THE POSET OF INFINITARY TRACES[*]

Paul GASTIN
LITP
Université Paris 6
4, place Jussieu
75 252 Paris Cedex 05
FRANCE

Brigitte ROZOY
LRI, URA CNRS 410
Université Paris 11
Bât. 490
91 405 Orsay Cedex
FRANCE

Abstract: *Partially Commutative Monoids,* also called *Trace Monoids,* are among the most studied formalisms which describe the behaviour of distributed systems. In order to modelize systems which never stop, we have to consider an extension of traces, namely *Infinite traces*. Finite trace monoids are strongly related to *Partial Order Sets (PoSets), Domains and Event Structures,* which are other models to describe the behaviour of distributed systems. The aim of this paper is to establish similar connexions between infinite trace monoids, PoSets and event structures. We prove that the set of finite and infinite traces with the prefix order is a *Scott-domain* and a *coherently complete prime algebraic PoSet*. Moreover, we establish a representation theorem between the class of finite and infinite trace PoSets and a subclass of labelled prime event structures.

Résumé: Les *monoïdes partiellement commutatifs* ou *monoides de traces* sont parmi les formalismes les plus étudiés pour décrire les comportements des systèmes distribués. Afin de modéliser des systèmes qui ne s'arrètent pas, on doit considérer une extension des traces: les *traces infinies*. Les monoïdes de traces finies sont fortement liés aux *ensembles partiellement ordonnés (PoSets)*, aux *domaines* et aux *structures d'événements*, qui sont d'autres modèles décrivant les comportements des systèmes distribués. Le but de ce papier est d'établir des liens similaires entre les monoides de traces infinies, les PoSets et les structures d'événements. On prouve que l'ensemble des traces finies ou infinies muni de l'ordre préfixe est un *domaine de Scott* et un ensemble *premier algébrique cohéremment complet*. De plus, on établit un théorème de représentation entre la classe des ensembles de traces finies et infinies et une sous-classe des structures d'événements premières et étiquetées.

---

[*] The work presented in this paper has been partially carried out within the Esprit Basic Research Action “Demon” and the P.R.C. C^3, CNRS.

1. Introduction

Free partially commutative monoids were first introduced by Cartier and Foata in order to study problems of rearrangements [C.F]. In computer science, they were used for the first time by Mazurkiewicz [M1], under the name of traces, in order to describe behaviours of distributed systems. In such systems, actions executed by different components may be completely independent. Thus, considering a distributed computation, different external observers may perceive independent actions in a different order. Therefore, a distributed computation can be modelized by the collection of all its (sequential) observations, called a trace, which is precisely an equivalence class in some partially commutative monoid. Due to their natural modelization of distributed systems, trace monoids were extensively studied in the last years. See, for instance, the surveys [A.R], [D1], [M2] or [P2], where further references can also be found.

A theory of infinite traces has recently begun in order to modelize behaviours of distributed systems which never stop, as for instance distributed operating systems. Infinite traces were first introduced in [G1] and more extensively studied in [G2]. Their partial order properties were first studied in [G.R1] and [G.R2] where relationships with event structures were also established. Independently, infinite traces were introduced in [K1] as the natural order completion of the Partial Order Set (PoSet) of finite traces. The importance of the infinite trace theory is shown by all the researches that have already been started on this topic, despite of its recent introduction [B.M.P], [D2], [G3], [G.P.Z], [K2]…

On the other hand, PoSets have often been used for the description of the behaviours of distributed systems. In that approach, the description does not insist on the possible observations of a behaviour, but, on the occurrences of its actions and on their causal relations. The first attempt of such an “event based” model has probably been performed by L. Lamport [La], and intensively studied since then [B.C1], [Dr], [G1], [R1], [Sh]. Finally, event structure models arise naturally in the theory of Petri nets and Domains [N.P.W]. They became a substantial theory, mainly through the efforts of G.Winskel [W1], [W2], [W3]. An event structure is basically a partially ordered set of event occurrences together with a binary conflict relation. As before, the partial ordering is meant to capture causal dependences, whereas the conflict relation modelizes choice, so that two events that are in conflict cannot both occur in a single behaviour. Since its introduction by Winskel, the event structure theory has received a good deal of attention [B.C2], [C.F.M], [R2], [R.T].

The connexions between finite traces and event structures have been observed in [R1], [R.T], [Sh], [St] where related PoSet properties have been stated.

The aim of this paper is to study properties of the PoSet of infinite traces and relationships between infinite traces and event structures.

In next section, we recall definitions on finite and infinite traces and results needed in the sequel. Section 3 is devoted to the study of the PoSet properties of traces. We prove that the PoSet of finite and infinite traces is coherently complete, is a Scott-Domain and is Prime Algebraic. In section 4, we establish a representation theorem between the class of infinite trace PoSets and a subclass of labelled prime event structures.

2. Infinite traces.

In this section, we provide the definitions and the notations on traces. If more details are needed, we refer the reader to [B.N], [E], [H.R], [P1] or [P.P] for the theory of infinite words, to [A.R], [D1], [M2] or [P2] for recent surveys about finite traces and to [G2] as regards infinite traces. In this paper we are mainly concerned in the partial order properties of the set of traces. Therefore, we will use the dependence graph approach of traces which is the most suitable for this purpose. Two other well known points of view about traces are the equivalence classes of words and the projections on the dependence cliques. The generalization of these approaches to infinite traces can be found in [G2].

We consider a finite alphabet A and as usual A^* denotes the free monoid over A, that is, the set of finite words over A. An infinite word u is simply a countable infinite sequence of events of A. We denote A^ω the set of infinite words over A and $A^\infty = A^* \cup A^\omega$ the infinitary free monoid with the usual concatenation. As for finite words, a prefix order may be derived from this concatenation: $u < v$ iff there exists $w \in A^\infty$ such that $v = u.w$. For u in A^∞, alph(u) denotes the set of letters of u, $|u|$ denotes the length of u and $|u|_a$ denotes the number of occurences of the letter a in the word u.

We consider a binary irreflexive and symmetric relation I over the alphabet A called the independence relation. The letters of A can be viewed as events in a distributed system and two events are independent iff they are related by I. We denote $D = A \times A \setminus I$ the dependence relation over A. According to the relation I, we define an equivalence relation $\sim_I$ or simply $\sim$ on sequences of letters which is the reflexive and transitive closure of the relation $\{(uabv,ubav) / u,v \in A^*, (a,b) \in I\}$. The relation $\sim$ is indeed a congruence over A^* and the quotient monoid $A^*/\sim$ is denoted by $M(A^*,I)$. The members of $M(A^*,I)$ are called traces.

Dependence graphs, or occurrence graphs, are known as a fundamental representation for finite traces [Du], [A.R]. They are a very good support for the intuition on traces and can be used to present conveniently a lot of properties about traces [A.R]. A dependence relation D over A being fixed, a dependence graph is a labelled acyclic graph or rather an isomorphism class of a labelled acyclic graph. We recall that an isomorphism of labelled graphs is a bijection between the vertices of two graphs which preserves the edges and the labelling.

Definition 2.1. A finite dependence graph is (an isomorphism class of) a labelled acyclic graph (V,E,λ) with V a finite set of vertices, $E \subset V \times V$ a set of edges and $\lambda: V \to A$ a labelling function which satisfies:

(γ_1) $\forall x,y \in V, (\lambda(x),\lambda(y)) \in D \Leftrightarrow (x,y) \in E$ or $(y,x) \in E$

The set of finite dependence graphs will be denoted $\mathcal{F}\mathcal{G}(A,D)$.

For instance, let $A = \{a,b,c\}$ and $I = \{(a,c),(c,a)\}$. A dependence graph is presented in Figure 2.1. Since we are only interested in the isomorphism class of the graph, we only write the labelling of a vertex. Note that the relation E of this graph is not transitive since there is no edge between the first a and the second c.

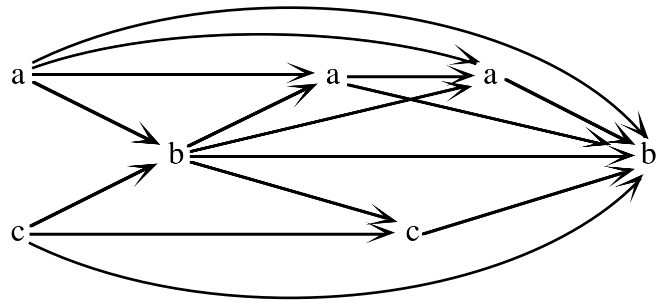

Figure 2.1

In fact, the relevant relation that we will consider in these graphs is the causal relation, that is, the existence of a path from a vertex to another one. This relation, which is the reflexive and transitive closure of E, is denoted E*. Since the graph is acyclic, E* is a partial order on V. With respect to this relation, some edges are not necessary. Thus, in following figures, we will only draw the Hasse diagram of this relation (its minimal representation). The previous graph will thus be simply pictured as in Figure 2.2.

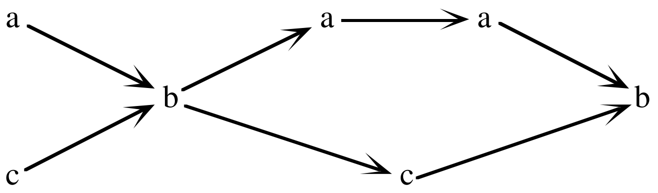

Figure 2.2

We introduce now the dependence graph associated with a word.

