), \partial_\alpha(t))$.

Now we show that $\mathrm{alph}(\partial_\alpha(\nabla_\beta(t))) = G(\partial_{\bar\beta}(t), \partial_\alpha(t))$, which accomplishes the proof. Using the definition of ∇ and Proposition 2.5 we get

$$\partial_\alpha(t) = \partial_\alpha(\partial_{\bar\beta}(t)\,\nabla_\beta(t)) = \partial_{\alpha \cup \gamma}(\partial_{\bar\beta}(t))\,\partial_\alpha(\nabla_\beta(t)), \tag{1}$$

where $\gamma = \bar\Theta(\mathrm{alph}(\partial_\alpha(\nabla_\beta(t))))$.

If $c \in \mathrm{alph}(\partial_\alpha(\nabla_\beta(t)))$ then from the formula above we get

$$\partial_c(\partial_{\alpha \cup \gamma}(\partial_{\bar\beta}(t))) \prec \partial_c(\partial_\alpha(t)).$$

But since $c \in \mathrm{alph}(\partial_\alpha(\nabla_\beta(t))) \subseteq \gamma$, by Proposition 2.4 (2) we obtain $\partial_c(\partial_{\alpha \cup \gamma}(\partial_{\bar\beta}(t))) = \partial_c(\partial_{\bar\beta}(t))$, and finally $\partial_c(\partial_{\bar\beta}(t)) \prec \partial_c(\partial_\alpha(t))$; i.e., $c \in G(\partial_{\bar\beta}(t), \partial_\alpha(t))$.

Now let us consider the case $c \notin \mathrm{alph}(\partial_\alpha(\nabla_\beta(t)))$. Applying ∂_c to the formula (1) we get

$$\partial_c(\partial_\alpha(t)) = \partial_c(\partial_{\alpha \cup \gamma}(\partial_{\bar\beta}(t))).$$

But $\partial_{\alpha \cup \gamma}(\partial_{\bar\beta}(t)) \preccurlyeq \partial_{\bar\beta}(t)$ and thus we have

$$\partial_c(\partial_\alpha(t)) \preccurlyeq \partial_c(\partial_{\bar\beta}(t)), \qquad \text{i.e., } c \notin G(\partial_{\bar\beta}(t), \partial_\alpha(t)). \quad \blacksquare$$

B. *Asynchronous Mapping—First Construction*

In the theorem below we present a construction of an asynchronous mapping recognizing a given recognizable trace language T. This construction is obtained by adding to the basic asynchronous mapping ν_t information about suffixes of the traces of T.

THEOREM 5.5. *Let T be a recognizable subset of $M(A, \Theta)$ and let f be a homomorphism from $M(A, \Theta)$ into a finite monoid H such that $T = f^{-1}(G)$ for some subset G of H. For any trace t, by f^*_t we denote the function from $\mathscr{P}(A)$ into H that maps any subset α of A to $f(\nabla_\alpha(t))$.*

Then the mapping φ that maps any trace t of $M(A, \Theta)$ to $\varphi(t) = (\nu_t, f^*_t)$ is an asynchronous mapping recognizing T.

Proof. First note that if we know the value $\varphi(t)$ it is possible to verify if t belongs to T or not; namely, the following equivalences hold:

$$t \in T \Leftrightarrow f(t) \in G \Leftrightarrow f(\nabla_A(t)) \in G \Leftrightarrow f^*_t(A) \in G.$$

Thus φ recognizes T.

It remains to prove that φ is locally right regular and uniform. At the beginning we show that f^* is locally right regular. Note that, since v is locally right regular, this fact will prove that φ is locally right regular also. Let $u \in M(A, \Theta)$ and $a \in A$ be such that $ua \in Pr(A, \Theta)$. Let $\alpha \subseteq A$. Then

$$\nabla_\alpha(ua) = \begin{cases} \nabla_\alpha(u)\, a & \text{if } a \in \alpha, \\ \nabla_{\alpha \setminus \Theta(a)}(u) & \text{if } a \notin \alpha, \end{cases}$$

thus

$$f^*_{ua}(\alpha) = \begin{cases} f^*_u(\alpha)\, f(a) & \text{if } a \in \alpha, \\ f^*_u(\alpha \setminus \bar{\Theta}(a)) & \text{if } a \notin \alpha, \end{cases}$$

and we see that f^*_{ua} is entirely determined by f^*_u and a; i.e., f^* is locally right regular.

We proceed to the proof of uniformity of φ. Let $u = \partial_\alpha(t)$, $v = \partial_\beta(t)$, $w = \partial_{\alpha \cup \beta}(t)$. We show how to compute $\varphi(w)$ by means of $\varphi(u)$ and $\varphi(v)$. First of all, since v is uniform, we can find v_w using v_u and v_v. Now to compute f^*_w we need not only f^*_u and f^*_v but also v_u and v_v.

