An Extensional Formulation of ST Bisimulation

Roberto Gorrieri[‡] Cosimo Laneve[†]

[‡]Dipartimento di Matematica, Università di Bologna
Piazza di Porta S. Donato 5, I-40100 Bologna, Italy
e-mail: gorrieri@dm.unibo.it

[†]Dipartimento di Informatica, Università di Pisa
Corso Italia 40, I-56125, Pisa, Italy
e-mail: laneve@di.unipi.it

Abstract

In this paper we characterize in an alternative, extensional way the non atomic equivalence based on the notion of ST bisimulation. This equivalence, originally proposed in [10] over Petri Nets, states that actions can be observed in the middle of their evolution and, additionally, that the individuality of each action can be unambiguously recovered from the observation of its subparts. The problem is studied over the very general model of stable event structures [19], on which also some further equivalence notions [8, 1, ?, 18], technically rather different although based on a similar intuition, are compared with and proved equivalent to the one we propose. The relevance of ST bisimulation equivalence emerges when considering an operator for action refinement.

1 Introduction

A taxonomy of the equivalences usually described in the literature can be represented through a two-dimensional tableau where the orthogonal axes denote, respectively, the various "degrees" of nondeterminism and causality which can be observed. The so-called *linear-time/branching-time* spectrum (see, e.g., [5, 7]) accounts for the nondeterministic aspect of system behaviour, while the *interleaving/truly concurrent* spectrum (see, e.g., [?]) describes the amount of causal dependencies that actions a system performs can manifest.

All these equivalences are based on the strong assumption that the execution of an action is an *atomic* activity which cannot be observed or tested in the middle of its evolution. As a matter of fact, this abstraction step from the actual behaviour of systems, which is behind almost all the theories of concurrency, produces the unrealistic consequence that actions do not take time. Therefore, a considerable amount of concurrency research has been recently devoted to cope with the description of the execution of actions, possibly generalizing the machinery developed for the atomic case. We remark that expressing time duration of actions turns out to be crucial for *action refinement* (see, e.g., [12] and the references therein): an action, having duration in time, can be considered as an abstraction of a whole system, thus introducing the possibility of relating descriptions of the same system at different levels of detail. Even if the introduction in concurrency theories of an operation of action refinement has not yet reached a completely satisfactory treatment, the relevance of this concept for software engineering cannot be underestimated. In this perspective, *non atomic* equivalences are gaining

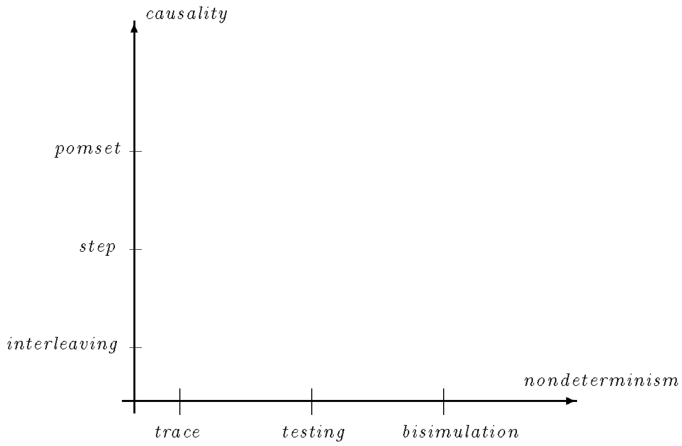

Figure 1: Traditional equivalences in tableau format.

more and more prominence as possible candidates for *congruences* (thus permitting "substitution of equals for equals") w.r.t. suitable refinement operations.

Hennessy [14] was probably the first who relaxed the atomicity assumption by permitting actions to be observed in the middle of their evolution. In particular, he suggested atomic actions composed of two phases, their *beginnings* and their *endings*. This proposal can be generalized to an arbitrary number of phases, yielding equivalences named split_n. Therefore, a third orthogonal direction can be added to the diagram in Figure 1, accounting for "degrees" of *granularity*, as illustrated in Figure 2.

In principle, we can study any combination of the ingredients on the three axes; just to mention a possible combination: testing-step-split_3. Once fixed the choice on the two-dimensional tableau, the interesting point is to discover if splitting actions in more and more phases produces more and more discriminating equivalences and if there exists a limit to this chain. The main result of [13, ?] is the so called *limit theorem*, which states that, for interleaving bisimulation semantics, such a limit does really exist and coincide with ST bisimulation equivalence. This equivalence was originally proposed by van Glabbeek and Vaandrager [10] on Petri Nets, by mimicking split equivalence. A state in the bisimulation relation is modelled by the set of those places (*Stellen*) that contain a token and of the currently firing transitions (*Transitionen*), hence the name ST.