Definition 2.2. Let D be a dependence relation and let u be in A^*. The graph $\Gamma(u) = (V,E,\lambda)$ is defined by:

$V = \{ (a,j) / a \in A, 1 \le j \le |u|_a \}$

$E = \{ (a,j) \to (b,k) / (a,b) \in D$ and the j^{th} a occurs before the k^{th} b in u $\}$

$\forall (a,j) \in V, \lambda((a,j)) = a.$

For instance, let A = {a,b,c}, I = {(a,c),(c,a)} and u = acbaacb. The graph associated with u is presented in Figure 2.3. Note that this is a finite dependence graph. In fact and up to an isomorphism, it is the same graph as in Figure 2.3. Note that this is a finite dependence graph.

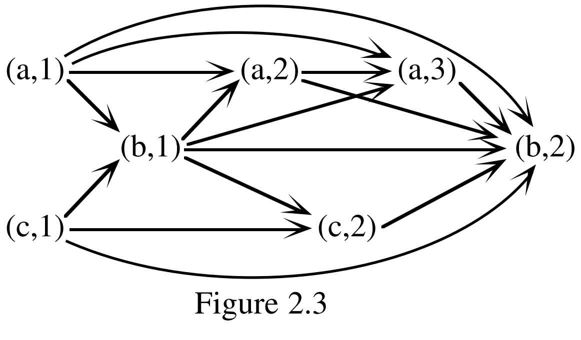

Figure 2.3

The fact that finite dependence graphs are representations of finite traces is well known. Namely, for all (finite) words u and v, we have $u \sim v$ iff $\Gamma(u) = \Gamma(v)$. Moreover, Γ is a surjective mapping from A^* onto $\mathcal{F}\mathcal{G}(A,D)$. Therefore, Γ induces a bijection between the set of traces $M(A^*,I)$ and $\mathcal{F}\mathcal{G}(A,D)$.

We will use now the dependence graph approach to introduce infinite traces. Of course, we will allow infinitely many vertices in a graph. But we are only interested in infinite traces in which all events are reachable. Therefore, we need a restriction which ensures that no vertex depends on infinitely many vertices.

Definition 2.3. A dependence graph is a labelled acyclic graph (V,E,λ) with V a countable set of vertices, $E \subset V \times V$ a set of edges and $\lambda: V \rightarrow A$ a labelling function which satisfies:

(γ_1) $\forall\, x,y \in V,\ (\lambda(x),\lambda(y)) \in D \Leftrightarrow (x,y) \in E$ or $(y,x) \in E$

(γ_2) $\forall\, x \in V,\ \{y \in V,\ (y,x) \in E^*\}$ is finite.

The set of dependence graphs will be denoted by $\mathcal{G}(A,D)$.

For instance, let $A = \{a,b,c\}$ and $I = \{(a,c),(c,a)\}$. The Hasse diagram of an infinite dependence graph is presented in Figure 2.4 but the graph of Figure 2.5 is not a dependence graph since it does not satisfy (γ2).

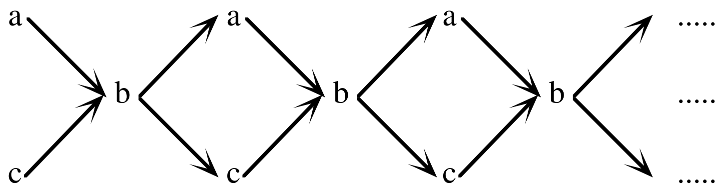

Figure 2.4

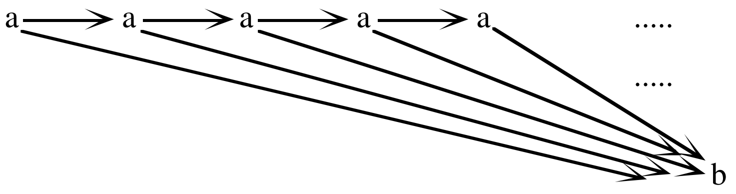

Figure 2.5

Note that in Definition 2.3, the hypothesis that V is countable is not necessary since it can be derived from the other points. Note also that if we consider the natural partial order on the vertices of the graph, namely the existence of a path, this definition implies that every chain between two vertices is finite.

Definition 2.2 extends to infinite words without any modification. Clearly $\Gamma(u)$ is again a dependence graph for any infinite word u. Therefore Γ is a mapping from A^∞ to $\mathcal{G}(A,D)$.

Proposition 2.4. Γ is a surjective mapping from A^∞ onto $\mathcal{G}(A,D)$.

Proof. In order to prove this result, let us first define the depth of a vertex x in a graph G by:

$$\delta(x) = \sup\{\text{length}(p) \text{ where } p \text{ is any path from any vertex } y \text{ to } x\}.$$

For instance, if G is the graph of Figure 2.3, we have $\delta(a,1) = 0$, $\delta(a,2) = 2$, $\delta(b,2) = 4$.

Let $G = (V,E,\lambda)$ be in $\mathcal{G}(A,D)$. Since G is acyclic and satisfies (γ_2), the depth of any vertex of G is finite. Now from (γ_1) we can easily deduce that for all $x,y \in V$, $\delta(x) = \delta(y)$ implies $x = y$ or $(\lambda(x),\lambda(y)) \in I$. Let us now construct an antecedent to G. For any integer k, let w_k be a word formed with the letters of depth k in G. For instance, if G is the graph of Figure 2.1, we get $w_0 = ac$, $w_1 = b$, $w_2 = ac$, $w_3 = a$ and $w_4 = b$. The previous remark shows that each word w_k is composed of independent letters. This explains why we have not fixed any order on the letters in w_k. Let v be the infinite word $w_0w_1w_2\ldots$. We claim that $\Gamma(v) = (V',E',\lambda')$ is isomorphic to G. Define the mapping $g: V \to V'$ by $g(x) = (\lambda(x),k(x))$ where $k(x) = |w_0\ldots w_{\delta(x)}|_{\lambda(x)}$. If $g(x) = g(y)$ then $\lambda(x) = \lambda(y)$ and $\delta(x) = \delta(y)$, whence $x = y$. Therefore g is injective. Now g is also surjective: if $(a,k) \in V'$, there exists n such that the k^{th} a of w occurs in w_n. Hence, there exists $x \in V$ such that $\delta(x) = n$ and $\lambda(x) = a$. We obtain $g(x) = (a,k)$.

It remains to verify that the bijection g preserves the edges and the labelling. For the labelling, it is trivial. For the edges we have:

$$
\begin{aligned}
(x,y) \in E \quad &\Leftrightarrow \quad \delta(x) < \delta(y) \text{ and } (\lambda(x),\lambda(y)) \in D \\
&\Leftrightarrow \quad \text{the } k(x)^{th}\ \lambda(x) \text{ is before the } k(y)^{th}\ \lambda(y) \text{ in } v \text{ and } (\lambda(x),\lambda(y)) \in D \\
&\Leftrightarrow \quad (g(x),g(y)) \in V'
\end{aligned}
$$

This proves the claim and therefore Γ is surjective. ◊

The mapping Γ allows us to define an equivalence relation on A^∞ and using the previous proposition, we obtain two equivalent descriptions of infinite traces.

Definition 2.5. Let $u,v \in A^\infty$, u and v are equivalent, denoted by $u \sim v$, iff $\Gamma(u) = \Gamma(v)$. The set $A^\infty/\sim$ is the set of finite and infinite traces.

Proposition 2.4 implies that Γ is a bijection between $A^\infty/\sim$ and $\mathcal{G}(A,D)$. Therefore, we will identify an infinite trace t in $A^\infty/\sim$ with its dependence graph $\Gamma(t)$ in $\mathcal{G}(A,D)$. As in the finite case, it is also possible to give a direct definition of the equivalence on words. But this equivalence relation is not simply the reflexive and transitive closure of the relation $\{(uabv,ubav) \mathbin{/} u \in A^*, v \in A^\infty \text{ and } (a,b) \in I\}$ (see [G2] for more details).

We now introduce a partial order on traces. For finite traces, the partial order relation is simply the prefix order. We generalize this partial order to infinite traces, using the dependence graph approach. Intuitively, G_1 is a prefix of G_2 if G_1 is a subgraph of G_2 which is closed in the past. We recall that $G_1 = (V_1,E_1,\lambda_1)$ is a subgraph of $G_2 = (V_2,E_2,\lambda_2)$ iff $V_1 \subset V_2$, $E_1 = E_{2|V_1\times V_1}$ and $\lambda_1 = \lambda_{2|V_1}$.

Definition 2.6. Let $G_1 = (V_1,E_1,\lambda_1)$ and $G_2 = (V_2,E_2,\lambda_2)$ be two traces of $\mathcal{G}(A,D)$. $G_1 < G_2$ if and only if G_1 is a subgraph of G_2 such that for all $x \in V_1$ and $y \in V_2$, $(y,x) \in E_2$ implies $y \in V_1$.

For instance, let G_1, G_2, G_3 be the dependence graphs presented in Figure 2.2, 2.4 and 2.6 respectively. We have $G_3 < G_1$ and $G_3 < G_2$.