By Proposition 3.2,

$$u = t_0 t_1,\ v = t_0 t_2,\ t_1 \Theta t_2,\ t_0 = \partial_\tau(v), \qquad \text{where} \quad \tau = E(u, v).$$

Thus $\partial_\tau(v)\, \nabla_{\bar{\tau}}(v) = v = t_0 t_2 = \partial_\tau(v)\, t_2$ and by cancellative property $t_2 = \nabla_{\bar{\tau}}(v)$. Hence we obtain

$$w = t_0 t_1 t_2 = u t_2 = u\, \nabla_{\bar{\tau}}(v). \tag{1}$$

Let $\gamma \subseteq A$. We compute $f^*_w(\gamma) = f(\nabla_\gamma(w))$. Using Proposition 5.2 and Proposition 5.1 we get from (1)

$$\nabla_\gamma(w) = \nabla_\gamma(u\, \nabla_{\bar{\tau}}(v)) = \nabla_{\gamma \setminus \rho}(u)\, \nabla_\gamma(\nabla_{\bar{\tau}}(v)) = \nabla_{\gamma \setminus \rho}(u)\, \nabla_{\gamma \cap \bar{\tau}}(v),$$

where $\rho = \bar{\Theta}(\mathrm{alph}(\partial_{\bar{\gamma}}(\nabla_{\bar{\tau}}(v))))$.

Now applying the homomorphism f to this equality we obtain

$$f^*_w(\gamma) = f^*_u(\gamma \setminus \rho)\, f^*_v(\gamma \cap \bar{\tau}).$$

In this way, we have a formula that describes f^*_w in terms of f^*_u and f^*_v. Now we have to determine the unknown sets ρ and τ that appear in the formula; this will be done using properties of v.

First, Proposition 4.4 makes it possible to determine $\tau = E(u, v)$ by means of v_u and v_v. And finally, by Lemma 5.4, the set $\mathrm{alph}(\partial_{\bar{\gamma}}(\nabla_{\bar{\tau}}(v)))$, and consequently the set ρ, are determined by v_v. ∎

Theorem 5.5 proves the implication (1) ⇒ (2) of Theorem 4.1, giving explicitly the construction of an asynchronous mapping φ. At the end we would like to point out that since $\nu_t \in F(A \times A; \{1, ..., \mathrm{card}(A)\})$ and $f^*_t \in F(\mathscr{P}(A); H)$, the codomain of φ is equal to $F(A \times A; \{1, ..., \mathrm{card}(A)\} \times F(\mathscr{P}(A); H)$ and thus it is finite.

C. Asynchronous Mapping—Second Construction

Let T be a recognizable subset of $M(A, \Theta)$. In this subsection we present another construction of asynchronous mapping recognizing T, given in Theorem 5.8.

Similarly to the previous one, this new asynchronous mapping recognizing T arises by augmenting the basic mapping ν by information concerning prefixes of traces of T. For this reason the new construction seems to be more natural. Nevertheless the proof is more technical than the preceding one.

We begin with a lemma revealing a property of $M(A, \Theta)$-automata.

LEMMA 5.6. *Assume T is a recognizable trace language and $(M(A, \Theta), Q, \delta, q_0, F)$ is an automaton recognizing T. Let $q \in Q$ be a state, and $u_1, ..., u_n, v_1, ..., v_n$ be traces such that*

(1) $\forall i\ (1 \leqslant i \leqslant n)\ \delta(q, u_i) = \delta(q, v_i)$,

(2) $\forall i, j\ (1 \leqslant i, j \leqslant n)\ j < i \Rightarrow v_j \Theta u_i$,

(3) $\forall i, j\ (1 \leqslant i, j \leqslant n)\ i \neq j \Rightarrow v_i \Theta v_j$.

Then $\delta(q, u_1 \cdots u_n) = \delta(q, v_1 \cdots v_n)$.

Proof. We use induction on n. If n equals one, the result is obvious. Suppose $n > 1$. We compute

$$\begin{aligned}
\delta(q, u_1 \cdots u_n) &= \delta(\delta(q, u_1), u_2 \cdots u_n) \\
&= \delta(\delta(q, v_1), u_2 \cdots u_n) && \text{(by (1))} \\
&= \delta(q, v_1 u_2 \cdots u_n) \\
&= \delta(q, u_2 \cdots u_n v_1) && \text{(by (2))} \\
&= \delta(\delta(q, u_2 \cdots u_n), v_1) \\
&= \delta(\delta(q, v_2 \cdots v_n), v_1) && \text{(induction hypothesis)} \\
&= \delta(q, v_2 \cdots v_n v_1) \\
&= \delta(q, v_1 \cdots v_n) && \text{(by (3)).} \quad \blacksquare
\end{aligned}$$

LEMMA 5.7. *Let t be a trace, α, β, γ be subsets of A, $\tau = E(\partial_\alpha(t), \partial_\beta(t))$ and let*

(1) $\xi = \bar{\Theta}(\mathrm{alph}(\partial_\gamma(\nabla_{\bar{\tau}}(\partial_\beta(t)))))$, (2) $u_0 = \partial_\xi(\partial_\alpha(t))$,

(3) $u_1 = \nabla_{\bar{\xi}}(\partial_{\gamma \cup \xi}(\partial_\alpha(t)))$, (4) $u_2 = \partial_\gamma(\nabla_{\bar{\tau}}(\partial_\beta(t)))$.

