CONSIGLIO
NAZIONALE
DELLE RICERCHE

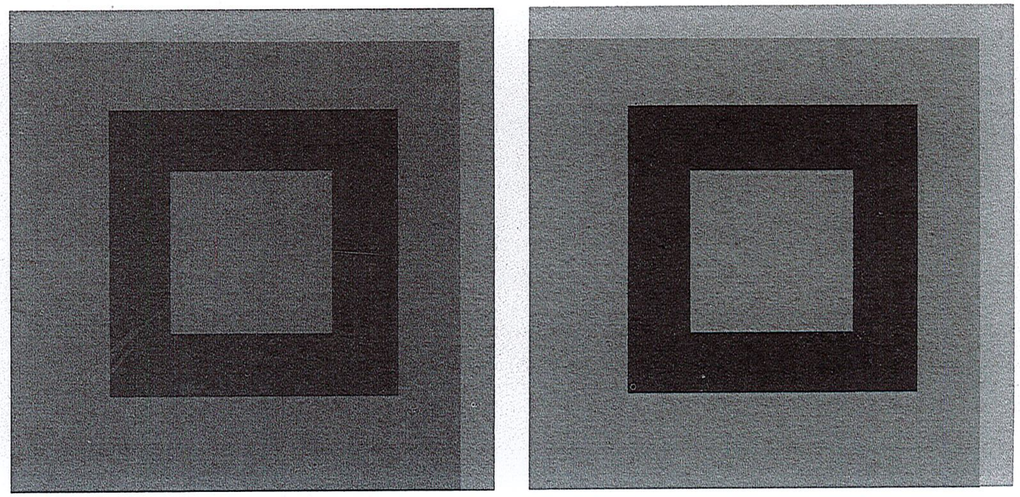
PROGETTO
FINALIZZATO
INFORMATICA

Cnet

43

Labeled Event Structures: A Model for Observable Concurrency

I. Castellani, P. Franceschi, U. Montanari



ETS/PISA

N design

Proc. IFIP TC2 Working Conference on Formal Description of Programming Concepts II, North-Holland Publ. Company, 1983

PROGETTO FINALIZZATO INFORMATICA
C.N.R.
Progetto P_1 Cnet

I. CASTELLANI - P. FRANCESCHI - U. MONTANARI

LABELED EVENT STRUCTURES: A MODEL FOR OBSERVABLE CONCURRENCY

Computer Science Department, Università di Pisa

ETS/PISA

Labeled Event Structures: A Model for Observable Concurrency

by Ilaria Castellani, Paola Franceschi and Ugo Montanari

Computer Science Department, University of Pisa, Italy.

Abstract

The paper describes a model representing a nondeterministic, finit concurrent computation as a partial order of events. Each event is labele with a character, showing the result of the measurement of some sequentia observer. Many sequential observers constitute a concurrent observer, an an axiom states that two concurrent events cannot be seen by the sam sequential observer. A second label assigned to every event allows to se a nondeterministic computation as a tree of partial orders, each repre senting a deterministic computation.

The model is introduced as an extension of Nielsen, Plotkin an Winskel "event structures" and a necessary and sufficient condition i given which guarantees that a labeled event structure is simple, i.e. tha a corresponding (unextended) event structure exists. The problem of repre senting the external behaviour of a system with a labeled event structur is considered, and a simplification relation is defined. This relation i proved confluent and finite terminating, and thus a minimal element exist for every equivalence class.

The above "observation equivalence" is then compared with Milner's and, after defining in a straightforward way the synchronization tree cor responding to a labeled event structure, it is shown that our equivalen relation is finer than Milner's, but coincides with it in the sequenti case.

Finally, the extra expressive power of our model with respect Nielsen, Plotkin and Winskel's is discussed, and an example is giv where two observationally equivalent, simple labeled event structures c be both simplified only to a nonsimple labeled event structure. Thus o extension can be seen from this point of view as a convenient completi operation.

!. Introduction

A number of formal models have been recently introduced to represent ›ondeterministic concurrent computations /1-12, 14/. In many cases, however, 'he concurrency aspects are not considered primitive and can in fact be con- ·eptually reduced to nondeterminism /2,6,9/.

Let us take Milner asynchronous CCS as an example /2/. The behaviour of : system is described as an equivalence class of expressions, congruent with ·espect to a number of operators and characterized (at least in the absence ›f recursion and value passing) by a set of axioms or "laws". The expres- ;ions themselves are built up with the operator symbols, which represent language constructs. The composition operator (expressing interaction be- ·ween concurrently active components) is not primitive, in the sense that it ·an be completely eliminated from an expression (at least in the absence of ·ecursion) using a law which describes it in terms of interleaving between the components. Thus concurrency is modelled as a way of composing systems, ›ut is not observable within the behaviour of a single system. A well-known ›ethod of explicit representation of concurrency is by means of partial ›rders /1,4,10,14/.

According to other proposed models /5,7,8/ the nondeterministic be- ›aviour of a system can be represented as a set of descriptions (vectors of :trings, partial orders) each corresponding to a deterministic computation. 'his assumption usually leads to regard as equivalent also two systems which ›ave the same set of deterministic computations, but "decide" which one to ·hoose at a different stage and/or with a different mechanism. The ability ›f describing such a difference is essential in many systems, e.g. in ›odeling deadlock, and thus we prefer, as Milner does, to represent the ›ehaviour of a nondeterministic system as a tree whose paths are deter- ›inistic computations.