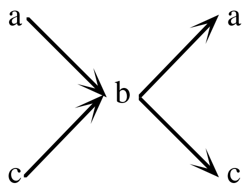

Figure 2.6

One can easily verify that $<$ is a partial order relation on $\mathcal{G}(A,D)$ which extends both the prefix order relations on finite traces and on infinite words. Moreover, Proposition 2.9 will state that this partial order can be derived from the concatenation on traces that we will define. Therefore, the partial order on traces will be called a prefix order. Note that, contrary to the case of infinite words, the prefix order on infinite traces does not reduce to the equality. For instance, if (a,c) is in I, the trace $\Gamma(a^\infty)$ is a prefix of $\Gamma(ca^\infty)$. The properties of the PoSet $(\mathcal{G}(A,D),<)$ will be investigated in next sections. Now we define a concatenation on infinite traces.

Definition 2.7. Let $G_1 = (V_1,E_1,\lambda_1)$ and $G_2 = (V_2,E_2,\lambda_2)$ be two traces of $\mathcal{G}(A,D)$ with $V_1 \cap V_2 = \emptyset$. Their concatenation is defined by:

$$G_1 \otimes G_2 = (V_1 \cup V_2 \,,\, E_1 \cup E_2 \cup \{(x,y) \in V_1 \times V_2 \,/\, (\lambda_1(x) \,,\, \lambda_2(y)) \in D\} \,,\, \lambda_1 \cup \lambda_2)$$

Note that the assumption $V_1 \cap V_2 = \emptyset$ is not restrictive since dependence graphs are defined up to isomorphisms. Clearly, $G_1 \otimes G_2$ is a labelled acyclic graph with a countable set of vertices and satisfies (γ_1). Now it verifies (γ_2) iff $\mathrm{alphinf}(G_1) \times \mathrm{alph}(G_2) \subset I$, where $\mathrm{alph}(G) = \lambda(V)$ and $\mathrm{alphinf}(G) = \{a \in A, |\lambda^{-1}(a)| = \infty\}$. Therefore the concatenation is only partially defined on $\mathcal{G}(A,D)$. We define a new element $\perp$ which means error and will be used to denote that the concatenation is not a dependence graph. We denote $\mathcal{A}$ the set $\mathcal{G}(A,D) \cup \{\perp\}$ (or $A^\infty/\!\sim \cup \{\perp\}$ with the identification of $\mathcal{G}(A,D)$ and $A^\infty/\!\sim$).

Definition 2.8. The concatenation on $\mathcal{A}$ is defined as follows: let $G_1,G_2 \in \mathcal{G}(A,D)$, we define $G_1.G_2 = G_1 \otimes G_2$ if $\mathrm{alphinf}(G_1) \times \mathrm{alph}(G_2) \subset I$ and we set $G_1.G_2 = \perp$ otherwise. Moreover, $\forall\, G \in \mathcal{A}$ we set $G.\perp = \perp.G = \perp$ ($\perp$ is a zero).

One can simply consider $\perp$ as an easy way to denote that the concatenation is not defined but the following remark gives a justification of this notation. Let $\mathcal{G}'$ be the set of dependence graphs which don't necessarily satisfy (γ_2). With the concatenation of Definition 2.7, $\mathcal{G}'$ is clearly a monoid. Now, let $\mathcal{R} = \{G \in \mathcal{G}'$ such that G does not satisfy $(\gamma_2)\}$, it is easy to see that $\mathcal{R}$ is an ideal of $\mathcal{G}'$. Therefore $(\mathcal{A},.)$ is exactly the Rees quotient of $(\mathcal{G}',.)$.
For instance, let $A = \{a,b,c,d\}$, $I = \{(a,c),(c,a),(b,d),(d,b)\}$ and let $r = \Gamma(acbdca^\omega)$, $s = \Gamma(cc)$ and $t = \Gamma(cbc)$. We get $r.s = \Gamma(acbdccca^\omega)$ and $r.t = \perp$. We would like to emphasize on the following fact: if the first trace is finite, then this concatenation coincides with the *natural* one obtained from the quotient of the free monoid. That is, if $r = \Gamma(u)$ is finite and $s = \Gamma(v)$ then $r.s = \Gamma(u.v)$. In fact, we only need this partial case in this paper. Note also that if the first trace $\Gamma(u)$ is infinite and if the concatenation $\Gamma(u).\Gamma(v)$ is not $\perp$, we can express the concatenation with the shuffle. More precisely, we may write $u = u'.u''$ with $u' \in A^*$ and $\mathrm{alph}(u'') = \mathrm{alphinf}(u)$. Then we get $\Gamma(u).\Gamma(v) = \Gamma(u'.(u'' \sqcup\!\sqcup\, v))$.

We will now state some properties of this concatenation. Mainly, $(\mathcal{A},.)$ is a monoid which is nearly cancellative and the concatenation is related to the partial order.

Proposition 2.9.

i) $(\mathcal{A},.)$ is a monoid.

ii) $\forall\ G_1, G_2 \in \mathcal{G}(A,D),\ G_1 < G_2 \Leftrightarrow \exists\ G_3 \in \mathcal{G}(A,D)$ such that $G_2 = G_1.G_3$

iii) $\forall\ G_1, G_2, G_3 \in \mathcal{G}(A,D),\ G_1.G_2 = G_1.G_3 \neq \perp \Rightarrow G_2 = G_3$

iv) $\forall\ G_1 \in \mathcal{F}\mathcal{G}(A,D),\ \forall\ G_2, G_3 \in \mathcal{G}(A,D),\ G_2.G_1 = G_3.G_1 \neq \perp \Rightarrow G_2 = G_3$

Proof. i) The empty graph, denoted ε, is clearly a neutral element. Let $r,s,t \in \mathcal{A}$, the associativity is clear if one of them is equal to $\perp$. Hence we assume that $r,s,t \in \mathcal{G}(A,D)$.

$$\begin{aligned}(r.s).t = \perp \quad &\Leftrightarrow \quad \mathrm{alphinf}(r) \times \mathrm{alph}(s) \not\subset I \text{ or } (\mathrm{alphinf}(r) \cup \mathrm{alphinf}(s)) \times \mathrm{alph}(t) \not\subset I \\ &\Leftrightarrow \quad \mathrm{alphinf}(r) \times (\mathrm{alph}(s) \cup \mathrm{alph}(t)) \not\subset I \text{ or } \mathrm{alphinf}(s) \times \mathrm{alph}(t) \not\subset I \\ &\Leftrightarrow \quad r.(s.t) = \perp\end{aligned}$$

Finally, when $(r.s).t \neq \perp$, the problem reduces to the associativity of the graph concatenation (Definition 2.7) which is obvious.

ii) Let $G_1 = (V_1,E_1,\lambda_1)$ and $G_2 = (V_2,E_2,\lambda_2)$ be two traces of $\mathcal{G}(A,D)$ such that $G_1 < G_2$. Define $G_3 = (V_3,E_3,\lambda_3)$ by $V_3 = V_2 - V_1$, $E_3 = E_{2|V_3 \times V_3}$ and $\lambda_3 = \lambda_{2|V_3}$. It is easy to check that G_3 is a dependence graph which satisfies $\mathrm{alphinf}(G_1) \times \mathrm{alph}(G_3) \subset I$ and $G_2 = G_1.G_3$. The converse is trivial.

iii) Let G_1, G_2, G_3 be three traces of $\mathcal{G}(A,D)$ such that $G_1.G_2 = G_1.G_3 \neq \perp$. Let g be a graph isomorphism between $G_1.G_2$ and $G_1.G_3$. It is easy to check that g must map the k^{th} occurrence of a in $G_1.G_2$ to the k^{th} occurrence of a in $G_1.G_3$. Hence, g maps G_1 to G_1 and G_2 to G_3. Therefore, G_2 and G_3 are isomorphic dependence graphs.

iv) As above. ◊

3. Properties of the Partial Order Set of Traces.

We will now investigate PoSet properties of traces. They derive from their interpretation as dependence graphs. Recall that the notion of dependence graph has been defined up to an isomorphism, that is up to a bijection between vertices which preserves edges and labelling. As the proofs of partial order properties of traces use inclusion, union and intersection of subsets of vertices, we need to fix a canonical dependence graph associated with each trace. This canonical dependence graph is given by the mapping Γ (Definiton 2.2), which extends clearly to traces.

The first result is that the PoSet of traces is coherently complete. Let us recall first some definitions. Let $(Z,<)$ be a PoSet and let X be a subset of Z. X is pairwise bounded (or coherent) if any pair $\{x,y\} \subset X$ admits a common upper bound in Z. A PoSet $(Z,<)$ is coherently complete if any pairwise bounded subset of Z admits a least upper bound. Whenever it exists, the least upper bound of a subset X will be denoted by $\sqcup X$.

Theorem 3.1. The PoSet of traces is coherently complete.

Proof. Let X be a pairwise bounded set of traces. For every trace t, we consider the canonical representation $\Gamma(t)=(V_t,E_t,\lambda_t)$ (Definition 2.2). Define the labelled graph $G=(V,E,\lambda) = \bigcup_{t\in X} \Gamma(t)$ by $V = \bigcup_{t\in X} V_t$, $E = \bigcup_{t\in X} E_t$ and for all $(a,j) \in V$, $\lambda(a,j) = a$.
Note that if $x \in V_t$ then $\lambda_t(x) = \lambda(x)$.
- We claim that if $(y,x) \in E$ and $t \in X$ is such that $x \in V_t$ then $y \in V_t$ and $(y,x) \in E_t$.