Then:

(i) $\partial_\gamma(\partial_{\alpha \cup \beta}(t)) = u_0 u_1 u_2$, (ii) $\partial_{\gamma \cup \xi}(\partial_\alpha(t)) = u_0 u_1$,

(iii) $\partial_{\gamma \cap \xi}(\partial_\beta(t)) = u_0 u_2$, (iv) $u_1 \Theta u_2$.

Proof. Let t_0, t_1, t_2 be such that $\partial_\alpha(t) = t_0 t_1$, $\partial_\beta(t) = t_0 t_2$, and $t_1 \Theta t_2$. From Proposition 3.2 we get $t_0 = \partial_\tau(\partial_\alpha(t)) = \partial_\tau(\partial_\beta(t))$, and hence

$$t_2 = \nabla_{\bar{\tau}}(\partial_\beta(t)). \tag{5}$$

Note that by (1) and (4) $\xi = \bar{\Theta}(\mathrm{alph}(u_2))$. On the other hand

$$\mathrm{alph}(u_1) = \mathrm{alph}(\nabla_{\bar{\xi}}(\partial_{\gamma \cup \xi}(\partial_\alpha(t)))) \subseteq \bar{\xi};$$

thus

$$\mathrm{alph}(u_1) \times \mathrm{alph}(u_2) \subseteq \Theta, \text{ i.e., (iv) holds.}$$

Since by Proposition 2.4 (2) $\partial_\xi(\partial_\alpha(t)) = \partial_\xi(\partial_{\xi \cup \gamma}(\partial_\alpha(t)))$, we obtain

$$\begin{aligned} u_0 u_1 &= \partial_\xi(\partial_\alpha(t))\, \nabla_{\bar{\xi}}(\partial_{\gamma \cup \xi}(\partial_\alpha(t))) \\ &= \partial_\xi(\partial_{\xi \cup \gamma}(\partial_\alpha(t)))\, \nabla_{\bar{\xi}}(\partial_{\gamma \cup \xi}(\partial_\alpha(t))) = \partial_{\xi \cup \gamma}(\partial_\alpha(t)); \end{aligned}$$

i.e., (ii) holds.

Using (ii), (5), and (1) we get

$$u_0 u_1 u_2 = \partial_{\gamma \cup \xi}(\partial_\alpha(t))\, \partial_\gamma(t_2) = \partial_\gamma(\partial_\alpha(t)\, t_2) = \partial_\gamma(\partial_{\alpha \cup \beta}(t));$$

i.e., (i) holds.

By (5) and (1) we have $\xi = \bar{\Theta}(\mathrm{alph}(\partial_\gamma(t_2))) \subseteq \bar{\Theta}(\mathrm{alph}(t_2))$, and since $t_1 \Theta t_2$, this implies $\xi \cap \mathrm{alph}(t_1) = \varnothing$. Therefore

$$u_0 = \partial_\xi(\partial_\alpha(t)) = \partial_\xi(t_0 t_1) = \partial_\xi(t_0). \tag{6}$$

We have $\mathrm{alph}(u_2) \subseteq \xi$, which implies $\mathrm{Max}(u_2) \subseteq \xi$. On the other hand, since $u_2 = \partial_\gamma(t_2)$ we have $\mathrm{Max}(u_2) \subseteq \gamma$. Thus $\mathrm{Max}(u_2) \subseteq \xi \cap \gamma$ and Propositions 2.8 and 2.4 (2) yield

$$u_2 = \partial_{\gamma \cap \xi}(u_2) = \partial_{\gamma \cap \xi}(\partial_\gamma(t_2)) = \partial_{\gamma \cap \xi}(t_2). \tag{7}$$

Taking (6) and (7) into account, we have

$$\partial_{\gamma\cap\xi}(\partial_\beta(t))=\partial_{\gamma\cap\xi}(t_0t_2)=\partial_{(\gamma\cap\xi)\cup\xi}(t_0)\,\partial_{\gamma\cap\xi}(t_2)$$
$$=\partial_\xi(t_0)\,\partial_{\gamma\cap\xi}(t_2)=u_0u_2;$$

i.e., we obtain (iii). ∎

THEOREM 5.8. *Let T be a recognizable subset of $M(A,\Theta)$ and let $(M(A,\Theta), Q, \delta, q_0, F)$ be a finite automaton recognizing T. For any trace t of $M(A,\Theta)$, by δ^*_t we denote the function from $\mathscr{P}(A)$ into Q that maps every subset γ of A to $\delta(q_0, \partial_\gamma(t))$.*

Then the mapping ψ that maps any trace t of $M(A,\Theta)$ to $\psi(t)=(v_t, \delta^*_t)$ is an asynchronous mapping recognizing T.