The work by Nielsen, Plotkin and Winskel (NPW) /1,14/ tries to combine the approaches to semantics of Petri and Scott. However, both Petri's ›onsequential processes /4/ and NPW's event structures, while having an ·xplicit representation of concurrency, do not possess an "observation" :tructure which supports operators as composition or restriction /2/, or ›otions as observability or full abstractness.

If we put together the considerations above, a candidate model should ·onsist in our view of a tree of partial orders, together with a "concur- ·ent" observer. In this paper we present a possible model, valid for the 'inite case.

2. Event structures

In this section we shortly introduce some concepts about event str tures /1,14/. Event structures can be considered as derived from Petri ne where conditions are omitted and an explicit relation describes confli Furthermore, event structures have neither cycles nor backward conflic since they are supposed to represent computations rather than systems.

Formally, an event structure (ES) is a triple $S = (E, \leqslant, \#)$, where

- E is a set of events;
- $\leqslant$ is a partial ordering over E, called the causality relation;
- $\#$ is a symmetrical and irriflexive relation over E called confl or mutual exclusion relation;
- relation $\#$ is inherited, namely the following axiom holds

(2.1) $\forall\, e_1, e_1', e_2 \in E,\ e_1 \# e_2$ and $e_1 \leqslant e_1'$ implies $e_1' \# e_2$.

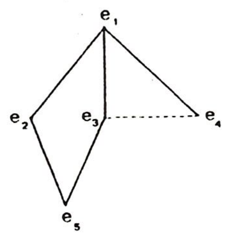


Fig. 1

In Fig. 1 we see an event structure where the partial order $(E, \leqslant$ represented as usual with its Hasse diagram, while mutual exclusio explicitly marked (dotted line) only within those pairs of events w cannot inherit it. Thus we have also $e_1 \leqslant e_5$ and $e_5 \# e_4$. Here and in follows the partial orders representing the flow of time grow downwards.

A set $E' \subseteq E$ of events is left closed iff

(2.2) $\forall\, e \in E, \forall\, e' \in E',\ e \leqslant e'$ implies $e \in E'$.

A set $E' \subseteq E$ is conflict free if no pairs of E' are in conflict. Se events which are left closed and conflict free are partial orders r senting deterministic computations.

Given an event structure S, let $\mathscr{L}(S)$ be the partial order of lef c sed and conflict free subsets of E, ordered by inclusion. Let us call it the state domain of S.

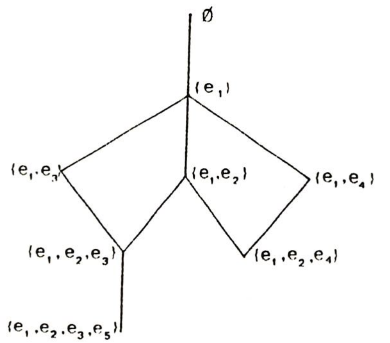


Fig. 2

In Fig. 2 we see the state domain of the ES in Fig. 1.

Let us consider now the maximal elements of the state domain of some event structure S and let us call them maximal deterministic computations (mdc). They specify a covering of S which has the form of a tree of partial orders. In fact, given a mdc x and an event e, if e belongs to x, then all its predecessors belong to x.

In our example, there are two mdc: $\{ e_1, e_2, e_3, e_5 \}$ and $\{ e_1, e_2, e_4 \}$ (see Fig. 1). The meaning of the intersection $\{ e_1, e_2 \}$ is the maximal computation which can be carried on without deciding between the two mdc.

3. Labeled event structures

As discussed in the introduction, we want to include in our model the notion of a "concurrent" observer. Here a concurrent observer is simply a set of sequential observers, which, in turn, are disjoint sets of characters. The events of our structures will be labeled with characters and thus, implicitly, with sequential observers. The underlying idea is that a sequential observer represents the point in space where the system is observed. Thus a system may have many such points, and at every point the observation may yield one among a set of possible values.

A further innovation with respect to NPW's event structures is that we derive the conflict relation from a more primitive notion. We define a set A of alternatives which represent the possible final states of a

decision process, and we label every event with a non empty set of alter- natives: When we observe an event, the (hidden) decision process must st- ill allow a set of alternatives which is smaller than or equal to the la- bel of the event. Clearly, two events are in conflict iff they have n common alternative. Alternatives are related to maximal deterministic com- putations introduced in the last section.

Formally, a Labeled Event Structure (LES) is a sextuple

$$S = (A, O, E, \leqslant, al, ch)$$

where:

$$A = \{ a_1, \ldots, a_n \}$$

is a non empty set of alternatives;

$$O = \{ C_1, \ldots, C_m \}$$

is a concurrent observer, namely a non empty set of sequential observers

Sequential observers

$$C_i = \{ c_{i1}, \ldots, c_{ik_i} \} \qquad i=1,\ldots m$$

are disjoint non empty sets of characters. Let $\bar{C} = \bigcup_{i=1}^{m} C_i$ be the alphabet o all characters;

$$(E, \leqslant)$$

is a partial order of events;

$$al: E \rightarrow 2^A - \emptyset$$

associates to every event a non empty set of alternatives; and

$$ch: E \rightarrow \bar{C}$$

associates to every event a character.