These non atomic notions have been rephrased over event structures in [8], according to which the states involved in the definition of ST and split_2 bisimulations are the same, namely split_2 configurations. However, the two equivalences are different because ST bisimulation also exploits bijections between action phases which allow to connect the ending phases to those beginning phases occurred at the same time. ¿From a philosophical point of view, the definition of ST bisimulation in [8] is *not purely observational*, as it assumes a smart observer which is also able to build up a correspondence between action phases. The same criticism applies also to the so-called *history preserving* bisimulation [?, 6, 9]; however, for this equivalence an extensional, yet completely equivalent, formulation was given in [3, 4] as ordinary bisimulation on the model of *Causal Trees* [1].

Following a similar intuition, in this paper ST bisimulation is defined in an extensional way over stable event structures. Instead of changing the notion of bisimulation, we prefer to enrich slightly the information in the ending phases of the split_2 semantics: every ending phase has a backward pointer towards its own beginning phase. To be more precise, a number $\boldsymbol{k}$ is associated to each ending phase, telling that it refers to the *unique*

[1] Causal Trees are *Synchronization Trees* with a richer labelling: any arc, besides the action name, is labelled by the set of backward pointers to all the previous transitions which caused the action itself

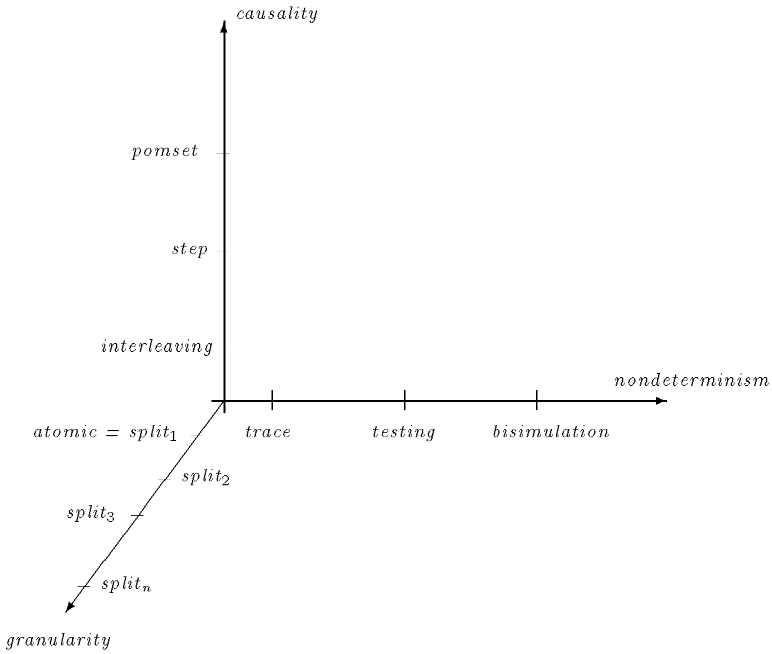

Figure 2: Equivalences for concurrent systems.

k^{th}-started action. This implies that the ST transition system derived from a stable event structure must be acyclic and cannot have a state with two ingoing transitions labelled by an initial phase. On the model of ST transition systems, ST bisimulation is simply ordinary bisimulation where the labels on the arcs express the set of the possible experiments. The advantage of our approach is twofold: from one hand, all the results that bisimulation theory has produced can be profitably exploited also in the present case; from the other, it is much clearer what is the “essence” of the ST idea.

Of course, we cannot simply limit ourselves to a purely descriptional enterprise: we have to prove that our notion really captures the same indentifications induced by the original proposal in [8]. Indeed, as the main result of the paper we provide such a proof. Not only we show that the two notions do coincide, but also investigate the relationship with other notions sharing similar intuitions: *refine* equivalence [1] and *maximality preserving* equivalence [?, 18], which we rephrase over stable event structures.

The relevance of our studies will be better appreciated in a forthcoming paper [?], where we show that interleaving ST bisimulation equivalence is the coarsest congruence, w.r.t. the operation of refining events with finite stable event structures, which is contained in interleaving bisimulation.

2 Stable Event Structures and Labelled Transition Systems

In this section we will briefly recall those notions which will be needed in the following. In particular, *stable event structures* are taken from [19]; we also add the notion of associated (atomic) labelled transition system; finally bisimulation semantics is taken from [15]. First of all, some notation facilities.

ω is the set of natural numbers; the set of its nonzero elements is denoted by ω^+.