Proof. First, we prove that the mapping ψ is uniform. Let α and β be subsets of A. Let t and r be traces such that

$$(v_{\partial_\alpha(t)}, \delta^*_{\partial_\alpha(t)})=(v_{\partial_\alpha(r)}, \delta^*_{\delta_\alpha(r)}), \tag{1}$$

$$(v_{\partial_\beta(t)}, \delta^*_{\partial_\beta(t)})=(v_{\partial_\beta(r)}, \delta^*_{\partial_\beta(r)}). \tag{2}$$

Since v is uniform, to accomplish the proof it suffices to show that

$$\delta^*_{\partial_{\alpha\cup\beta}(t)}=\delta^*_{\partial_{\alpha\cup\beta}(r)}. \tag{3}$$

From (1) and (2) we deduce by Proposition 4.4:

$$\tau=E(\partial_\alpha(t), \partial_\beta(t))=E(\partial_\alpha(r), \partial_\beta(r)). \tag{4}$$

Let γ be a subset of A. Now $v_{\partial_\beta(t)}=v_{\partial_\beta(r)}$ implies by Lemma 5.4 that

$$\mathrm{alph}(\partial_\gamma(\nabla_{\bar\tau}(\partial_\beta(t))))=\mathrm{alph}(\partial_\gamma(\nabla_{\bar\tau}(\partial_\beta(r)))),$$

which yields in turn

$$\xi=\bar\Theta(\mathrm{alph}(\partial_\gamma(\nabla_{\bar\tau}(\partial_\beta(t)))))=\bar\Theta(\mathrm{alph}(\partial_\gamma(\nabla_{\bar\tau}(\partial_\beta(r))))). \tag{5}$$

Let

$$u_0=\partial_\xi(\partial_\alpha(t)), \qquad v_0=\partial_\xi(\partial_\alpha(r)),$$
$$u_1=\nabla_{\bar\xi}(\partial_{\gamma\cup\xi}(\partial_\alpha(t))), \qquad v_1=\nabla_{\bar\xi}(\partial_{\gamma\cup\xi}(\partial_\alpha(r))),$$
$$u_2=\partial_\gamma(\nabla_{\bar\tau}(\partial_\beta(t))), \qquad v_2=\partial_\gamma(\nabla_{\bar\tau}(\partial_\beta(r))).$$

By (1), (2), (5) and by Lemma 5.7 we get

$$\begin{aligned}\delta(q_0, u_0) &= \delta(q_0, \partial_\xi(\partial_\alpha(t))) = \delta^*_{\partial_\alpha(t)}(\xi) \\ &= \delta^*_{\partial_\alpha(r)}(\xi) = \delta(q_0, \partial_\xi(\partial_\alpha(r))) = \delta(q_0, v_0),\end{aligned} \tag{6}$$

$$\begin{aligned}\delta(q_0, u_0u_1) &= \delta(q_0, \partial_{\gamma\cup\xi}(\partial_\alpha(t))) = \delta^*_{\partial_\alpha(t)}(\gamma\cup\xi) \\ &= \delta^*_{\partial_\alpha(r)}(\gamma\cup\xi) = \delta(q_0, v_0v_1),\end{aligned} \tag{7}$$

$$\begin{aligned}\delta(q_0, u_0u_2) &= \delta(q_0, \partial_{\gamma\cap\xi}(\partial_\beta(t))) = \delta^*_{\partial_\beta(t)}(\gamma\cap\xi) \\ &= \delta^*_{\partial_\beta(r)}(\gamma\cap\xi) = \delta(q_0, v_0v_2),\end{aligned} \tag{8}$$

$$u_1 \Theta u_2, \qquad v_1 \Theta v_2. \tag{9}$$

Now note that since $v_{\partial_\beta(t)} = v_{\partial_\beta(r)}$, by Lemma 5.4 we get

$$\text{alph}(u_2) = \text{alph}(\partial_\gamma(\nabla_{\bar{\tau}}(\partial_\beta(t)))) = \text{alph}(\partial_\gamma(\nabla_{\bar{\tau}}(\partial_\beta(r)))) = \text{alph}(v_2).$$

This fact and (9) imply directly that

$$u_1 \Theta v_2, \qquad u_2 \Theta v_1. \tag{10}$$

Thus we get by Lemma 5.6 applied to $q = \delta(q_0, u_0) = \delta(q_0, v_0)$

$$\begin{aligned}\delta^*_{\partial_{\alpha\cup\beta}(t)}(\gamma) &= \delta(q_0, \partial_\gamma(\partial_{\alpha\cup\beta}(t))) = \delta(q_0, u_0u_1u_2) \\ &= \delta(q_0, v_0v_1v_2) = \delta^*_{\partial_{\alpha\cup\beta}(r)}(\gamma);\end{aligned}$$

i.e., (3) holds.

Now, we prove that ψ is locally right regular. Since v is locally right regular, it suffices to show that δ^* is locally right regular either.

Let a be a letter. Let t be a trace in $Pr(A, \Theta)\setminus\{\varepsilon\}$ such that $\partial_a(t) = t = ra$. We shall show that δ^*_r and a determine δ^*_t. Let γ be a subset of A. There are two cases to examine.