Two events e_1, e_2 are causally related iff $e_1 \leqslant e_2$ or $e_2 \leqslant e_1$; are in conflict or are mutually exclusive: $e_1 \# e_2$, iff $al(e_1) \cap al(e_2) = \emptyset$; ar concurrent (we write e_1 co e_2 as in /4/) if they are neither causally re lated nor in conflict. Two characters are concurrent (c_1 co c_2) if the belong to different sequential observers.

A LES must satisfy the following two axioms.

(3.1) $e_1 \leqslant e_2$ implies $al(e_1) \supseteq al(e_2)$

(3.2) e_1 co e_2 implies $ch(e_1)$ co $ch(e_2)$

Axiom (3.1) means that the flow of time restricts the choice amo alternatives; and axiom (3.2) states that two concurrent events cannot seen by the same sequential observer, namely they occur at differe points in space. Notice that the three relations between events, namely causal relation, mutual exclusion and concurrency, are disjoint and exhaustive.

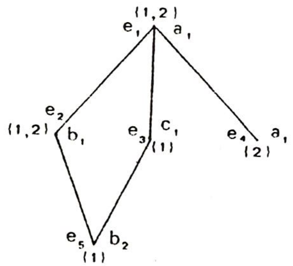


Fig. 3

As an example, in Fig. 3 we see a LES S with $A=\{1,2\}$ and $O=\{\{a_1\},\{b_1,b_2\},\{c_1\}\}$. It corresponds to the ES in Fig. 1, if we associate alternative 1 to mdc $\{e_1,e_2,e_3,e_5\}$ and alternative 2 to mdc $\{e_1,e_2,e_4\}$. A detailed comparison between LESes and ESes will be carried on in section 6.

We can now give some further definitions. We first introduce two functions

$$\text{pred},\text{succ}: E \rightarrow 2^E$$

$$\text{pred}(e)=\{e' \mid e' \leqslant e,\ e \neq e'\},\quad \text{succ}(e)=\{e' \mid e \leqslant e',\ e \neq e'\}$$

which associate to every event of a LES the (possibly empty) sets of its predecessors and successors in the partial order. Furthermore, we define another function

$$\text{comp}: A \rightarrow 2^E$$
$$\text{comp}(a) = \{e \mid a \in al(e)\}$$

which associates to every final state of the "decision process" the set of events observable in that state. Notice that, due to axiom (3.1), the range of comp contains only left closed subsets of E.

The rationale for introducing the alternatives in our LESes is to make as explicit as possible the global structure of the model. In a similar way, while Petri nonsequential processes /4/ are defined in terms of a relation F expressing <u>immediate</u> causal dependency, and partial order F^* is introduced later, NPW's event structures (and also our LESes) are defined directly in terms of $\leqslant$. We want to pursue this way further, with

the purpose of making operations among LESes (a subject not treated in this paper) as simple and natural as possible.

However, the above point of view tends to collide with the goal of representing the behaviour of a system as abstractly as possible. In our case, we have the choice to consider alternatives as either directly observable, or unobservable. The first approach is pursued in /13/ and may be useful to model languages where many composition operations exist with different rules for handling synchronization and global nondeterminism. There a notion of "firing and decision tree" is introduced, and two LESes result equivalent (i.e. with the same tree) iff they are isomorphic. Here we follow the second approach. For instance, it is quite reasonable that two alternatives a_1 and a_2, such that $\text{comp}(a_1) = \text{comp}(a_2)$, are merged. In the next section we give our definition of "observational equivalence" and compare it with Milner's /2/.

4. <u>The abstraction homomorphism</u>

Here we want to define a notion of simplification on LESes. Given two LESes $S_1=(A_1,O,E_1,\leqslant_1,al_1,ch_1)$ and $S_2=(A_2,O,E_2,\leqslant_2,al_2,ch_2)$, a pair of surjective functions (h_A,h_E)

$$h_A: A_1 \rightarrow A_2,\quad h_E: E_1 \rightarrow E_2$$

define an <u>abstraction homomorphism</u> between S_1 and S_2 iff the following axioms hold.

(4.1) $h_E(\text{pred}_1(e)) = \text{pred}_2(h_E(e))$

(4.2) $h_A(al_1(e)) = al_2(h_E(e))$

(4.3) $h_E(\text{comp}_1(a)) = \text{comp}_2(h_A(a))$

(4.4) $ch_1(e) = ch_2(h_E(e))$

If h_A and h_E are also injective, they define an <u>isomorphism</u>.

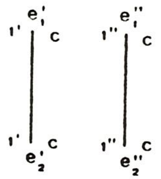
Notice that our axioms imply the properties

$$h_E(\text{succ}_1(e)) = \text{succ}_2(h_E(e))$$
$$e' \leqslant_1 e'' \text{ implies } h(e') \leqslant_2 h(e'')$$

but not the property

$$e' \leqslant_1 e'' \text{ iff } h(e') \leqslant_2 h(e'')$$

as shown by the example in Fig. 4.



$h_A(1')=h_A(1'')=1$

$h_E(e_1')=h_E(e_1'')=e_1$

$h_E(e_2')=h_E(e_2'')=e_2$

Fig. 4

In fact $h(e_1') \leq h(e_2'')$ but $e_1' \not\leq e_2''$. Notice also that abstraction homomorphism is defined only between LESes with the same concurrent observer.