$|S|$ stands for the *cardinality* of the set S;

$S_0 \times S_1$ stands for the (*cartesian*) *product* of the sets S_0 and S_1, i.e. $\{(s_0, s_1) \mid s_0 \in S_0 \wedge s_1 \in S_1\}$ (generalizable to the product of $S_0 \times S_1 \times \ldots \times S_{n-1}$);

π_0, π_1 stand respectively for the *first* and *second projection* of the set $S_0 \times S_1$, i.e. $\pi_i(S_0 \times S_1) = S_i$ (generalizable to $\pi_i(S_0 \times S_1 \times \ldots \times S_{n-1})$;

$f\lceil A$ denotes the restriction of function f to the set A (i.e., $f\lceil A(x) = f(x)$ whenever $x \in A$, undefined otherwise).

Finally, we denote with $\wp_f(S)$ the set of all the finite subsets of S, R^* stays for the reflexive and transitive closure of the relation R, whilst R^+ denotes the transitive closure of R.

According to [19], let us recall the definition of stable event structure.

Definition 2.1 A *stable event structure* $\mathcal{E}$, labelled on $\mathcal{M}$, (les, in the following) is a quadruple $\langle E, \, Con, \, \vdash, \, \ell \rangle$ where

- E is an enumerable set of events,
- Con is a nonempty subset of $\wp_f(E)$, called the *consistency predicate*, satisfying

$$X \in Con \textbf{ and } Y \subseteq X \quad \Rightarrow \quad Y \in Con \quad (\textit{downward closure});$$

- $\vdash \subseteq Con \times E$ is the *enabling relation* which satisfies:

$$\begin{array}{lll} \textit{i.} & X \vdash e \textbf{ and } X \subseteq Y \textbf{ and } Y \in Con \quad \Rightarrow \quad Y \vdash e & (\textit{monotonicity}), \\ \textit{ii.} & X \vdash e \textbf{ and } Y \vdash e \textbf{ and } X \cup Y \cup \{e\} \in Con \quad \Rightarrow \quad X \cap Y \vdash e & (\textit{stability}) \end{array}$$

- $\ell : E \to \mathcal{M}$ is the *labelling function.*

Let $\mathcal{ES}_{\mathcal{M}}$ be the domain of les's labelled over $\mathcal{M}$. ■

Event structures represent naively a concurrent system: the label of an event gives the action name whilst the event itself denotes the occurrence of that action in a particular run. The enabling relation $X \vdash e$ means that e can occur after the occurrence of the events in X; the consistency relation is exploited for describing which sets of events can actually be executed in a run. Les's will be ranged over by $\mathcal{E}, \mathcal{F}, \cdots$ and the components of $\mathcal{E}$ will be denoted by $E_{\mathcal{E}}$, $Con_{\mathcal{E}}$, $\vdash_{\mathcal{E}}$, $\ell_{\mathcal{E}}$, respectively. If $\mathcal{E}$ is a les modelling a concurrent system, a state s, which the system can evolve to, is identified by the set of events which occurred before reaching s. This idea is formalized through the notion of *configuration*. In order to define this notion, let us define a *proving sequence* w.r.t. $\mathcal{E}$ as a finite consistent sequence $e_1, e_2, \cdots e_n$ of distinct events satisfying:

$$\forall i \leq n \, . \, \exists F \subseteq \{e_1, e_2, \cdots e_{i-1}\} \, . \, F \vdash e_i$$

We say that such a sequence is a *proof* of e in X if $e_n = e$ and $\{e_1, e_2, \cdots e_n\} \subseteq X$.

Definition 2.2 A subset $C \subseteq E_{\mathcal{E}}$ is a *configuration* of $\mathcal{E}$, if:

- $\forall X \subseteq_{fin} C \, . \, X \in Con_{\mathcal{E}}$,
- every event e in C has a proof in C.

The set $\mathcal{C}_{\mathcal{E}}$ of configurations of the les $\mathcal{E}$ will be ranged over by $C, D, \cdots$ ■

In the following it will be useful to observe that any configuration C has associated a partial ordering relation on its events. Let C be a configuration of $\mathcal{E}$. For $e, e' \in C$, we define

$$e' \leq_C e \quad \Leftrightarrow \quad \forall D \in \mathcal{C}_{\mathcal{E}},\ e \in D \textbf{ and } D \subseteq C \ \Rightarrow\ e' \in D.$$

Given the partial ordering $(C, \leq_C)$, e is *maximal* in C if $e \in C$ and $\forall e' \in C.\, e \leq_C e' \ \Rightarrow e = e'$. We denote with $\boldsymbol{Max}(C)$ the set of maximal event in the configuration C. Note that for any $C' = C \cup \{e\} \in \mathcal{C}_{\mathcal{E}}$, the event e is maximal in C'. Finally, given an event $e \in C$, let e^*_C be the set of all the events less than or equal to e itself in C; formally:

$$e^*_C = \{e' \mid e' \leq_C e\}$$

Generally, two event structures are identified up to isomorphism, thus inducing a very concrete semantics for concurrent systems modelled by les's. However, this semantics is too informative (indeed, too intensional) because it distinguishes systems which should be reasonably identified. More abstract descriptions can be obtained by equating event structures according to some behavioural equivalence. A standard tool relies upon the notion of *bisimulation* [16], defined over labelled transition systems, which we rephrase over les's. To this aim we need to define, first of all, labelled transition systems and then how a labelled transition system can be associated to a les.