Case 1. $a \notin \gamma$. Then $\partial_\gamma(t) = \partial_\gamma(ra) = \partial_\gamma(r)$, which implies directly that $\delta^*_t(\gamma) = \delta^*_r(\gamma)$.

Case 2. $a \in \gamma$. Then $\partial_\gamma(t) = \partial_\gamma(ra) = \partial_{\gamma\cup\bar{\Theta}(a)}(r)\,\partial_\gamma(a) = \partial_{\gamma\cup\bar{\Theta}(a)}(r)\,a$. Thus

$$\delta^*_t(\gamma) = \delta(q_0, \partial_\gamma(t)) = \delta(\delta(q_0, \partial_{\gamma\cup\bar{\Theta}(a)}(r)), a) = \delta(\delta^*_r(\gamma\cup\bar{\Theta}(a)), a). \quad \blacksquare$$

Thus we have another construction of an asynchronous mapping recognizing a given recognizable trace languages T. This gives, of course, another proof of the implication (1) ⇒ (2) of the main Theorem 4.1. Note only that the codomain of the asynchronous mapping ψ constructed above is equal to

$$F(A \times A; \{1, ..., \mathrm{card}(A)\}) \times F(\mathscr{P}(A), Q)$$

and is finite.

6. Bounded Time-Stamps in a Distributed System

Suppose that in a distributed system some agents communicate by means of messages. Usually, to execute the prescribed protocol correctly the agents should have some knowledge about the relative order of messages. To this end, they add to every message a tag, called a time-stamp, enabling them to find out the necessary information about the ordering of messages. The importance of an appropriate stamping algorithm was for the first time emphasized by Lamport (1978), to which we refer the reader for further discussion. In most cases, it is a relatively easy task to construct an appropriate stamping system if no bounds on the size of stamps are imposed. But if we allow only a finite set of time-stamps then the construction of an appropriate stamping system becomes difficult or sometimes even impossible; see, for example, Li and Vitanyi (1989) for such a construction. In this section we show how to use the result of Section 4 to construct a special finite time-stamp system.

The distributed system considered here consists of a finite set A of agents and a finite set B of boxes. The agents communicate by messages that they leave in some boxes. Every agent $a \in A$ has access only to a subset $\mathrm{Dom}(a) \subseteq B$ of boxes. Conversely, for every box $i \in B$, $A_i = \{a \in A \mid i \in \mathrm{Dom}(a)\}$ is the set of agenst which have access to i. If $i \in \mathrm{Dom}(a)$ then we say that the box i and the agent a are adjacent.

By B_i, for $i \in B$, we denote the contents of the box i, i.e., the set of messages that i contains. We assume that at the beginning all boxes are empty: for all $i \in B$, $B_i = \varnothing$.

Every message is a triple (m, a, d), where m is the contents of the message taken from some set M of possible contents; $a \in A$ identifies the sender of the message; and finally d is a time-stamp from some set Stamps of time-stamps. Thus formally the cartesian product $U = M \times A \times \mathrm{Stamps}$ is the set of all messages. In the following, by contents, sender and stamp we denote the projection of U onto M, A, and Stamps, respectively. Furthermore, we assume that any box contains for any $a \in A$ at most one message sent by a.

During their moves the agents not only send new messages but also will retransmit messages sent by other agents. For this reason, besides messages left by agents adjacent to i, every box $i \in B$ can contain messages sent by other agents and retransmitted by agents from A_i.

A single move of each agent $a \in A$ consists of four phases. During the first phase a reads the contents of all adjacent boxes, emptying them in this way. Let R be the set of messages that were read in this phase.

In the second phase, for every $b \in A \setminus \{a\}$, if R contains messages sent by b then a selects the last of them; denote it by u_b.

In the third phase, a chooses $m \in M$ that it wishes to send and computes a time-stamp $d \in$ Stamps. Let $u_a = (m, a, d)$.

Finally, in the last phase a transmits to all adjacent boxes all messages from the set $\{u_c \mid c \in A\}$.

This entire move (consisting of reading, selecting, constructing a new message, and sending) is considered atomic. This implies that the access to every box is sequential, and moreover, at a given moment, an agent a has access either to all its adjacent boxes or to none of them. Note that immediately after the move all boxes adjacent to a have the same contents: for every agent $b \in A$, they contain at most one message issued by b, namely the last message sent by b and known to a. We assume that for every message $u \in U$ the field contents(u) does not provide any information concerning the relative order of messages. Thus during the second phase of every move, agent a can use only the fields sender(u) and stamp(u) of $u \in R$ to find out for every $b \in A \setminus \{a\}$ the last message in R sent by b.

To implement this system we should specify the set Stamps, the algorithm selecting messages in the second phase of each move, and the algorithm assigning a stamp to the new message created in the third phase.