We can informally justify our definition as follows. If h_A and h_E are also injective, then S_2 is simply obtained from S_1 by renaming alternatives and events. Otherwise some alternatives (and events) are identified. It is easy to see that

$$h_E(e_1)=h_E(e_2),\ e_1 \neq e_2 \text{ implies } e_1 \# e_2$$

i.e. two events may be identified only if in conflict. In fact, two causally related events cannot be identified due to axiom (4.1), while two concurrent events must be labeled with different characters for (3.2) and thus incur in axiom (4.4). If two mutually exclusive events are identified, they are, in a sense, indistinguishable, since they are labeled with the same character and their sets of predecessors, successors and alternatives are (recursively) indistinguishable.

Given two LESes S_1 and S_2, we write S_1 iso S_2 iff there exists an isomorphism between them. Of course iso is an equivalence relation and we designate with [S] the class of all LESes isomorphic to S. Two different such classes are in the irreflexive relation <u>simp</u>

$$[S_1] \text{ simp } [S_2]$$

iff there exists an abstraction homomorphism between S_1 and S_2.

Let now $\simeq$ be the reflexive, symmetric and transitive closure of simp. We also write $S_1 \simeq S_2$ iff $[S_1] \simeq [S_2]$. Relation $\simeq$ is our <u>observational equivalence</u> and its equivalence classes represent the observable behaviours of systems in our model. An advantage of our approach is that, at least in the finite case, every equivalence class has here a natural representative, i.e. a <u>minimal</u>, or <u>terminal</u> element.

<u>Theorem 4.1</u>. Relation simp is <u>confluent</u>, namely

(4.5) $x \text{ simp}^* y_1$ <u>and</u> $x \text{ simp}^* y_2$ <u>implies</u> <u>there exists</u> z <u>such that</u> $y_1 \text{ simp}^* z$ <u>and</u> $y_2 \text{ simp}^* z$.

<u>Proof</u> Notice that [S] simp* $[S_1]$ iff there exists an homomorphism between S and S_1. Let $x=[S]$, $y_1=[S_1]$, $A_1=h^1_A(A)$, $E_1=h^1_E(E)$, $y_2=[S_2]$, $A_2=h^2_A(A)$ and $E_2=h^2_E(E)$. Here (A,E,O,pred,al,comp,ch), $(A_1, E_1, O, pred_1, al_1, comp_1, ch_1)$ and $(A_2, E_2, O, pred_2, al_2, comp_2, ch_2)$ can be considered as homomorphic many-sorted algebras. As A_1,A_2 and E_1,E_2 we choose the equivalence classes of A and E induced by functions h^1_A, h^2_A and h^1_E, h^2_E. Thus A_1,E_1 and A_2,E_2 are also congruences with respect to operations $pred_1, al_1, comp_1, ch_1$ and $pred_2, al_2, comp_2, ch_2$. Let A',E' be the finest congruence coarser than both A_1,E_1 and A_2,E_2 and let pred',al',comp' and ch' be its operations. It is easy to see that pred' represents a partial order $\leq'$ which satisfies together with al',comp' and ch' axioms (3.1) and (3.2), since this is true of both $pred_1$, $\leq_1, al_1, comp_1, ch_1$ and $pred_2$, $\leq_2$, al_2, $comp_2, ch_2$. Thus z= [S'] , with S'=(A',O,E', $\leq'$,al',ch'). Since A', E' is coarser than both A_1,E_1 and A_2,E_2, there exist two homomorphisms (g^1_A, g^1_E), and (g^2_A, g^2_E) from S_1 to S' and from S_2 to S'.
■

<u>Corollary 4.1</u>. In the finite case every class of the equivalence relation $\simeq$ contains a unique simp-reduced element, namely there exists in every class exactly one element t such that no t' exists in the class with t simp t'. Furthermore, for every element x in the class, we have x simp t or x=t.

<u>Proof</u> Relation simp is finite terminating, namely it has only finite forward chains, since if $[S_1]$ simp $[S_2]$ we have $|A_2| + |E_2| \quad |A_1| + |E_1|$. Furthermore, simp is confluent for the previous theorem. Thus every class contains a unique reduced element t and $x \simeq t$ implies $x \text{ simp}^* t$ /15/ and thus x simp t or x=t since simp is transitively closed.
■

›. Comparison with synchronization trees

In this section, we try to relate our work to Milner work on synchro- ıization trees. We associate to every LES a synchronization tree and we ›rove that observationally equivalent LESes have observationally equiva- ent (according to Milner) synchronization trees. The converse is not true n general, but is true for LESes with a unique sequential observer.

We remind that a synchronization tree /2/ (ST) of sort L is a rooted, ınordered, finitely branching tree, each of whose arcs is labeled by a ıember of $L \cup \{\tau\}$. For every $\lambda \in L$, a binary relation $\xrightarrow{\lambda}$ is defined ›etween two STs: $t \xrightarrow{\lambda} t'$ iff t has a branch λ t'. Furthermore, we have $t \xrightarrow{\tau} t'$ ff t has a branch τ t'; and $t \overset{\tau}{\Rightarrow} t'$ iff there exists an n such that $t \xrightarrow{\tau^n} t'$.

Finally $t \overset{s}{\Rightarrow} t'$ where $s=\lambda_1 ... \lambda_n \in L^*$, iff $t \overset{\tau}{\Rightarrow} t_1 \xrightarrow{\lambda_1} t_1' \overset{\tau}{\Rightarrow} t_2, ..., t_n \xrightarrow{\lambda_n}$ $t_n' \overset{\tau}{\Rightarrow} t'$.