Definition 2.3 A *labelled transition system* (lts, for short) $\mathcal{G}$ is a quadruple $\langle \mathcal{S}, \mathcal{M}, \mathcal{T}, s_0 \rangle$ where $\mathcal{S}$ is a set of *states*, $\mathcal{M}$ is a set of *labels*, $\mathcal{T} = \{ \xrightarrow{\mu} \subseteq \mathcal{S} \times \mathcal{S} \mid \mu \in \mathcal{M} \}$ is the set of *transition relations* and $s_0 \in \mathcal{S}$ is the *initial state*. We will write $s \xrightarrow{\mu} s'$ instead of $\langle s, s' \rangle \in \xrightarrow{\mu}$. ■

Informally, a lts is a rooted directed graph with labelled arcs where nodes and arcs represent states and transitions, respectively. Every transition $s \xrightarrow{\mu} s'$ specifies that the system in the state s can transit to the state s' by performing the action μ. As usual, the notation $s_1 \xrightarrow{\mu(1)\cdot\mu(2)\cdots\mu(k-1)} s_k$ is a shortening for the sequence of transitions $s_1 \xrightarrow{\mu(1)} s_2 \xrightarrow{\mu(2)} \ldots \xrightarrow{\mu(k-1)} s_k$. When labels are not relevant, $s_1 \xrightarrow{\mu(1)\cdot\mu(2)\cdots\mu(k-1)} s_k$ may be represented more compactly as $s_1 \rightarrow^* s_k$; in such a case, we will say that s_k is *reachable* from s_1.

Definition 2.4 Let $\mathcal{E} \in \mathcal{ES}_{\mathcal{M}}$. The lts $\mathcal{G}(\mathcal{E}) = \langle \mathcal{C}_{\mathcal{E}}, \mathcal{M}, \mathcal{T}_{\mathcal{E}}, \emptyset \rangle$, where $\mathcal{T}_{\mathcal{E}} = \{ \xrightarrow{\mu} \subseteq (\mathcal{C}_{\mathcal{E}} \times \mathcal{C}_{\mathcal{E}}) \mid \mu \in \mathcal{M} \}$ is the set of transition relations defined as follows:

$$C \xrightarrow{\mu} C' \qquad \mathit{iff} \qquad \exists e\,.\,(\,C' = C \cup \{e\} \ \textbf{and}\ \ell_{\mathcal{E}}(e) = \mu\,)$$

is the *atomic* (or split$_1$) lts related to $\mathcal{E}$. ■

The lts $\mathcal{G}(\mathcal{E})$ has the relevant property of being acyclic. In the next section we will see that other transition systems, still acyclic, can be associated to a les.

Definition 2.5 Let $\mathcal{G} = \langle \mathcal{S}, \mathcal{M}, \mathcal{T}, s_0 \rangle$ and $\mathcal{G}' = \langle \mathcal{S}', \mathcal{M}, \mathcal{T}', s'_0 \rangle$ be two lts's. A binary relation $\Re \subseteq \mathcal{S} \times \mathcal{S}'$ is a *bisimulation* if $\langle s_0, s'_0 \rangle \in \Re$ and $\langle s, s' \rangle \in \Re$ implies $\forall \mu \in \mathcal{M}$:

- whenever $s \xrightarrow{\mu} s_1$ then $\exists s'_1$ such that $s' \xrightarrow{\mu} s'_1$ and $\langle s_1, s'_1 \rangle \in \Re$;
- whenever $s' \xrightarrow{\mu} s'_1$ then $\exists s_1$ such that $s \xrightarrow{\mu} s_1$ and $\langle s_1, s'_1 \rangle \in \Re$.

$\mathcal{G}$ and $\mathcal{G}'$ are bisimilar, $\mathcal{G} \sim \mathcal{G}'$, if there exists a bisimulation between them. ■

The notion of bisimulation can be trivially lifted to les's: two les's $\mathcal{E}$ and $\mathcal{F}$ are atomic bisimilar (notation $\mathcal{E} \stackrel{1}{\sim} \mathcal{F}$) if and only if $\mathcal{G}(\mathcal{E}) \sim \mathcal{G}(\mathcal{F})$.