A simple implementation exists if we allow the set Stamps to be infinite. Let Stamps $= \mathbb{N}_+$, and assume that every agent is equipped with a counter initially set to 0. Then during its move, agent a increases its counter by 1 and takes the obtained value as the time-stamp d for its new message in the third phase, $u_a = (m, a, d)$. The selection procedure in the second phase of the move is trivial in this implementation. For every $b \in A \setminus \{a\}$, a takes all messages in R with the sender field equal to b and selects among them the one with the greatest stamp field.

The aim of this section is to present another implementation with a bounded number of time-stamps. First we define some auxiliary notions. By a serial event we mean any finite sequence of elements of the set $M \times A$, $SE = (M \times A)^*$. Any occurrence of $(m, a) \in M \times A$ in a serial event $x \in SE$ represents a move performed by the agent a such that m is the contents of the new message sent by a during this move. Let us suppose that α is an algorithm implementing the system. For every $x \in SE$ and $i \in$ Box by $B_i^\alpha(x)$ we denote the contents of the box i after the execution of the serial event x in the implementation α; $B_i^\alpha(x)$ can be defined in the following inductive manner:

(i) for all $i \in \text{Box}$, $B_i^\alpha(\varepsilon) = \emptyset$,

(ii) if $y = x(m, a) \in SE$ then

(1) for all $i \in B \backslash \text{Dom}(a)$, $B_i^\alpha(y) = B_i^\alpha(x)$,

(2) to obtain the new contents of all boxes adjacent to a apply the algorithm α to $B_i = B_i^\alpha(x)$ with $i \in \text{Dom}(a)$.

EXAMPLE. Let $A = \{a, b, c, d\}$ and $B = \{B_{ab}, B_{bc}, B_{cd}, B_{da}\}$, where B_{xy} with $x, y \in A$, denotes a box adjacent to x and y.

Let $S = (m_1, a)(m_2, b)(m_3, a)(m_4, c)(m_5, d)$ be a serial event and let s_i denote the prefix of s of length i, $i = 0, ..., 5$.

The contents of the boxes after the execution of S_i in the counter implementation that was considered previously:

After S_0:

$$B_{ab} = \emptyset,\ B_{bc} = \emptyset,\ B_{cd} = \emptyset,\ B_{da} = \emptyset;$$

After S_1:

$$B_{ab} = \{(m_1, a, 1)\},\ B_{bc} = \emptyset,\ B_{cd} = \emptyset,\ B_{da} = \{(m_1, a, 1)\};$$

After S_2:

$$B_{ab} = \{(m_1, a, 1), (m_2, b, 1)\},\ B_{bc} = \{(m_1, a, 1), (m_2, b, 1)\}$$
$$B_{cd} = \emptyset,\ B_{da} = \{(m_1, a, 1)\};$$

After S_3:

$$B_{ab} = \{(m_3, a, 2), (m_2, b, 1)\},\ B_{bc} = \{(m_1, a, 1), (m_2, b, 1)\},$$
$$B_{cd} = \emptyset,\ B_{da} = \{(m_3, a, 2), (m_2, b, 1)\};$$

After S_4:

$$B_{ab} = \{(m_3, a, 2), (m_2, b, 1)\},\ B_{bc} = \{(m_1, a, 1), (m_2, b, 1), (m_4, c, 1)\},$$
$$B_{cd} = \{(m_1, a, 1), (m_2, b, 1), (m_4, c, 1)\},\ B_{da} = \{(m_3, a, 2), (m_2, b, 1)\};$$

After S_5:

$$B_{ab} = \{(m_3, a, 2), (m_2, b, 1)\},\ B_{bc} = \{(m_1, a, 1), (m_2, b, 1), (m_4, c, 1)\},$$
$$B_{cd} = \{(m_3, a, 2), (m_2, b, 1), (m_4, c, 1), (m_5, d, 1)\},$$
$$B_{da} = \{(m_3, a, 2), (m_2, b, 1), (m_4, c, 1), (m_5, d, 1)\}.$$

Let Θ be the independency relation over A defined in the following way:

$$\Theta = \{(a, b) \in A^2 \mid \mathrm{Dom}(a) \cap \mathrm{Dom}(b) = \emptyset\}.$$

Fact 6.1. For every $a \in A$, $\bar{\Theta}(a) = \bigcup_{i \in \mathrm{Dom}(a)} A_i$.

We have also a natural independency relation Υ over $M \times A$:

$$\Upsilon = \{((m, a), (m', b)) \in (M \times A)^2 \mid a\Theta b\}.$$

Let $H = SE/\sim_\Upsilon$ be the partially commutative monoid obtained as the quotient of SE by the relation $\sim_\Upsilon$ induced by Υ. Elements of H are called histories. In the following, we assume that *sender* also denotes the projection from $M \times A$ into A. It can be extended to a homomorphism from SE into A^*. Note that since

$$\forall x, y \in SE,\ x \sim_\Upsilon y \Rightarrow \mathrm{sender}(x) \sim_\Theta \mathrm{sender}(y)$$

we can extend the mapping sender to a homomorphism from H onto $M(A, \Theta)$ by

$$\forall x \in SE,\ \mathrm{sender}\ ([x]_\Upsilon) = [\mathrm{sender}(x)]_\Theta.$$

The following fact shows that two serial events equivalent under $\sim_\Upsilon$ are indistinguishable in the system; i.e., independent agents can perform their moves simultaneously.