Similarly, for every $c \in \overline{C}$, we define a binary relation $\xrightarrow{c}$ between wo LESes $S_1 = (A_1, O, E_1, \leqslant_1, al_1, ch_1)$ and $S_2 = (A_2, O, E_2, \leqslant_2, al_2, ch_2)$ as fol- ows.

'e have $S_1 \xrightarrow{c} S_2$ if

) $A_1 = A_2$;

i) there exists an event $e \in E_1$, which is a minimal of partial order $\leqslant_1$, such that $al_1(e)=A_1$ and $ch_1(e)=c$;

ii) $E_2=E_1-\{e\}$ and $\leqslant_2, al_2, ch_2$ are the restrictions to E_2 of $\leqslant_1$, al_1, ch_1.

We say that S_2 is obtained from S_1 by firing event e. Notice that iven S_1 and c there is at most one S_2 such that $S_1 \xrightarrow{c} S_2$, since two dis- inct events cannot satisfy ii) due to axiom (3.2).

Furthermore, a relation $\xrightarrow{\tau}$ holds between two LESes S_1 and S_2 if

) $A_2=A_1-\{a\}$

i) $E_2=E_1-\{e \mid e \in E_1, al_1(e)=\{a\}\}$ and $\leqslant_2, al_2, ch_2$ are the restrictions to A_2 and E_2 of $\leqslant_1, al_1, ch_1$.

nformally, if $S_1 \xrightarrow{\tau} S_2$, then S_2 is obtained by excluding one alternative nd by erasing all events which appear only in this alternative.

otice that a LES is reduced w.r.t. $\xrightarrow{\tau}$ iff it is deterministic, namely if ts alternative set A is a singleton; is reduced w.r.t. $\xrightarrow{\tau}$ and $\xrightarrow{c}$ for all c ff A is a singleton and E is empty.

The ST t(S) of sort $\overline{C}$ associated to a LES S is defined recursively as follows

(5.1) $t(S) = \{ ct' \mid \exists S', S \xrightarrow{c} S' \text{ and } t'=t(S')\} \cup \{\tau t' \mid \exists S', S \xrightarrow{\tau} S' \text{ and } t'=t(S')\}$

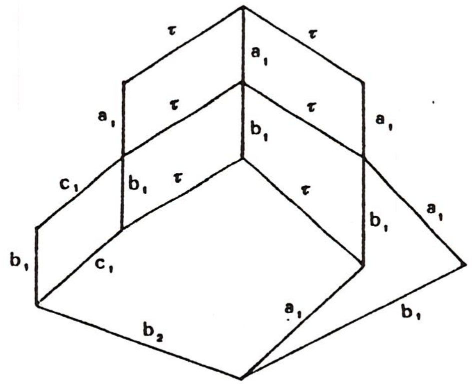


Fig. 5

In Fig. 5 we see the factorized synchronization tree of the LES in Fig.3.

Milner defines his observation equivalence $\approx$ as follows.

a) $p \approx_0 q$ is always true;

b) $p \approx_{k+1} q$ iff $\forall s \in L^*$

(i) if $p \overset{s}{\Rightarrow} p'$ then for some q', $q \overset{s}{\Rightarrow} q'$ and $p' \approx_k q'$

(ii) if $q \overset{s}{\Rightarrow} q'$ then for some p', $p \overset{s}{\Rightarrow} p'$ and $p' \approx_k q'$;

c) $p \approx q$ iff $\forall k \geqslant 0$ $p \approx_k q$.

We can prove the following theorem.

<u>Theorem 5.1</u> Given two LESes S_1 and S_2, let $t(S_1)$ and $t(S_2)$ be their synchronization trees. $S_1 \simeq S_2$ implies $t(S_1) \approx t(S_2)$.

<u>Proof</u> We first prove that $[S_1]$ simp $[S_2]$ or S_1 iso S_2 implies $t(S_1) \approx t(S_2)$.

We introduce a new relation R between synchronization trees. We have $t(S_1)$ R $t(S_2)$ iff an homomorphism exists between S_1 and S_2, namely iff

$[S_1]$ simp $[S_2]$ or S_1 iso S_2. Thus we want to prove that R is finer than $\approx$. To do this we prove that R satisfies the equation

pRq implies $\forall s \in L^*$

(i) if p $\overset{s}{\Rightarrow}$p' then for some q', q $\overset{s}{\Rightarrow}$q' and p'R q'

(ii) if q $\overset{s}{\Rightarrow}$q' then for some p', p $\overset{s}{\Rightarrow}$p' and p'R q'.

In fact it is easy to see that $\approx$ is the maximal solution of the above equation.

We further simplify our goal as follows

pRq implies $\forall \alpha \in \bar{C} \cup \tau$

(i) if p $\overset{\alpha}{\rightarrow}$ p' then for some q',q $\overset{\alpha}{\Rightarrow}$q' and p'R q'

(ii) if q $\overset{\alpha}{\rightarrow}$ q' then for some p',p $\overset{\alpha}{\Rightarrow}$p' and p'R q'.