3 Split bisimulation semantics

We start our investigation about non atomic semantics by removing the assumption that events are instantaneous and indivisible in favour of a more realistic one in which they have duration in time. This view is got here by considering events as splitted in two phases. As a consequence, the notion of state of a system (or, better, of configuration) must be generalized to express situations in which only a part of an event has been performed. In this section we will formally define split$_2$ semantics (for a general definition of the entire spectrum of split$_n$ semantics for any natural $n \geq 2$, see [?]).

Let $\mathcal{M}_2$ be the set $\{\mu_i \mid \mu \in \mathcal{M} \wedge 1 \leq i \leq 2\}$. We adopt the convention of distinguishing the various phases of an action by indexing the action itself with the corresponding natural number. Let us define the notions of split$_2$ configuration and split$_2$ transition system associated to a les.

Definition 3.1 A *split$_2$ configuration* of a stable event structure $\mathcal{E}$ is a pair (C, P), where $C \in \mathcal{C}_{\mathcal{E}}$ (the *past*) and $P \subseteq E$ (the *present*) are such that

$$C \cup P \in \mathcal{C}_{\mathcal{E}} \quad \textbf{and} \quad C \cap P = \emptyset \quad \textbf{and} \quad P \subseteq \boldsymbol{Max}(C \cup P)$$

Let $\mathcal{C}^2_{\mathcal{E}}$ be the set of split$_2$ configurations of $\mathcal{E}$. ■

Let us briefly comment on the definition of split$_2$ configuration. The completed events (those in C) alone, and also together with the partially executed events (those in P), constitute a legal configuration of the system. Note that the definition of split$_2$-configuration in [8] is slightly different since, there, P is the set of events which have been completely performed (the *past*), that is $P \subseteq C$. Moreover, a *maximality* condition over the events in P is enforced; this means that any event in P is enabled by a set of completely executed events.

Definition 3.2 Let $\mathcal{G}_2(\mathcal{E}) = \langle \mathcal{C}^2_{\mathcal{E}}, \mathcal{M}_2, \mathcal{T}^2_{\mathcal{E}}, (\emptyset, \emptyset)\rangle$ be the *split$_2$ transition system* related to the les $\mathcal{E}$, where $\mathcal{T}^2_{\mathcal{E}} = \{\xrightarrow{\mu_i} \subseteq (\mathcal{C}^2_{\mathcal{E}} \times \mathcal{C}^2_{\mathcal{E}}) \mid \mu_i \in \mathcal{M}_2\}$ is the set of transition relations defined as follows:
$(C, P) \xrightarrow{\mu_i} (C', P')$ if and only if one of the following conditions holds:

- $i = 1$ **and** $C = C'$ **and** $\exists e. (\ P' = P \cup \{e\}$ **and** $\ell_{\mathcal{E}}(e) = \mu\ )$;
- $i = 2$ **and** $\exists e. (C' = C \cup \{e\}$ **and** $e \in P$ **and** $P' = P \setminus \{e\}$ **and** $\ell_{\mathcal{E}}(e) = \mu\ )$. ■

Definition 3.3 Let $\mathcal{E}, \mathcal{F} \in \mathcal{ES}_{\mathcal{M}}$. $\mathcal{E}$ and $\mathcal{F}$ are *split$_2$ bisimilar* ($\mathcal{E} \overset{2}{\sim} \mathcal{F}$) if and only if $\mathcal{G}_2(\mathcal{E}) \sim \mathcal{G}_2(\mathcal{F})$. ■

4 ST Bisimulation in Implicit Form

In this section we recall (or raphrase) some equivalence notions which give rise to a slight strengthening of split$_2$ bisimulation.

The first one has ben introduced in [8], where it was named ST bisimulation; hereafter we prefer to call it *link preserving* bisimulation.

Definition 4.1 Let $\mathcal{E}$ and $\mathcal{F}$ be two les's. A relation $\Re_{lp} \subseteq \mathcal{C}^2_{\mathcal{E}} \times \mathcal{C}^2_{\mathcal{F}} \times \wp(E_{\mathcal{E}} \times E_{\mathcal{F}})$ such that:

1. $\langle(\emptyset, \emptyset), (\emptyset, \emptyset), \emptyset\rangle \in \Re_{lp}$
2. if $\langle(C, P), (D, Q), f\rangle \in \Re_{lp}$, then for every $\mu_i \in \mathcal{M}_2$:
   - $f : C \cup P \rightarrow D \cup Q$ is a bijection satisfying $\ell_{\mathcal{F}}(f(e)) = \ell_{\mathcal{E}}(e)$ and $f(P) = Q$;