If $(y,x) \in E$ then there exists s in X such that $(y,x) \in E_s$. Now let $t \in X$ be such that $x \in V_t$. Since X is pairwise bounded, there exists a trace r such that $s < r$ and $t < r$. Then, by definition of the prefix order on traces (Definition 2.6), we get $(y,x) \in E_r$ and then $(y,x) \in E_t$ and $y \in V_t$.
- Next, we claim that if $x_0 \to x_1 \to \dots \to x_n$ is a finite path in G and if t is a trace in X such that $x_n \in V_t$, then $x_0 \to x_1 \to \dots \to x_n$ is a path in $\Gamma(t)$.

We will prove this claim by induction on the length of the path. By claim 1, this is true for $n = 1$. Next, let n be greater than 1. By claim 1 we get $x_{n-1} \in V_t$ and $(x_{n-1},x_n) \in E_t$. Then, by induction we get $x_0 \to x_1 \to \dots \to x_{n-1}$ is a path in $\Gamma(t)$. Therefore, $x_0 \to x_1 \to \dots \to x_n$ is a path in $\Gamma(t)$ and claim 2 is proved.
- We will prove now that G is a dependence graph. First, V is a subset of $A \times \mathbb{N}$, hence it is countable. Next, let us see that G is acyclic. Assume that $x_0 \to x_1 \to \dots \to x_n = x_0$ is a cycle in G. Let $t \in X$ be such that $x_0 \in V_t$. By claim 2, we get that $x_0 \to x_1 \to \dots \to x_n$ is a cycle in $\Gamma(t)$ which is a contradiction. Now we prove that G satisfies (γ_1). Let $x,y \in V$ be such that $(\lambda(x),\lambda(y)) \in D$. There exist r and s in X such that $x \in V_r$ and $y \in V_s$. Let t be a trace such that $r < t$ and $s < t$. Then $x,y \in V_t$ and for instance $(x,y) \in E_t$. Since $s < t$ we obtain $(x,y) \in E_s \subset E$. Conversely, if $(x,y) \in E$, there exists t in X such that $(x,y) \in E_t$ and $(\lambda(x),\lambda(y)) \in D$.

Let us prove now that G satisfies (γ_2). Let x be a vertex of G. There exists $t \in X$ such that $x \in V_t$. Let $y \in V$ be such that $(y,x) \in E^*$. There is a finite path from y to x in G:
$y = x_0 \to x_1 \to \dots \to x_n = x$. By claim 2, we get that this is a path of $\Gamma(t)$. Hence we get $\{y \in V, (y,x) \in E^*\} = \{y \in V_t, (y,x) \in E_t^*\}$ which is finite since t satisfies (γ_2).
- This proves that G is a dependence graph. Now we easily deduce from claim 1 that G is an upper bound of X. Finally, it is clear that G is the least upper bound of X. ◊

Note that from the above construction we can easily deduce that the least upper bound of a trace language X is finite iff X is a finite set of finite traces. Many properties follow directly from previous theorem. We state them in the following corollary. Let us first recall some definitions. A subset X of a PoSet $(Z,<)$ is directed if any pair $\{x,y\} \subset X$ admits a common upper bound **in X**. A PoSet is a CPO (Complete Partial Order) if it admits a least element and if any directed subset admits a least upper bound. A PoSet is a complete lattice iff every subset admits a least upper bound and a greatest lower bound.

Corollary 3.2.

i) The PoSet of traces is a CPO with the empty trace as least element.
ii) Any increasing sequence of traces admits a least upper bound.

iii) Every non empty set of traces admits a greatest lower bound.
iv) For any trace t, the set of prefixes of t is a complete lattice.

For a C.P.O. some elements are essential in the sense that they allow to reconstruct the whole PoSet starting only from them. Compacts and primes are such elements. Let $(Z,<)$ be a CPO. An element y of Z is said to be compact (resp. prime) iff whenever X is a directed subset (resp. X is a subset such that $\sqcup X$ exists) and $y < \sqcup X$, then $y < x$ for some x in X. The sets of compact and prime elements of Z are denoted C(Z) and PR(Z) respectively. For any element x in Z, the sets of compact and prime elements lower than x are denoted C(x) and PR(x) respectively. The CPO $(Z,<)$ is a Scott-Domain iff C(Z) is countable and $x = \sqcup C(x)$ for every x in Z. The CPO $(Z,<)$ is Prime Algebraic iff $x = \sqcup PR(x)$ for every x in Z. We will see that the set of traces is a Scott-Domain and is Prime Algebraic. But first, we prove an auxiliary result.
Let $G = (V,E,\lambda)$ be a dependence graph and let x be a vertex of G. We say that x is a greatest (resp. maximal) vertex of G if $\{y \in V \,/\, (y,x) \in E^*\} = V$ (resp. $\{y \in V \,/\, (x,y) \in E\} = \emptyset$). Note that a trace which admits a greatest vertex must be finite. Let t be a trace. We denote by FP(t) the set of finite prefixes of t and by GP(t) the set of prefixes of t which admit a greatest vertex.

Proposition 3.3. For any trace t , we have $t = \sqcup GP(t) = \sqcup FP(t)$.

Proof. GP(t) is bounded by t, then it admits a least upper bound (Theorem 3.1). Let $t' = \sqcup GP(t)$. Clearly we have $t' < t$. Conversely let $\Gamma(t) = (V, E, \lambda)$ and $\Gamma(t') = (V', E', \lambda')$. We claim that $V \subset V'$. Let x be a vertex in V. Define the graph $G_x = (V_x, E_x, \lambda_x)$ by $V_x = \{y \in V \,/\, (y,x) \in E^*\}$, $E_x = E_{|V_x \times V_x}$ and $\lambda_x = \lambda_{|E_x}$. Clearly G_x is a finite dependence graph, x is a greatest vertex in G_x and G_x is a prefix of $\Gamma(t')$. Then x is a vertex in V', which proves the claim. Now $t' < t$ and $V \subset V'$ imply that $t = t'$. Therefore t is the least upper bound of GP(t). Finally, $t = \sqcup FP(t)$ follows from $GP(t) \subset FP(t)$. ◊

The following theorem characterizes compact traces and prime traces. Note that these characterizations prove that FP(t) = C(t) and that GP(t) = PR(t). The previous proposition and the characterization of primes imply obviously that the PoSet of traces is Prime Algebraic and then is a Scott-Domain.

Theorem 3.4. The PoSet of traces is a Scott-Domain and is Prime Algebraic. Moreover
i) A trace is compact iff it is finite.
ii) A trace is prime iff it admits a greatest vertex.

Proof.
i) Let X be a directed trace language and let $G = \Gamma(\sqcup X)$. Recall that $G = (V,E,\lambda) = \bigcup_{t \in X} \Gamma(t)$ (proof of Theorem 3.1). Now let s be a finite trace, let $G' = (V', E', \lambda') = \Gamma(s)$ and assume that $s < \sqcup X$. Since G' is a subgraph of G, for all x in V' there exists t_x in X such that x is a vertex of $\Gamma(t_x)$. Since G' is finite and X is directed, there exists t in X such that $t_x < t$ for all x in V'. Therefore, if $\Gamma(t) = (V_t, E_t, \lambda_t)$ we get $V' \subset V_t$. Then, since s and t are both prefixes of $\sqcup X$,

we necessarily get $s < t$. This proves that any finite trace is compact. Conversely, let t be an infinite trace. By Proposition 3.3, t is the least upper bound of FP(t). Now FP(t) is clearly directed and t cannot be a prefix of a finite trace. Therefore t is not compact.

ii) Let s be a trace such that $\Gamma(s) = (V',E',\lambda')$ admits a greatest vertex x. Let X be a trace language which admits a least upper bound and let $G = (V,E,\lambda) = \Gamma(\sqcup X)$. If $s < \sqcup X$ then $x \in V$, thus there exists t in X such that $x \in V_t$ (with $\Gamma(t) = (V_t, E_t, \lambda_t)$). Since s and t are prefixes of $\sqcup X$ and since x is a greatest vertex of s, we get that s is a prefix of t. This proves that s is a prime. Conversely, let s be a prime. Since a prime is compact, s is a finite trace. Let $G = (V, E, \lambda) = \Gamma(s)$ and let $x_1, \dots, x_k$ be the maximal vertices in G. For all i, let $G_i = G_{x_i}$ be the dependence graph defined from G and x_i as in the proof of Proposition 3.3. Clearly G is the least upper bound of $G_1, \dots, G_k$. Since s is a prime, G is a prefix of some G_i. Therefore there is only one maximal vertex in G, namely x_i. ◊

Note that similar results have been partly established for finite traces in [R.T], with a rather distinct formalism based on finite inductions. In that case we get only a finitely coherent PoSet, since an infinite set of finite traces does not admit a least upper bound within the set of finite traces.

4. Prime Event Structures Representation of Traces.

We will now turn to the prime event structure representation of the PoSet of finite and infinite traces. Analogous results have been established for the PoSet of finite traces in [R.T] and we generalize them now to the infinite case. The results are mostly similar but the proofs are completely different since, in the infinite case, they cannot be performed using finite inductions on configurations.

The event structure model has been introduced by G. Winskel. Here, we only use what Winskel calls Prime Event Structure. Basically a *Prime Event Structure Model* is a partially ordered set of events together with a conflict relation. The partial order relation is meant to capture causal dependences between events. A conflict relation # on a PoSet $(E,<)$ is an irreflexive and symmetric relation which is inherited, that is: $\forall$ u,v,w $\in$ E, u # v and $u < w$ imply w # v. The conflict relation modelizes non-determinism (choice) so that two events that are in conflict cannot both occur in a same behaviour. Consequently two events that are neither ordered nor in conflict may occur concurrently in some behaviour. Thus, in a prime event structure, basic phenomena of a behaviour, such as non-determinism and concurrency, are represented in an explicit and clearly separated fashion. By labelling the events with the actions taken from some alphabet we get *Labelled Prime Event Structures.* The characterization of infinite trace PoSets will be in terms of subclasses of labelled prime event structures.