To prove the equivalence it is enough to decompose relation $\overset{s}{\Rightarrow}$ into elementary $\overset{\alpha}{\rightarrow}$ moves, and to apply induction. Now let $p=t(S_1)$ and $q=t(S_2)$ and let (h_A,h_E) be an abstraction homomorphism between S_1 and S_2. Let us prove the (i) part. Due to (5.1), if p $\overset{\alpha}{\rightarrow}$ p' then also $S_1 \overset{\alpha}{\rightarrow} S_1'$ with $p'=t(S_1')$. If $S_1 \overset{c}{\rightarrow} S_1'$ applying (h_A,h_E) to S_1' we clearly obtain a LES S_2' such that $S_2 \overset{c}{\rightarrow} S_2'$. If $S_1 \overset{\tau}{\rightarrow} S_1'$, let a be the excluded alternative. If there exists another alternative a' such that $h_A(a)=h_A(a')$, then (h_A,h_E) maps S_1' into S_2, and thus we choose $S_2'=S_2$. If such an a' does not exists we obtain S_2' by excluding alternative $h_A(a)$. In all cases we have $q'=t(S_2')$, q $\overset{\alpha}{\Rightarrow}$q' and p'R q'.

To prove (ii), if q $\overset{\alpha}{\rightarrow}$ q' then also $S_2 \overset{\alpha}{\rightarrow} S_2'$ with $q'=t(S_2')$. Let us distinguish again the case $\alpha = c \in \bar{C}$ and $\alpha = \tau$. In the first case, let e_2 be the event fired, and let $e^1_1, e^2_1, \ldots, e^n_1$ be the events of E_1 such that $h_E(e^i_1)=e_2, i=1,\ldots n$. It is easy to see that if we take an event in the set, say e^1_1, we exclude from S_1 all alternatives in $A_1 - al_1(e^1_1)$, and then we fire e^1_1, we get a LES S_1' which is mapped by (h_A,h_E) into S_2'. Thus $p'=t(S_1')$ with p $\overset{c}{\Rightarrow}$p' and p'R q'. In the second case, S_2' can be obtained from S_2 by excluding an alternative a_2. Let $a^1_1, a^2_1, \ldots, a^n_1$ be the alternatives of A_1 such that $h_A(a^i_1)=a_2, i=1,\ldots,n$. It is easy to see that if we exclude from S_1 all these alternatives, we get a LES S_1' which is mapped by (h_A,h_E) into S_2'. Thus $p'=t(S_1')$ with p $\overset{\tau}{\Rightarrow}$q' and p'R q'.

Finally, to prove that $S_1 \simeq S_2$ implies $t(S_1) \approx t(S_2)$, let $[S]$ be the minimal, simp-reduced element introduced by Corollary 4.1. Applying twice what we just proved, we get $t(S_1) \approx t(S)$ and $t(S_2) \approx t(S)$. Thus $t(S_1) \approx t(S_2)$, since $\approx$ is an equivalence relation. ■

What can be said about the converse of theorem 5.1? It is false, and we have a simple counterexample.

Fig. 6

In Fig. 6 we have two simp-reduced LESes, both with $A=\{1,2,3\}$ and $O=\{\{c_1\},\{c_2\}\}$. Thus they are not equivalent, but have equivalent synchronization trees, shown in Fig. 7.

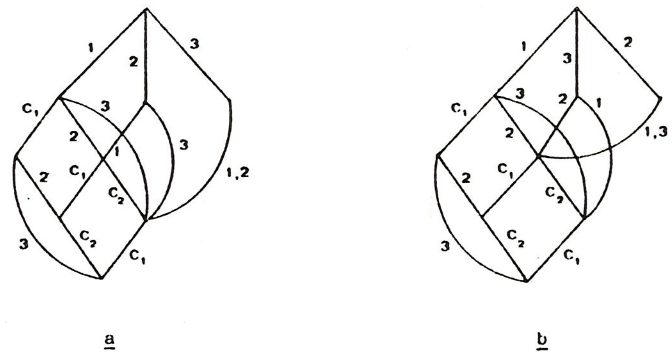


Fig. 7

To help the reader, the τ-arcs are labeled with the excluded alternative. To show that the two trees are equivalent, it is enough to consider the first τ^n-move, as shown in Table 1.

τ^n-move for Fig. 7a	τ^n-move for Fig. 7b
1	1
2	3
3,13,23,31,32	13,31
12,21	2,12,21,23,32

Table 1

If we restrict ourselves to the sequential case, however, the two equivalences coincide, as shown by the following theorem.

Theorem 5.2 Given two LESes S_1 and S_2, both with the concurrent observer consisting of a single sequential observer, let $t(S_1)$ and $t(S_2)$ be their synchronization trees. We have $S_1 \approx S_2$ iff $t(S_1) \approx t(S_2)$.

Proof The "only if" part is a special case of theorem 5.1. For the "if" part we assume $t(S_1) \approx t(S_2)$, with S_2 simp-reduced, and we prove

(5.3) pRq if $\forall \alpha \in \bar{C} \cup \tau$
(i) if $p \overset{\alpha}{\Rightarrow} p'$ then for some q', $q \overset{\alpha}{\Rightarrow} q'$ and p'R q'
(ii) if $q \overset{\alpha}{\Rightarrow} q'$ then for some p', $p \overset{\alpha}{\Rightarrow} p'$ and p'R q'

where as usual $p = t(S_1)$, $q = t(S_2)$ and $t(S_1)$ R $t(S_2)$ iff ($[S_1]$ simp $[S_2]$ or S_1 iso S_2). In fact, by Corollary (4.1) if S_2 is not simp-reduced, there exists a simp-reduced LES $\bar{S}_2$ such that S_2 simp $\bar{S}_2$ and thus $t(S_2) \approx t(\bar{S}_2)$

$\approx t(S_1)$ by theorem (5.1). Therefore S_2 simp $\bar{S}_2$ and S_1 simp $\bar{S}_2$ implies S_1 $\approx S_2$ also in this case.