- whenever $(C, P) \xrightarrow{\mu_i} (C', P')$ then $\exists (D', Q'), f'$ such that $(D, Q) \xrightarrow{\mu_i} (D', Q')$, $\langle (C', P'), (D', Q'), f' \rangle \in \Re_{lp}$ and $f' \lceil (C \cup P) = f$;
- whenever $(D, Q) \xrightarrow{\mu_i} (D', Q')$ then $\exists (C', P'), f'$ such that $(C, P) \xrightarrow{\mu_i} (C', P')$, $\langle (C', P'), (D', Q'), f' \rangle \in \Re_{lp}$ and $f' \lceil (C \cup P) = f$.

is called a *link preserving* bisimulation. $\mathcal{E}$ and $\mathcal{F}$ are link equivalent ($\mathcal{E} \overset{lp}{\sim} \mathcal{F}$) if there exists a link preserving bisimulation between them. ■

Link preserving bisimulation is finer than split$_2$ bisimulation because the former is obtained from the latter by adding the further conditions on the bijection f. Function f fixes once and for all the correspondence among the events in the two structures; as a consequence, each transition labelled by an ending phase is unambiguously associated to the transition labelled by the beginning phase generated by the same event. In [11] (see also [1, ?]) the so-called 'owl' example is introduced, which illustrates that link equivalence is strictly finer than split$_2$ equivalence

The second variant of ST equivalence is the one introduced in [1] over split$_2$ transition systems, called there *refine* bisimulation. Here we rephrase it over stable event structures. This definition may be considered a very slight refinement of link preserving bisimulation where only the links associated to the partially executed events are kept.

Definition 4.2 Let $\mathcal{E}$ and $\mathcal{F}$ be two les's. A relation $\Re_\rho \subseteq \mathcal{C}^2_{\mathcal{E}} \times \mathcal{C}^2_{\mathcal{F}} \times \wp(E_{\mathcal{E}} \times E_{\mathcal{F}})$ such that:

1. $\langle (\emptyset, \emptyset), (\emptyset, \emptyset), \emptyset \rangle \in \Re_\rho$
2. if $\langle (C, P), (D, Q), g \rangle \in \Re_\rho$, then for every $\mu_i \in \mathcal{M}_2$:
   - $g : P \rightarrow Q$ is a bijection satisfying $\ell_{\mathcal{F}}(f(e)) = \ell_{\mathcal{E}}(e)$;
   - whenever $(C, P) \xrightarrow{\mu_1} (C, P')$ with $P' = P \cup \{e\}$, then $\exists Q' = Q \cup \{e'\}$ such that $(D, Q) \xrightarrow{\mu_1} (D, Q')$, $\langle (C, P'), (D, Q'), g' \rangle \in \Re_\rho$ and $g' = g \cup \{(e, e')\}$;
   - whenever $(D, Q) \xrightarrow{\mu_1} (D, Q')$ with $Q' = Q \cup \{e'\}$, then $\exists P' = P \cup \{e\}$ such that $(C, P) \xrightarrow{\mu_1} (C, P')$, $\langle (C, P'), (D, Q'), g' \rangle \in \Re_\rho$ and $g' = g \cup \{(e, e')\}$;
   - whenever $(C, P) \xrightarrow{\mu_2} (C', P')$ with $C' = C \cup \{e\}$, $P' = P \setminus \{e\}$, then $(D, Q) \xrightarrow{\mu_2} (D', Q')$, $\langle (C', P'), (D', Q'), g' \rangle \in \Re_\rho$, where $D' = D \cup \{g(e)\}$, $Q' = Q \setminus \{g(e)\}$ and $g' = g \setminus \{(e, g(e))\}$.
   - whenever $(D, Q) \xrightarrow{\mu_2} (D', Q')$ with $D' = D \cup \{e'\}$, $Q' = Q \setminus \{e'\}$, then $(C, P) \xrightarrow{\mu_2} (C', P')$, $\langle (C', P'), (D', Q'), g' \rangle \in \Re_\rho$, where $C' = C \cup \{g^{-1}(e')\}$, $Q' = Q \setminus \{g^{-1}(e')\}$ and $g' = g \setminus \{(g^{-1}(e'), e')\}$.

is called a *refine* bisimulation. $\mathcal{E}$ and $\mathcal{F}$ are refine equivalent ($\mathcal{E} \overset{\rho}{\sim} \mathcal{F}$) if there exists a refine bisimulation between them. ■

Link equivalence imposes a stronger condition than refine equivalence and so we may think that the former is finer than the latter. On the contrary, the associations among completely executed events do not affect the future of the bisimulation definition. This consideration justifies the following theorem.