Definition 4.1. A labelled prime event structure is a 5-uple $ES = (E,<,\#,\ell,A)$, where $(E,<)$ is a PoSet, # is a conflict relation on $(E,<)$ and ℓ is a labelling function from E onto A.

The 3-uple $(E,<,\#)$ is called the underlying prime event structure, elements in E are called events. A prime event structure ES is said to be finitary iff any element u in E admits a finite past, that is, iff the set of elements lower than u is finite. *From now on, every event structure that we consider will be assumed to be prime and finitary.* Note that if two events are in conflict then they do not have a common bound. Otherwise e' # e", e' < e, e" < e would imply e # e, which contradicts the fact that # is irreflexive.

Figure 4.1 gives an example of an event structure which modelizes a situation where a producer communicates items via an unbounded buffer to a consumer. The producer can stop after producing zero, one, or more items. Both the producer and the consumer are assumed to work sequentially. The set of events is $E = \{p_1,p_2,p_3,\ldots,s_0,s_1,s_2,s_3,\ldots,c_1,c_2,c_3,\ldots\}$ where p stands for "produce", c for "consume" and s for "stop". The alphabet of actions is {p,c,s}, with the obvious labelling function $\ell(x_i) = x$. The partial order relation is the reflexive and transitive closure of the relation depicted by the arrows. Finally, the conflict relation is the closure under inheritance of the initial conflict relation depicted by the grey lines.

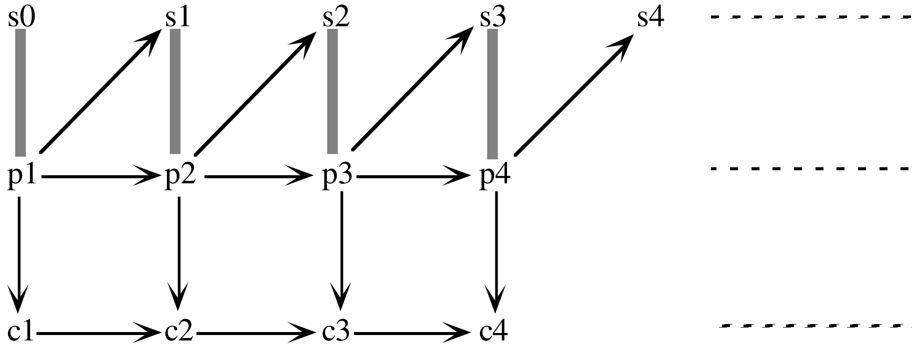

Figure 4.1

First, we introduce and study a labelled event structure canonically associated with a PoSet of traces.

Definition 4.2. Let D be a dependence relation over an alphabet A. We define the event structure $\nabla(\mathcal{G}(A,D)) = (E,<,\#,\ell,A)$ associated with the PoSet of traces $\mathcal{G}(A,D)$ by:

- $E = PR(\mathcal{G}(A,D))$ is the set of prime traces.
- < is the prefix order on traces restricted to E.
- Let u,v be in E, u # v iff u and v do not admit a common upper bound in $\mathcal{G}(A,D)$.
- Let u be in E, $\ell(u) = \lambda(\max(u))$ where max(u) is the greatest vertex in u.

Note that the relation # defined above is clearly irreflexive, symmetric and inherited. Therefore it is a conflict relation and $\nabla(\mathcal{G}(A,D))$ is an event structure. In order to exhibit the connection between $\mathcal{G}(A,D)$ and $\nabla(\mathcal{G}(A,D))$, we define now the notion of configuration of an event structure.

Definition 4.3. Let $ES = (E,<,\#,\ell,A)$ be a labelled event structure. A subset C of E is closed in the past iff $(p \in C \text{ and } p' < p) \Rightarrow (p' \in C)$ and conflict-free iff $\# \cap (C \times C) = \emptyset$. A configuration of ES is a subset of E that is both conflict-free and closed in the past. The sets of configurations and finite configurations of ES are denoted by $\mathcal{C}(ES)$ and $\mathcal{F}\mathcal{C}(ES)$.

For instance, in the event structure of Figure 4.1, $\{p_1,p_2,p_3,\ldots,c_1,c_2,c_3,\ldots\}$ is an infinite configuration and $\{p_1,p_2,p_3,c_1,c_2,c_3,s_3\}$ and $\{p_1,p_2,p_3,p_4,p_5,c_1,c_2\}$ are finite ones.
We prove now that $\nabla(\mathcal{G}(A,D))$ represents exactly $\mathcal{G}(A,D)$. Namely, we show that there is a one to one correspondance between traces of $\mathcal{G}(A,D)$ and configurations of $\nabla(\mathcal{G}(A,D))$.

Proposition 4.4. $(\mathcal{C}(\nabla(\mathcal{G}(A,D))),\subset)$ is isomorphic to $(\mathcal{G}(A,D),<)$.

Proof. Let us define two mappings $\underline{\nabla}$ and $\underline{\Delta}$ between $\mathcal{G}(A,D)$ and the set of configurations of $\nabla(\mathcal{G}(A,D))$. For every trace u, set $\underline{\nabla}(u) = PR(u)$, the set of primes that are lower than u. Clearly PR(u) is closed in the past and conflict-free, thus $\underline{\nabla}(u)$ is a configuration and the mapping $\underline{\nabla}$ is well-defined. On the other hand, for every configuration C, we set $\underline{\Delta}(C) = \sqcup C$, the least upper bound of C. Since a configuration is conflict-free, it is pairwise bounded. Thus, by Theorem 3.1, it admits a least upper bound and the mapping $\underline{\Delta}$ is well-defined too.
By Proposition 3.3 and Theorem 3.4, we get $\underline{\Delta}\circ\underline{\nabla}(u) = u$. It remains to prove that $\underline{\nabla}\circ\underline{\Delta}(C) = PR(\sqcup C) = C$. If p is a prime bounded by $\sqcup C$ then it is bounded by some element in C. Since C is closed in the past, we get that p is in C and the inclusion $PR(\sqcup C) \subset C$ is proved. Conversely, if p is in C, it is bounded by $\sqcup C$. Moreover, by definition of E, it is a prime. Thus p is in $PR(\sqcup C)$ and the converse inclusion is proved. Since clearly $u < u' \Leftrightarrow PR(u) \subset PR(u')$, the mappings $\underline{\nabla}$ and $\underline{\Delta}$ are PoSet isomorphisms. ◊

Note that this proof associates a single trace $u = \underline{\Delta}(C) = \sqcup C$ with every configuration C, and a configuration $C = PR(u) = \underline{\nabla}(u)$ with every trace u. We get a new equivalent vision of traces and can identify a trace with a configuration. Following this identification, events are associated with dependence graphs that admit a greatest vertex. For instance, let $A = \{a,c\}$ be an alphabet and let $I = \{(a,c), (c,a)\}$ be an independence relation over A. The set of events is $E = \{a^n, n \in \mathbb{N}\} \cup \{c^n, n \in \mathbb{N}\}$, where the trace associated with the word u is simply denoted by u in order to simplify the notations. The order relation is the reflexive and transitive closure of $a^n < a^{n+1}$ and $c^n < c^{n+1}$ for all n in $\mathbb{N}$. The conflict relation is empty and the labelling function is $\ell(a^n) = a$, $\ell(c^n) = c$. For the same independence relation but over the alphabet $A = \{a,b,c\}$, Figure 4.2 gives a small part of the associated event structure. The image of a trace by $\underline{\nabla}$ is the restriction to a subset of events (a configuration) of the event structure presented in Figure 4.2. For instance, the configuration associated with the trace acbac is {a, c, acb, acba, acbc}.

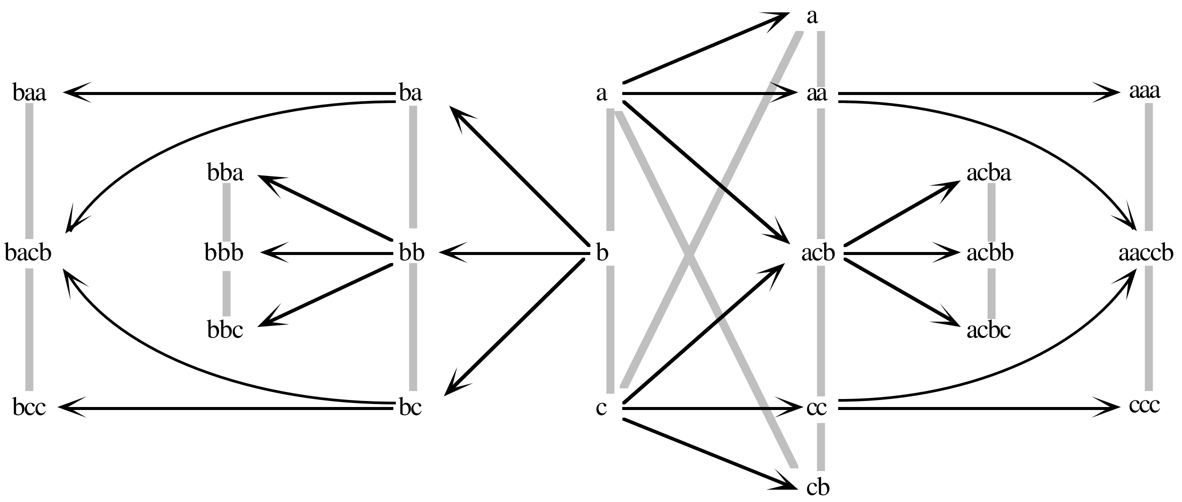

Figure 4.2

Till now, for every dependence relation D over an alphabet A, we have associated a labelled event structure ES in such a way that $(\mathcal{G}(A,D),<)$ and $(\mathcal{C}(ES),\subset)$ are isomorphic PoSets. We will now characterize the subclass of event structures that can be obtain in this way. Next, we will move back from these event structures to dependence relations over alphabets. But first we need some definitons.