We can partition "sequential" LESes into three classes: a) ready LESes, where exactly one event e exists which is the minimal of $\le$ and al(e) =A; b) fair LESes, where $\le$ has a set of minima of more than one element, and their sets of alternatives are disjoint and exhaustive; and c) unfair LESes, where the set(s) of alternatives of the minimum (minima) of $\le$ is (are disjoint but) not exhaustive. No other case exists, since there are no pairs of concurrent events. It is easy to see that if $t(S_1) \approx t(S_2)$ then S_1 and S_2 must be either both unfair or both not unfair.

Given S_1, let S_1' be a ready LES obtained from S_1 by excluding some alternatives (if S_1 is fair or unfair) or by firing the "root" event and by excluding some alternatives (if S_1 is ready). If we let $p' = t(S_1')$, we

clearly have $p \overset{\alpha}{\Rightarrow} p'$ and thus our hypothesis gives us a tree q', such that $q' = t(S_2')$, and an abstraction homomorphism (h_A', h_E'). This is true for all such S_1'. All such subLESes of S_1 are disjoint, and their union gives back S_1 if S_1 is fair. Thus the union of the corresponding homomorphisms gives the mapping (h_A, h_E) between S_1 and S_2 we are looking for. If S_1 is unfair, then also S_2 is unfair, and to get h_A we extend the union above by mapping all the uncovered alternatives in A_1 into some uncovered alternative in A_2. Such alternative is unique. In fact, if S_2 had more than one alternative a such that comp(a)= $\emptyset$, it would not be simp-reduced. If S_1 is ready, we proceed as above, but we further complete h_E by mapping the root event of S_1 into the root event of S_2.

It is rather clear that (h_A, h_E) constructed as above is an abstraction homomorphism for all properties, except surjectivity. To prove it, we can use part (ii) of our hypothesis to construct a pair $(\bar{h}_A, \bar{h}_E)$ mapping S_2 (possibly not surjectively) into S_1. Composing $(\bar{h}_A, \bar{h}_E)$ with (h_A, h_E) above we get a mapping of S_2 into S_2 which is not surjective if (h_A, h_E) is not surjective. If we throw away the elements of A_2 and E_2 which are not reached, we get a new LES $\bar{S}_2$ such that S_2 simp $\bar{S}_2$. Thus if (h_A, h_E) is not surjective, S_2 is not simp-reduced, against the hypothesis.

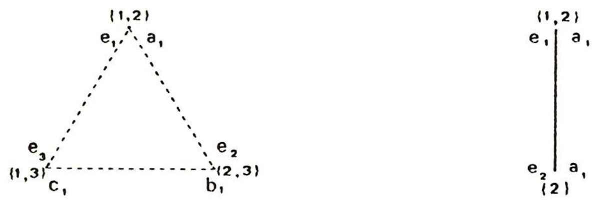
■

6. Comparison with event structures

Comparing the definition of LES given in Section 3 with the definition of ES given in Section 2, it should be clear the convenience of introducing a (concurrent) observer and of labeling events with characters. The introduction of alternatives might instead appear not sufficiently motivated. Thus we introduce in this section the class of simple LESes, as those LESes where the labeling function al can be reconstructed from the relation # of mutual exclusion. Simple LESes are thus the direct counterpart of ESes. Then we give a compact characterization of simple LESes. Finally we show two simple LESes which can be both reduced to the same nonsimple LES and thus are observationally equivalent, but which cann[illegible] be both reduced to any simple LES. Therefore the introduction of nonsimple

LESes may be considered from this point of view a convenient completion construction.

Given a LES $S = (A, O, E, \leqslant, al, ch)$, let $\#$ be the mutual exclusion relation and <u>comp</u> be the function introduced in section 3, and let S' be the event structure defined as $S' = (E, \leqslant, \#)$. The LES S is <u>simple</u> if the set M of the maximal deterministic computations of S' coincides with the range P of function comp.



a b

Fig. 8

For example, Fig. 3 represents a simple LES, while Fig. 8 a and b represent two nonsimple LESes. Dotted lines represent alternatives. In fact, Fig. 8 a has one mdc, i.e. $\{e_1, e_2, e_3\}$ and three elements in P, i.e. $\{e_1, e_3\}$, $\{e_1, e_2\}$ and $\{e_2, e_3\}$. Similarly, Fig. 8 b has one mdc $\{e_1, e_2\}$ while P is $\{\{e_1, e_2\}, \{e_1\}\}$. Note that in Fig. 8 a we have three events which are pairwise concurrent but not, so to say, globally concurrent. In Fig. 8 b we have the possibility, in alternative 1, of blocking a computation at an intermediate stage. Those are exactly the two facts one must exclude to get a simple LES.