Theorem 4.3 Given two les's $\mathcal{E}$ and $\mathcal{F}$, ($\mathcal{E} \overset{lp}{\sim} \mathcal{F}$) if and only if ($\mathcal{E} \overset{\rho}{\sim} \mathcal{F}$). ■

A further equivalence, technically defined in a rather different way, is the so-called *maximality preserving* bisimulation, independently proposed by R.Devillers [?] over Petri Nets and by W. Volger [18] over prime event structures. It requires the existence of a bijection h among the events of two related configurations C and D satisfying the strong condition that whenever an event e is maximal in the partial ordering associated to C, the corresponding event $h(e')$ must be maximal in D. Here we propose a slight rearrangement of this notion to the more general setting of stable event structures.

Definition 4.4 Let $\mathcal{E}$ and $\mathcal{F}$ be two les's. A relation $\Re_{max} \subseteq \mathcal{C}_{\mathcal{E}} \times \mathcal{C}_{\mathcal{F}} \times \wp(E_{\mathcal{E}} \times E_{\mathcal{F}})$ such that:

1. $\langle \emptyset, \emptyset, \emptyset \rangle \in \Re_{max}$
2. if $\langle C, D, h \rangle \in \Re_{max}$, then for every $\mu \in \mathcal{M}$:
   - $h : C \to D$ is a bijection which is *label preserving* ($\ell_{\mathcal{F}}(h(e)) = \ell_{\mathcal{E}}(e)$) and *maximality preserving* (e is maximal in $C \Leftrightarrow h(e)$ is maximal in D);
   - whenever $C \xrightarrow{\mu} C'$ then $\exists D', h'$ such that $D \xrightarrow{\mu} D'$, $\langle C', D', h' \rangle \in \Re_{max}$ and $h' \lceil C = h$;
   - whenever $D \xrightarrow{\mu} D'$ then $\exists C', h'$ such that $C \xrightarrow{\mu} C'$, $\langle C', D', h' \rangle \in \Re_{max}$ and $h' \lceil C = h$.

is called a *maximality preserving* bisimulation. $\mathcal{E}$ and $\mathcal{F}$ are maximality preserving equivalent ($\mathcal{E} \overset{max}{\sim} \mathcal{F}$) if there exists a maximality preserving bisimulation between them. ■

Maximality preserving equivalence cannot be considered a *purely observational* equivalence. As a matter of fact, the maximality condition may be tested only by building first the whole partial order associated to the configuration and then by forgetting all the unwanted information. This implicitly requires a very sophisticated observation strategy which is very far from the traditional definition of bisimulation. Moreover, the generalization of this notion to a *weak* (i.e., insensitive to the unobservable action τ) theory is absolutely a nonsense: the execution of an unobservable event labelled by τ may produce the observable effect of changing the maximality property of an already executed event! Nonetheless, maximilaty preserving equivalence is interesting *in se*, because it helps in better understanding what the meaning of ST bisimulation is.

We now prove that link preserving and maximality preserving bisimulations induce the same equivalence (for a different proof, see [18]).

Theorem 4.5 Given two les's $\mathcal{E}$ and $\mathcal{F}$, $\mathcal{E} \overset{lp}{\sim} \mathcal{F}$ if and only if $\mathcal{E} \overset{max}{\sim} \mathcal{F}$.

Proof: $\mathcal{E} \overset{lp}{\sim} \mathcal{F} \Rightarrow \mathcal{E} \overset{max}{\sim} \mathcal{F}$). Let $\Re_{lp}$ be a link preserving bisimulation. Consider the following relation

$$\Re'_{lp} = \{\langle (C, P), (D, Q), f \rangle \in \Re_{lp} \mid e \in C \wedge e \in \boldsymbol{Max}(C \cup P) \ \Rightarrow \ f(e) \in \boldsymbol{Max}(D \cup Q)\}$$

is still a link preserving bisimulation. Indeed, $\langle (\emptyset, \emptyset), (\emptyset, \emptyset), \emptyset \rangle \in \Re'_{lp}$ and, enriching the configurations inductively on the reachability relation, we preserve the maximality property also among completely executed events. ¿From $\Re'_{lp}$, we shall construct a maximality preserving bisimulation $\Re_{max}$ as follows:

$$\Re_{max} = \{(C \cup P, D \cup Q, f) \mid \langle (C, P), (D, Q), f \rangle \in \Re'_{lp}\}$$

Point 1 of Definition 4.4 is trivially satisfied. The first item of point 2 holds as well: f is label preserving by Definition 4.1 and maximality preserving because of the extra condition satisified by relation $\Re'_{lp}$.