Definition 4.5. Let $ES = (E,<,\#,\ell,A)$ be a labelled event structure. An event e is said to be enabled at a configuration C if $e \notin C$ and $C \cup \{e\}$ is a configuration. The set of events enabled at C is denoted by $\mathcal{E}(C)$. The concurrency relation $co = (E \times E) - (\# \cup < \cup >)$ is defined as the complementary of the order and the conflict relations. The immediate predecessor relation $<\bullet$ is defined by $u <\bullet v$ iff $u < v$, $u \neq v$ and for all w in E, $u < w < v$ implies $w = u$ or $w = v$.

For instance, in the event structure of Figure 4.1, c_4 is enabled at $\{p_1,p_2,p_3,\ldots,c_1,c_2,c_3\}$, s_5 is enabled at $\{p_1,p_2,p_3,p_4,p_5,c_1,c_2\}$ and no event is enabled at $\{p_1,p_2,p_3,c_1,c_2,c_3,s_3\}$. (c_2,p_4) and (c_2,s_2) are concurrent. c_2 is an immediate predecessor of c_3 and p_3 is an immediate predecessor of s_3, p_4 and c_3.

We define now the subclass of event structures which represent exactly the PoSets of traces, namely partially commutative event structures.

Definition 4.6. Let $ES = (E,<,\#,\ell,A)$ be a labelled event structure. We say that ES is a Partially Commutative Event Structure iff it satisfies conditions $\mathcal{M}$ and $\mathcal{I}$:

$\mathcal{M}$: $\forall a \in A, \forall C \in \mathcal{F}\mathcal{C}(ES)$, there exists one and only one e in $\mathcal{E}(C)$ such that $\ell(e) = a$

$\mathcal{I}$: $(\ell^{-1}(a) \times \ell^{-1}(b)) \cap co \neq \emptyset \Rightarrow (\ell^{-1}(a) \times \ell^{-1}(b)) \cap <\bullet = \emptyset$

Note that conditions $\mathcal{I}$ and $\mathcal{M}$ imply $(\ell^{-1}(a) \times \ell^{-1}(b)) \cap <\bullet = \emptyset \Rightarrow (\ell^{-1}(a) \times \ell^{-1}(b)) \cap co \neq \emptyset$.

Intuitively, condition $\mathcal{M}$ means that any finite configuration can be extended with any action. On the other hand, two events which are concurrent should be labelled by independent actions and if e is an immediate predecessor of f then their labels should be dependent. Therefore condition $\mathcal{I}$ means that if two actions must be independent as labels of some events, they

cannot be dependent as labels of some other events. Note that the converse of condition $\mathcal{I}$ (which intuitively must also be true) can be derived from $\mathcal{I}$ and $\mathcal{M}$. Consider the empty configuration, let e in $\mathcal{E}(\emptyset)$ be such that $\ell(e) = a$ and let f in $\mathcal{E}(\{e\})$ be such that $\ell(f) = b$. Since $\{e\}$ and $\{e,f\}$ are configurations, the following hold : not $(f < e)$, not $(e \# f)$ and either e co f or $e <\!\!\bullet f$. Therefore, $(\ell^{-1}(a) \times \ell^{-1}(b)) \cap <\!\!\bullet = \emptyset \Rightarrow (\ell^{-1}(a) \times \ell^{-1}(b)) \cap \mathrm{co} \neq \emptyset$.
We prove now that the event structure associated by ∇ with a set of dependence graphs is partially commutative.

Proposition 4.7. Let D be a dependence relation over a finite alphabet A. Then $\nabla(\mathcal{G}(A,D))$ is a partially commutative event structure.

Proof.

- Let us first describe PR(G) for any trace $G = (V,E,\lambda)$ in $\mathcal{G}(A,D)$. For any vertex x in V, define the trace G_x as in the proof of Proposition 3.3. Clearly, G_x is a prefix of G with x as greatest vertex. Thus G_x is a prime (Theorem 3.4) bounded by G. Now, let G' be a prime bounded by G. G' is a subgraph of G and by Theorem 3.4, it admits a greatest vertex, say x. Then, clearly $G' = G_x$. Therefore, we obtain $PR(G) = \{G_x, x \in V\}$. In particular, this implies that the number of primes bounded by G is the length of G : $|PR(G)| = |G| = |V|$.
- Now we prove that $\nabla(\mathcal{G}(A,D))$ satisfies $\mathcal{M}$. Let a be in A and let C be in $\mathcal{FC}$. We first prove the existence of an event e in $\mathcal{E}(C)$ such that $\ell(e) = a$. Define the traces $G = (V,E,\lambda) = \sqcup C$ and $G' = (V',E',\lambda') = G \otimes \Gamma(a)$. By Definition 2.7, G is a subgraph of G' and $V' = V \cup \{z\}$ with $\lambda'(z) = a$. Thus, $PR(G') = PR(G) \cup \{e\}$ with $e = G'_z$. Then, by Definition 4.2, e is an event of $\nabla(\mathcal{G}(A,D))$ and $\ell(e) = \lambda'(z) = a$. Now we have $PR(G) = PR(\sqcup C) = \underline{\nabla} \circ \underline{\Delta}(C) = C$ (Proposition 3.4). Thus, $C \cup \{e\} = PR(G') = \underline{\nabla}(G')$ is a configuration of $\nabla(\mathcal{G}(A,D))$. Since e is not in C, e is enabled at C and the existence is proved.
It remains to prove that this event is unique. Let $e \in \mathcal{E}(C)$ be such that $\ell(e) = a$. Set $G = \sqcup C$ and $G' = \sqcup (C \cup \{e\})$. G is clearly a prefix of G'. Now, $PR(G) = C$ and $PR(G') = C \cup \{e\}$, whence $|G'| = |PR(G')| = |C \cup \{e\}| = |C| + 1 = |G| + 1$. Thus, $G' = G \otimes \Gamma(x)$ for some letter x in A. But we saw above that in this case, $PR(G') = PR(G) \cup \{e'\}$ with $\ell(e') = x$. Consequently, $C \cup \{e\} = PR(G') = PR(G) \cup \{e'\} = C \cup \{e'\}$. Therefore, $e = e'$, $x = a$ and e is the unique prime in $PR(G \otimes \Gamma(a)) \setminus PR(G)$.
- It remains to prove that $\nabla(\mathcal{G}(A,D))$ satisfies $\mathcal{I}$. This point follows directly from the two following facts : $(\ell^{-1}(a) \times \ell^{-1}(b)) \cap \mathrm{co} \neq \emptyset$ implies $(a,b) \in I$ and $(\ell^{-1}(a) \times \ell^{-1}(b)) \cap <\!\!\bullet \neq \emptyset$ implies $(a,b) \in D$. We prove them now.

Let e and f be two concurrent events such that $\ell(e) = a$ and $\ell(f) = b$. Since e and f are conflict-free, $e \sqcup f = G = (V,E,\lambda)$ exists in $\mathcal{G}(A,D)$. Since e and f are primes, they admits greatest vertices x and y which satisfy $\lambda(x) = \ell(e) = a$ and $\lambda(y) = \ell(f) = b$. Since e and f are prefixes of G, x and y are vertices of G and since e and f are not ordered, x and y are distinct and neither (x,y) nor (y,x) is in E. By definition of a dependence graph, we get $(a,b) \in I$, which proves the first fact.
Let e and f be two events such that $e <\!\!\bullet f$, $\ell(e) = a$ and $\ell(f) = b$. Let x and y be the greatest vertices of e and f respectively. Since e is a prefix of f, there exists a path in f from x to y : $x = x_0 \rightarrow x_1 \rightarrow \ldots \rightarrow x_k = y$. The trace $g = G_{x_1}$ is prime and clearly $e < g < f$. Since $e <\!\!\bullet f$

we get g = f. Thus (x,y) is an edge of f, which implies $(\lambda(x),\lambda(y)) = (a,b) \in D$. This achieves the second fact. ◊

Now, from a partially commutative event structure, we have to go back to a PoSet of traces. Thus, starting with a partially commutative event structure, we must define a dependence relation over the alphabet of labels. In order to do this, we need the following lemma.

Lemma 4.8. Let $ES = (E,<,\#,\ell,A)$ be a partially commutative labelled event structure. Then the relation $I = \{(a,b) \in A \times A \,/\, (\ell^{-1}(a) \times \ell^{-1}(b)) \cap co \neq \emptyset\}$ is symmetric and irreflexive. Moreover, e' co e" implies $(\ell(e'),\ell(e'')) \in I$ and $e' <\bullet e''$ implies $(\ell(e'),\ell(e'')) \notin I$.