<u>Theorem 6.1</u> Let R be the union of relations $\leqslant$, $\geqslant$ and co. Thus R is the complement of $\#$. A LES S is simple iff

i) given n events $e_1, e_2, \dots, e_n$, if $e_i\ R\ e_j$ $i,j=1,\dots,n$, then there exists an alternative a such that $e_1, e_2, \dots, e_n \in comp(a)$;

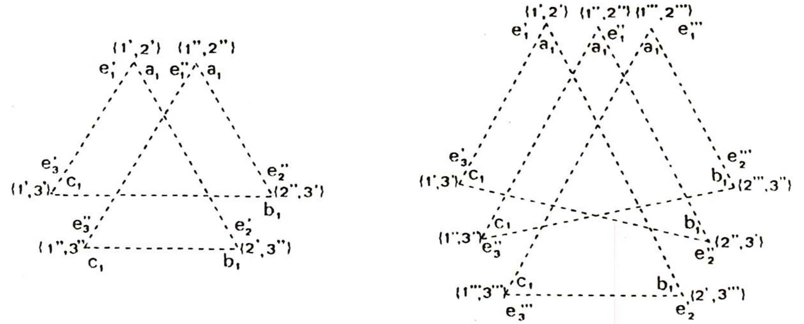
ii) do not exist two alternatives a_1 and a_2 such that $comp(a_1) \subset comp(a_2)$ (the inclusion is strict).

<u>Proof</u> <u>Only if part</u> It is easy to see that mdc's are the cliques (maximal compatibily classes) of R. Thus if $e_1, e_2, \dots, e_n$ are pairwise compatible,

they must appear together in at least a clique. Furthermore, $comp(a_1)$ above cannot be a clique, since it is not maximal.

<u>If part</u> Let us assume by absurd that a mdc m does not belong to P. However, its elements are all R-compatible and thus by i) there exists an alternative a with $m \subset comp(a)$. But comp(a) is a compatibility class of R. Thus m is not maximal. Conversely, let m'=comp(a) and let m' not be a mdc. However, m' is a compatibility class, and thus is not maximal. Let m be a maximal compatibility class, $m' \subset m$. Thus by i) there exists an alternative a' with $comp(a') \supseteq m \supset m' = comp(a)$ against ii). ■

The two nonsimple LESes represented in Fig. 8 can be both useful. In fact, we may want to define the operation of <u>observer cancellation</u> similar to Milner's restriction operator: The events seen by the cancelled observer are simply erased. If we restrict the LES in Fig. 3, a simple LES, by cancelling the sequential observers $\{b_1, b_2\}$ and $\{c_1\}$ we obtain the nonsimple LES in Fig. 8b. Similarly, the class of simple LESes is not closed with respect to the operation of abstraction homomorphism defined in section 4.



a b

Fig. 9

In fact, in Fig. 9 we see two simple LESes which can be simp-reduced to the LES in Fig. 8 a, which is nonsimple. Furthermore, the two LESes in Fig. 9, which are thus observationally equivalent, are both simp-reduced within the class of simple LESes, and cannot be further simplified.

References

/1/ M. Nielsen, G. Plotkin, G. Winskel, Petri Nets Event Structures and Domains, Semantics of Concurrent Computation, Springer LNCS N°70, 1979, pp.266-284.

/2/ R. Milner, A Calculus of Communicating Systems, Springer LNCS N°92, 1980.

/3/ R. Milner, On Relating Synchrony and Asynchrony, Internal Report CSR-75-80, Edinburgh, December 1980.

/4/ C.A. Petri, Concurrency, Proc. Net Theory and Applications, Springer LNCS N°84, 1980, pp.251-260.

/5/ C.A.R. Hoare, A Model for Communicating Sequential Processes, Technical Monograph PRG-22, Oxford, June 1981.

/6/ M. Francez, C.A.R. Hoare, D.J. Lehmann and W.P. deRoever, Semantics of Nondeterminism, Concurrency and Communication, JCSS,19, 3, 1979.

/7/ J. Beauquier and M. Nivat, Applications of Formal Language Theory to Problems of Security and Synchronization, Formal Language Theory, Academic Press, 1980.

/8/ P.E. Lauer, P.R. Torrigiani and M.W. Shields, COSY, A System Specification Language Based on Paths and Processes, Acta Informatica 12, 109-158, 1979.

/9/ P. Folkjar and D. Bjorner, A Formal Model of a Generalized CSP-like language, Information Processing 80, North Holland, 1980, pp.95-99.

/10/ J. Winkowski, Behaviours of Concurrent Systems, TCS, 11 (1980) pp.39-60.

/11/ P. Darondeau, Processus Non-Séquentiels et Leur Observation en Univers Non-Centralisé; International Symposium on Programming, Springer, LNCS N°83, 1980, pp.92-107.

/12/ D.W. Mitzell, Verification and Design Aspects of True Concurrency, Proc. ACM POPL, Tucson, Jan. 23-25, 1978, pp.171-175.

/13/ U. Montanari, C. Simonelli, On Distinguishing Concurrency from Nondeterminism, Proc. Réseaux de Petri et Parallélisme, Colleville sur mer, May 1980.

/14/ G. Winskel, Events in Computation, PhD Thesis CST-10-80, Edinburgh, December 1980.

/15/ J.A. Goguen, How to Prove Algebraic Inductive Hypotheses Without Induction, SH Conference on Automated Deduction, Springer LNCS N°87, 1980, pp.356-373.