Fact 6.2. Let $x, y \in SE$ be such that $x \sim_\Upsilon y$. Then for every algorithm α implementing the system and for each box $i \in B$, $B_i^\alpha(x) = B_i^\alpha(y)$.

Thus we can set $B_i^\alpha(h) = B_i^\alpha(x)$ for $x \in SE$ and $h = [x]_\Upsilon \in H$.

Fact 6.3. *Let* $h \in H$, $t = \mathrm{sender}(h) \in M(A, \Theta)$. Then for every algorithm α implementing the system and for all $b \in A$ and $i \in B$:

(1) $\partial_b(\partial_{A_i}(t)) = \varepsilon$ if and only if $B_i^\alpha(h)$ does not contain messages sent by b,

(2) if $|\partial_b(\partial_{A_i}(t))|_b = |\partial_{A_i}(t)|_b = k > 0$ and (m, b) is the kth message of b in the history h then $B_i^\alpha(h)$ contains a message u such that $\mathrm{contents}(u) = m$ and $\mathrm{sender}(u) = b$.

Now we give an implementation γ of our system using a finite set Stamps.

We set $\mathrm{Stamps} = F(A; \{1, ..., n\})$, where $n = \mathrm{Card}(A)$. Let tag be the mapping from $Pr(A, \Theta)$ into Stamps defined in the following way,

$$\forall t \in Pr(A, \Theta),\ \forall a \in A \quad \mathrm{tag}(t)(a) = \lambda(\partial_a(t)),$$

where λ is the function defined in Section 4.

Now the idea of the implementation γ is to assign to messages elements of Stamps in such a way that the following condition is satisfied.

Condition Γ. $\forall h \in H$, $\forall b \in A$, $\forall i \in B$, $\forall t \in M(A, \Theta)$, $\forall k \in \mathbb{N}_+$, if

(1) $t = \text{sender}(h) \in M(A, \Theta)$,

(2) $|\partial_b(\partial_{A_i}(t))|_b = |\partial_{A_i}(t)|_b = k > 0$,

(3) (m, b) is the k-th message of b in h,

then

$$(m, b, \text{tag}(\partial_b(\partial_{A_i}(t)))) \in B_i^\gamma(h).$$

If Condition Γ holds, $h \in H$, and $t = \text{sender}(h)$ then for each $i \in B$ the time-stamps of messages in the box i determine the mapping $v_{\partial_{A_i}(t)}$.

Fact 6.4. Suppose that an implementation γ satisfies Γ. Let $h \in H$, $t = \text{sender}(h) \in M(A, \Theta)$. Then

$$v_{\partial_{A_i}(t)}(c, b) = \begin{cases} n & \text{if } B_i^\gamma(h) \text{ does not contain} \\ & \text{messages sent by } b, \\ f(c) & \text{if } \quad (m, b, f) \in B_i^\gamma(h). \end{cases}$$

Proof. By Fact 6.3 if $B_i^\gamma(h)$ does not contain messages sent by b then

$$\partial_b(\partial_{A_i}(t)) = \varepsilon$$

and

$$v_{\partial_{A_i}(t)}(c, b) = \lambda(\partial_c(\partial_b(\partial_{A_i}(t)))) = \lambda(\varepsilon) = n.$$

Otherwise

$$v_{\partial_{A_i}(t)}(c, b) = \lambda(\partial_c(\partial_b(\partial_{A_i}(t)))) = \text{tag}(\partial_b(\partial_{A_i}(t)))(c) = f(c). \quad \blacksquare$$

Now we can present the details of the algorithm γ. Let $a \in A$ and let $h \in H$ be the history executed up to now. Let $t = \text{sender}(h) \in M(A, \Theta)$. By Fact 6.4, inspecting the contents of box i, a can calculate $v_{\partial_{A_i}(t)}$ for all $i \in \text{Dom}(a)$. Now by the uniformity of v and Fact 6.1, a obtains $v_{\partial_{\Theta(a)}(t)}$. During the selection phase, a chooses for every $b \in A \setminus \{a\}$ from the reading set R the message $u_b = (m, b, f)$ such that

$$\forall c \in A \qquad f(c) = v_{\partial_{\Theta(a)}(t)}(c, b).$$

By local right regularity of v, a can now get $v_{\partial_a(ta)}$. Let g be the mapping from A into $\{1, ..., n\}$ such that

$$\forall c \in A, \qquad g(c) = v_{\partial_a(ta)}(c, a).$$

Then g is the time-stamp for the new message that a creates during the third phase, $u_a = (m, a, g)$. To show the correctness of γ it suffices to obserbe that if Condition Γ holds for a history h then it holds for the history $h(m, a)$ for every $(m, a) \in M \times A$.