Proof. Since the relation co is symmetric, so is I. Suppose now that $(a,a) \in I$ for some a in A. Then by definition of I, there exist e' and e" in E such that e' co e" and $\ell(e') = \ell(e'') = a$. We can easily verify that the set $C = \{e \in E \,/\, (e < e' \text{ or } e < e'') \text{ and } e \neq e' \text{ and } e \neq e''\}$ is a configuration where e' and e" are both enabled. Using condition $\mathcal{M}$, we obtain e' = e" which contradicts e' co e". Thus I is irreflexive. The two other facts are clear from the definition of I and from condition $\mathcal{I}$. ◊

As announced above, this lemma allows us to define the set of traces associated with a partially commutative labelled event structure.

Definition 4.9. Let $ES = (E,<,\#,\ell,A)$ be a partially commutative labelled event structure. We define the relations I(ES) and D(ES) on A as follows :

$$I(ES) = \{(a,b) \,/\, (\ell^{-1}(a) \times \ell^{-1}(b)) \cap co \neq \emptyset\} \text{ and } D(ES) = A \times A - I(ES).$$

We denote $\Delta(ES) = \mathcal{G}(A,D(ES))$ the corresponding set of traces.

It remains to establish that Δ and ∇ are nearly inverse bijections. It is the purpose of the two following theorems.

Theorem 4.10. Let D be a dependence relation over an alphabet A. Then

$$\Delta\circ\nabla(\mathcal{G}(A,D)) = \mathcal{G}(A,D)$$

Proof. Let $ES = \nabla(\mathcal{G}(A,D))$ and $D' = D(ES)$. We must prove that D = D'. Let a,b be two letters in A. If $(a,b) \in D$ then u = a and v = ab are prime traces. Thus, by definition of ∇, u and v are events of ES such that $\ell(u) = a$ and $\ell(v) = b$. Clearly, we have $u <\bullet v$. Since ES is a partially commutative event structure, this implies $(a,b) \in D'$. Conversely, if $(a,b) \notin D$ then u = a and v = b are prime traces. Thus, u and v are events of ES such that $\ell(u) = a$ and $\ell(v) = b$. Since $u \sqcup v = ab$, u and v are neither in conflict nor ordered. Therefore u and v are concurrent and $(a,b) \notin D'$. ◊

Theorem 4.11. Let $ES = (E,<,\#,\ell,A)$ be a partially commutative event structure. The event structure $\nabla\circ\Delta(ES)$ is isomorphic to ES.

Proof. Let us denote simply $I = I(ES) = \{(a,b) \,/\, (\ell^{-1}(a) \times \ell^{-1}(b)) \cap co \neq \emptyset\}$ and $D = D(ES)$. First, we define inverse **isomorphisms** between $(\mathcal{F}\mathcal{C}(ES) , \subset)$ and $(\mathcal{F}\mathcal{G}(A,D) , <)$. Then, we will use them to define inverse isomorphisms between ES and $\nabla\circ\Delta(ES)$.

• Let C be a finite configuration of ES. Define $G = (C, \rightarrow, \ell)$ by: for all x,y in C, $x \rightarrow y$ iff $x < y$ and $(\ell(x),\ell(y)) \in D$. Since $<$ is a partial order on C, G is clearly a labelled acyclic graph. Let x, y in C be such that $(\ell(x),\ell(y)) \in D$. By definition of D, x and y cannot be concurrent. Since C is conflict-free, we get $x < y$ or $y < x$. Thus $x \rightarrow y$ or $y \rightarrow x$ and G verifies (γ_1). Therefore G is a finite dependence graph. Thus we can define a mapping Π from $\mathcal{F}\mathcal{C}(ES)$ to $\mathcal{F}\mathcal{G}(A,D)$ by $\Pi(C) = (C, \rightarrow, \ell)$. Note that obviously $C \subset C'$ implies $\Pi(C)$ is a prefix of $\Pi(C')$, thus, Π is an increasing mapping. Moreover we will prove that $<$ restricted to C is the reflexive and transitive closure of $\rightarrow$. Obviously, $x \rightarrow^* y$ implies that $x < y$. Conversely, let x, y in C be such that $x < y$ and $x \neq y$. Either $x <\!\bullet\, y$ then $(\ell(x),\ell(y)) \in D$ because ES verifies $\mathcal{I}$ and we get $x \rightarrow y$. Or there exists z in C such that $x < z <\!\bullet\, y$. As above we get $z < y$ and by induction we obtain $x \rightarrow^* y$.

• Conversely we define a mapping η from $\mathcal{F}\mathcal{G}(A,D)$ to $\mathcal{F}\mathcal{C}(ES)$. First, we define η on words by induction. Set $\eta(\varepsilon) = \emptyset$. Now let u be a finite word and let a be a letter of A. Assume that $C = \eta(u)$ is defined. Let e be the unique event enabled at C such that $\ell(e) = a$ (condition $\mathcal{M}$ on ES). We set $\eta(ua) = C \cup \{e\}$. Let us prove that η maps all equivalent words to the same configuration. Considering the definition of η and of the equivalence relation, it is sufficient to show that if $u \in A^*$ and $(a,b) \in I$ then $\eta(uab) = \eta(uba)$. Let $C = \eta(u)$, let e be the unique event in $\mathcal{E}(C)$ such that $\ell(e) = a$ and let f be the unique event in $\mathcal{E}(C \cup \{e\})$ such that $\ell(f) = b$. We have $\eta(uab) = C \cup \{e,f\}$. Let us see that $f \in \mathcal{E}(C)$. Assume that $e < f$, then clearly $e <\!\bullet\, f$. Thus, $(\ell^{-1}(a) \times \ell^{-1}(b)) \cap <\!\bullet \neq \emptyset$ and by condition $\mathcal{I}$, we get $(a,b) \notin I$, which is a contradiction. Since $C \cup \{e,f\}$ is a configuration and not $(e < f)$, the set $C \cup \{f\}$ is a configuration too, that is, $f \in \mathcal{E}(C)$. Since $\ell(f) = b$, we get $\eta(ub) = C \cup \{f\}$. Now, clearly $e \in \mathcal{E}(C \cup \{f\})$ and $\ell(e) = a$, thus we get $\eta(uba) = C \cup \{e,f\} = \eta(uab)$. This allows us to consider, in the remainder, η as a mapping from $\mathcal{F}\mathcal{G}(A,D)$ to $\mathcal{F}\mathcal{C}(ES)$. Note that if u is a prefix of v then $\eta(u) \subset \eta(v)$, thus, η is an increasing mapping.

• We prove by induction on $|C|$ that for all C in $\mathcal{F}\mathcal{C}(ES)$, $\eta\circ\Pi(C) = C$. It is clear that $\eta\circ\Pi(\emptyset) = \emptyset$. Let C be a non empty finite configuration and let e be a maximal event in C. Let C' be the finite configuration $C \setminus \{e\}$. Since e is maximal in C, it is easy to verify that $\Pi(C) = \Pi(C').\ell(e)$. By induction we get $\eta\circ\Pi(C') = C'$. Since e is enabled at C' we get $\eta\circ\Pi(C) = \eta(\Pi(C').\ell(e)) = C' \cup \{e\} = C$.

• Again by induction we prove that for all u in $\mathcal{F}\mathcal{G}(A,D)$, $\Pi\circ\eta(u) = u$. It is clear that $\Pi\circ\eta(\varepsilon) = \varepsilon$. Let u be a finite trace and let a be a letter in A. We set $C = \eta(u)$ and $C \cup \{e\} = \eta(ua)$. As stated above, we have $\Pi(C \cup \{e\}) = \Pi(C).\ell(e)$. By induction we get $\Pi(C) = \Pi\circ\eta(u) = u$. Thus $\Pi\circ\eta(ua) = \Pi(C).\ell(e) = ua$.

Finally, we have proved that Π and η are inverse **isomorphisms** between $(\mathcal{F}\mathcal{C}(ES) , \subset)$ and $(\mathcal{F}\mathcal{G}(A,D) , <)$.

Using the bijections Π and η we will now define inverse isomorphisms between ES and $ES' = \nabla\circ\Delta(ES) = (E',<',\#',\ell',A)$.