RECEIVED May 25, 1990; FINAL MANUSCRIPT RECEIVED July 11, 1991

REFERENCES

AALBERSBERG, I. J., AND ROZENBERG, G. (1986), Theory of traces, *Theoret. Comput. Sci.* **60**, 1–83.

BERTONI, A., MAURI, G., AND SABADINI, N. (1989), Membership problems for regular and context-free trace languages, *Inform. and Comput.* **82**, 135–150.

BROOKES, S. D., HOARE, C. A. R., AND ROSCOE, A. W. (1984), A theory of communicating sequential processes, *J. Assoc. Comput. Mach.* **31**, 560–599.

BRUSCHI, D., PIGHIZZINI, G., AND SABADINI, N. (1988), On the existence of the minimum asynchronous automaton and on decision problems for unambiguous trace languages, *Lecture Notes Comput. Sci.* **294**, 334–346.

CARTIER, P., AND FOATA, D. (1969), "Problèmes combinatoires de commutations et réarrangements," Lecture Notes in Mathematics, Vol. 85, Springer-Verlag, Berlin/New York, 1969.

CHOFFRUT, C. (1986), "Free Partially Commutative Monoid," LITP Report 20.

CONWAY, L., AND MEAD, C. (1980), "Introduction to VLSI Systems," Addison–Wesley, Reading MA.

CORI, R., LATTEUX, M., ROOS, Y., AND SOPENA, E. (1988), 2-asynchronous automata, *Theoret. Comput. Sci.* **60**, 93–102.

CORI, R., AND MÉTIVIER, Y. (1988), Approximation of a trace, asynchronous automata and the ordering of events in a distributed system, *Lecture Notes Comput. Sci.* **317**, 147–161.

CORI, R., AND PERRIN, D. (1985), Automates et commutations partielles, *RAIRO Inform. Theor. Appl.* **19**, 21–32.

DIEKERT, V. (1990), "Combinatorics on Traces," Lecture Notes in Computer Science, Vol. 454, Springer-Verlag, Berlin/New York.

EILENBERG, S. (1974), "Automata, Languages and Machines," Academic Press, New York.

FLÉ, M. P., AND ROUCAIROL, G. (1985), Maximal serializability of iterated transactions, *Theoret. Comput. Sci.* **46**, 1–16.

GASTIN, P., AND ROZOY, B. (1991), "The Poset of Infinitary Traces," LITP Report 91.07.

KARP, R. M., AND MILLER, R. E. (1969), Parallel program schemata, *J. Comput. System Sci.* **3**, 147–195.

LAMPORT, L. (1978), Time, clocks, and the ordering of events in a distributed system, *Comm. ACM* **21**, 558–565.

LI, M., AND VITANYI, P. M. B. (1989), How to share concurrent asynchronous wait-free variables, *Lecture Notes Comput. Sci.* **372**, 488–505.

LOTHAIRE, M. (1983), "Combinatorics on Words," Cambridge Univ. Press, London/New York.

MAZURKIEWICZ, A. (1977), "Concurrent Program Schemes and Their Interpretations," DAIMI-Report, Aarhus University.

MAZURKIEWICZ, A. (1984), Traces, histories, graphs: Instances of a process monoid, *Lecture Notes Comput. Sci.* **176**, 115–133.

MÉTIVIER, Y. (1986), On recognizable subsets of free partially commutative monoids, *Lecture Notes Comput. Sci.* **226**, 254–264.

MÉTIVIER, Y. (1987), An algorithm for computing asynchronous automata in the case of acyclic non-commutation graphs, *Lecture Notes Comput. Sci.* **267**, 226–236.

MILNER, R. (1983), Calculi for synchrony and asynchrony, *Theoret. Comput. Sci.* **25**, 267–310.

OCHMAŃSKI, E. (1985), Regular behaviour of concurrent systems, *EATCS Bull.* **27**, 56–67.

PERRIN, D. (1989), Partial commutations, *Lecture Notes Comput. Sci.* **372**, 637–651.

REISIG, W. (1982), "Petri Nets. An Introduction," Springer-Verlag, Berlin/New York.

SAKAROVITCH, J. (1987), On regular trace languages, *Theoret. Comput. Sci.* **52**, 59–75.

THOMAS, W. (1990), On logical definability of trace languages, *in* "Proc., ASMICS workshop Free partially commutative monoids" (V. Diekert, Ed.), Report TUM-I9002, München University.

VIENNOT, X. G. (1987), Heaps of pieces. I. Basic definitions and combinatorial lemmas, *Lecture Notes Math.* **123**, 321–350.

VON NEUMANN, J. (1966), "Theory of self-reproducing automata" (A. Burks, Ed.), Univ. of Illinois Press, Urbana, IL.

ZIELONKA, W. (1987), Notes on finite asynchronous automata, *RAIRO Inform. Theor. Appl.* **21**, 99–135.

ZIELONKA, W. (1989), Safe executions of recognizable trace languages, *Lecture Notes Comput. Sci.* **363**, 278–289.