- First, we define $\overline{\Pi}$ from E to E' by $\overline{\Pi} = \Pi \circ \mathrm{past}$, where $\mathrm{past}(e) = \{x \in E, x < e\}$ for all e in E. Let $G = (C, \rightarrow, \ell) = \overline{\Pi}(e)$. Since $<$ restricted to C is the reflexive and transitive closure of $\rightarrow$, e is the greatest vertex of G. Thus G is a prime and is an event of E'. Therefore the mapping $\overline{\Pi}$ is well defined.
- Remark that if a configuration C has several maximal events then $\eta(C)$ has several maximal vertices and thus is not prime. Therefore, since $u \in E'$ is a prime trace, the configuration $C = \eta(u)$ has a greatest event which will be denoted by max(C). Now, define the mapping $\overline{\eta}$ from E' to E by $\overline{\eta} = \max \circ \eta$.
- If e is an event of E then $\max \circ \mathrm{past}(e) = e$ since past(e) is a configuration with e as greatest event. Now, if a finite configuration C of E has a greatest event then clearly $\mathrm{past} \circ \max(C) = C$. Therefore, since Π and η are inverse bijections, so are $\overline{\Pi}$ and $\overline{\eta}$. It remains to prove that $\overline{\Pi}$ and $\overline{\eta}$ are event structure isomorphisms.
- Let $e,f \in E$ be such that $e < f$. We have $\mathrm{past}(e) \subset \mathrm{past}(f)$. Thus, $\Pi \circ \mathrm{past}(e)$ is a prefix of $\Pi \circ \mathrm{past}(f)$, that is, $\overline{\Pi}(e) <' \overline{\Pi}(f)$. Conversely, let $u,v \in E'$ be such that $u <' v$. We have $\eta(u) \subset \eta(v)$. Thus, $\max(\eta(u)) < \max(\eta(v))$, that is, $\overline{\eta}(u) < \overline{\eta}(v)$.
- Let $e,f \in E$ be not in conflict. Clearly, $C = \mathrm{past}(e) \cup \mathrm{past}(f)$ is a finite configuration of E. Thus, $\overline{\Pi}(e) = \Pi \circ \mathrm{past}(e)$ and $\overline{\Pi}(f) = \Pi \circ \mathrm{past}(f)$ are both prefixes of $\Pi(C)$. Consequently, $\overline{\Pi}(e)$ and $\overline{\Pi}(f)$ are not in conflict. Conversely, let $u,v \in E'$ be not in conflict. There exists a finite trace w such that u and v are both prefixes of w. Thus $\eta(u) \subset \eta(w)$ and $\eta(v) \subset \eta(w)$. Since $\eta(w)$ is conflict free, this implies that $\overline{\eta}(u) = \max(\eta(u))$ and $\overline{\eta}(v) = \max(\eta(v))$ are not in conflict.
- Finally, for all e in E, $\ell'(\overline{\Pi}(e)) = \ell(\max(\overline{\Pi}(e))) = \ell(e)$.

Thus, $\overline{\Pi}$ and $\overline{\eta}$ are inverse isomorphisms between ES and ES'. ◊

The first straightforward consequence is that the set of dependence graphs $\Delta(ES)$ associated with a partially commutative event structure ES is exactly the set of configurations of ES.

Proposition 4.12. Let $ES = (E,<,\#,\ell,A)$ be a partially commutative event structure. Then $(\Delta(ES),<)$ is isomorphic to $(\mathcal{C}(ES),\subset)$.

Proof. Setting $ES' = \nabla \circ \Delta(ES)$, by Proposition 4.4 we get that $(\mathcal{C}(ES'),\subset)$ is isomorphic to $(\Delta(ES),<)$, and as ES and ES' are isomorphic by Theorem 4.11 we get the required result. ◊

The following theorem sums up previous results, namely the representation of the class of PoSets of traces by that of partially commutative event structures. More precisely, a PoSet of traces is associated with a partially commutative event structure, and finite and infinite traces are associated with finite and infinite configurations; and vice versa.

Theorem 4.13. The class $\mathbb{T}$ of PoSets of finite and infinite traces admits an isomorphic representation within the class $\mathbb{ESC}$ of partially commutative event structures.

5. Conclusion

When looking at a representation of infinitary traces with event structures, we have had to introduce infinity in event structures. In this paper, this representation has been obtained considering possibly infinite configurations. Another way to introduce infinity in event structures would be to allow events to admit an infinite past. We have not considered this eventuality for two reasons. First, the consideration of infinite configurations gives the required relations between infinitary traces and event structures. Second, the study of infinite traces or infinite words is motivated by the consideration of systems which never stop, but in which each action may be viewed as "reachable somewhere in the future", and therefore within a finite past. However and till now, it seems for us that events with infinite past will probably lead to an interesting mathematical theory, the computer science counterpart of which is not straightforward for now!

Many thanks are due to A. Petit for its helpful comments on this paper.

6. References

[A.R] I.J. AALBERSBERG - G. ROZENBERG, Theory of Traces, T.C.S. 60 (1988), p. 1-82.

[B.C1] G. BOUDOL - I. CASTELLANI, Semantics of Concurrency : Partial Orders and Transition Systems, Techn. Rep. 550 (1987) I.N.R.I.A., Sophia Antipolis, France and TAPSOFT 87 , L.N.C.S. 249, p.123-137.

[B.C2] G. BOUDOL - I. CASTELLANI, Permutation of transitions : an Event Structure SEmantics for CCS and SCCS, REX workshop on Concurrency, 1988, Lecture Notes in Computer Science 354, p. 411-427.

[B.M.P] P. BONIZZONI, G. MAURI and G. PIGHIZZINI, "About infinite traces", Proceedings of the ASMICS Workshop on Partially Commutative Monoids, Tech. Rep. TUM-I 9002, Technische Universität München, 1989.

[B.N] L. BOASSON and M. NIVAT, "Adherence of languages", Jour. Comp. Syst. Sci., Vol. 20, 1980, p. 285-309.

[C.F] P. CARTIER - D. FOATA, Problèmes combinatoires de commutation et réarrangements, Lect. Notes in math. 85 (1969).

[C.F.M] I. CASTELLANI - P. FRANCESCHI - U. MONTANARI, Labelled Event Structures: A Model for Observable Concurrency, Proc. I.F.I.P. TC 2 , Formal Description of Programming Concepts II (D.Bjorner ed.), Garmisch-Partenkirchen (1982) North-Holland Amsterdam , p.383-400.

[D1] V. DIEKERT, Combinatorics on Traces, Lecture Notes in Computer Science 454, 1990.

[D2] V. DIEKERT, "On the concatenation of infinite traces", Proceedings of STACS 1991, to appear in Lecture Notes in Computer Science.

[Dr] M. DROSTE, Concurrency, Automata and Domains, Tech. Rep. 1989, University of Essen, to appear in ICALP 90.

[Du] C. DUBOC, Commutations dans les monoïdes libres : un cadre théorique pour l'étude du parrallélisme, Thèse (1986) Rouen Univ. France.

[E] S. EILENBERG, Automata, Languages, and Machines, Academic Press, 1974.

[G1] P. GASTIN, Un modèle asynchrone pour les systèmes distribués, Theoretical Computer Science 74, p. 121-162, 1990.

[G2] P. GASTIN, Infinite Traces, Proceedings of the Spring School of Theoretical Computer Science on "Semantics of concurrency", to appear in Lecture Notes in Computer Science 469, 1990.

[G3] P. GASTIN, "Recognizable and rational languages of finite and infinite traces", Proceedings of STACS 1991, to appear in Lecture Notes in Computer Science.

[G.P.Z] P. GASTIN, A. PETIT, W. ZIELONKA, "A Kleene theorem for infinite trace languages", Tech. Rep. 90, LITP, Université Paris 6, France, 1990.

[G.R1] P. GASTIN - B. ROZOY, Infinitary Partially Commutative Monoids, Tech. Rep. 89-24, (1989), L.I.T.P., University of Paris 7, France.

[G.R2] P. GASTIN - B. ROZOY, The PoSet of Infinitary Traces, Tech. Rep. 90-559, (1990), L.R.I., University of Paris 11, France.

[H.R] H.J. HOOGEBOOM and G. ROZENBERG, Infinitary languages: basic theory and applications to concurrent systems, Lecture Notes in Computer Science 224, p. 266-342, 1986.

[K1] M. Z. KWIATKOWSKA, On Infinitary Trace Languages, Tech. Rep. 31 (1989), University of Leicester, England.

[K2] M.Z. KWIATKOWSKA, "A metric for traces", Information and Processing Letters 35, p. 129-135, 1990.

[La] L. LAMPORT, Time, Clocks and the Ordering of Events in a Distributed System, Comm. A.C.M. Vol.21 n°7 (1978), p. 558-565.

[M1] A. MAZURKIEWICZ, "Concurrent program Schemes and their interpretations", Aarhus University, DAIMI Rep. PB 78, 1977.

[M2] A. MAZURKIEWICZ, Trace Theory, Paper n°16 , Advanced Course on Petri Nets, Bad Honnef (1986).

[N.P.W] M. NIELSEN - G. PLOTKIN - G. WINSKEL, Petri Nets, Event Structures and Domains, T.C.S. 13 (1981), p. 85-108.

[P1] D. PERRIN, Recent Results on Automata and Infinite Words, L.N.C.S. 176 (1984) p. 134-148.

[P2] D. PERRIN, Partial Commutations, ICALP 89, L.N.C.S. 372 (1989), p. 637-651.

[P.P] D. PERRIN - J.E. PIN, Mots Infinis, Tech. Rep. (1990), L.I.T.P. Paris 7 Univ. France.

[R1] B. ROZOY, Un Modèle du Parallélisme : le Monoïde Distribué, Thèse d'état (1987) L.I.U.C. Caen Univ. France.

[R2] B. ROZOY, On Distributed Languages and Models for Distributed Computations, Proceedings of the Spring School of Theoretical Computer Science on "Semantics of concurrency", to appear in Lecture Notes in Computer Science 469, 1990.

[R.T] B. ROZOY - P.S. THIAGARAJAN, Trace Monoids and Event Structures, Tech. Rep. 47 (1987), L.I.T.P., Paris 7 Univ. France, to appear in T.C.S.

[Sh] M.W. SHIELDS, Concurrent Machines, The Computer Journal Vol.28 n°5 (1985), p. 449-465.

[St] P.H. STARKE, Traces and Semi-words, L.N.C.S. 208 (1985), p.332-349.

[W1] G. WINSKEL, Events in Computation, Ph.D. Thesis (1980) Edinburgh Univ. England.

[W2] G. WINSKEL, Categories of Models for Concurrency, L.N.C.S. 197 (1984).

[W3] G. WINSKEL, Event Structures, L.N.C.S. 255 (1987), p. 325-392.