Hence assume $(C \cup P, D \cup Q, f) \in \Re_{max}$ and $C \cup P \xrightarrow{\mu} F$. By Definition 2.4, $F = C \cup P \cup \{e\}$ for some $e \in E_{\mathcal{E}}$. Since $\langle (C, P), (D, Q), f \rangle \in \Re'_{lp}$ and $F \in \mathcal{C}_{\mathcal{E}}$, by Definition 3.2 we have that

$$(C, P) \xrightarrow{\mu_1} (C, P \cup \{e\}) \xrightarrow{\mu_2} (C \cup \{e\}, P)$$

According to $\Re'_{lp}$, a similar two-step computation is derivable from $(D,\ Q)$, reaching the split$_2$ configuration $(D \cup \{e'\},\ Q)$. Moreover, f is extended to f' by mapping e to e' and $\langle (C \cup \{e\},\ P),\ (D \cup \{e'\},\ Q),\ f' \rangle \in \Re'_{lp}$. Hence, $(C \cup \{e\} \cup P,\ D \cup \{e'\} \cup Q,\ f') \in \Re_{max}$ because both e and e' are maximal events w.r.t. the final configurations they occur in. (Simmetrically, if we assume that a beginning transition starts from $D \cup Q$.)

$(\mathcal{E} \overset{max}{\sim} \mathcal{F} \Rightarrow \mathcal{E} \overset{lp}{\sim} \mathcal{F})$. Given a maximality preserving bisimulation $\Re_{max}$, we shall construct a link preserving bisimulation $\Re_{lp}$ as follows:

$$\Re_{lp} = \{\langle (C,\ P),\ (D,\ Q),\ f \rangle \mid (C \cup P,\ D \cup Q,\ f) \in \Re_{max}\ \wedge\ e \in P\ \Rightarrow\ e \in Max(C \cup P)\ \wedge\ f(P) = Q\}$$

Point 1 in Definition 4.1 trivially holds. Also the first item of point 2 is satisfied; indeed, f is label preserving by Definition 4.4 and $f(P) = Q$ by construction, as required.

Hence assume $\langle (C,\ P),\ (D,\ Q),\ f \rangle \in \Re_{lp}$ and $(C,\ P) \xrightarrow{\mu_i} (C',\ P')$. There are two subcases according to the value of the phase index i. If $i = 1$, then, by Definition 3.2, $C' = C$ and $P' = P \cup \{e\}$ and, by Definition 2.4, $C \cup P \xrightarrow{\mu} C \cup P'$. By definition of $\Re_{max}$, we have that $D \cup Q \xrightarrow{\mu} D \cup Q'$, with $Q' = Q \cup \{e'\}$, and $(C \cup P',\ D \cup Q',\ f') \in \Re_{max}$, where f is extended to f' by mapping e to e'. Finally, $\langle (C,\ P'),\ (D,\ Q'), f' \rangle \in \Re_{lp}$ because any event in P' is maximal (the new event e is maximal by definition of configuration) and $f'(P') = Q'$. (Simmetrically, if we assume that a beginning transition starts from $(D,\ Q)$.) If $i = 2$, then an event e is moved from P to C. Let us consider the event $e' = f(e)$. According to Definition 3.2, we obtain the transition $(D,\ Q) \xrightarrow{\mu_2} (D',\ Q')$, where $D' = D \cup \{e'\}$ and $P' = P \setminus \{e'\}$. Finally, $\langle (C',\ P'),\ (D',\ Q'), f \rangle \in \Re_{lp}$ because $C \cup P = C' \cup P'$, $D \cup Q = D' \cup Q'$ and, by hypothesis, $\langle (C,\ P),\ (D,\ Q),\ f \rangle \in \Re_{lp}$. (Simmetrically, if we assume that a beginning transition starts from $(D,\ Q)$.) ■

5 An Explicit Formulation of ST Semantics

In this section we provide an explicit formulation of ST bisimulation equivalence and compare it with link preserving bisimulation.

ST bisimulation is, essentially, split$_2$ bisimulation plus a mechanism to recover the individuality of actions. The drawback of split bisimulation, well illustrated in the 'owl' Example [11, 1, ?], is the inability to properly match up endings of actions with their beginnings. For instance, in the split$_2$ trace $\alpha.\gamma_1.\beta.\gamma_1.\gamma_2.\varepsilon$ (unsplitted actions stand for the consecutive execution of their beginnin and ending phases), the action phase γ_2 might be the ending for both the occurrences of the action phase γ_1. This problem is overcome in ST bisimulation by explicitly observing *causal links* connecting each ending phase of an action to its beginning phase. In this view, the split$_2$ trace above become the following ST trace:

$$\alpha.\gamma_1.\beta.\gamma_1.\gamma_2.\varepsilon$$

where a backward pointer stands for causal dependence. We decided to meet this requirement by coding the causal pointer inside the label, an idea which has analogies with [3]. A beginning phase is denoted by the action itself, while an ending phase is superscripted by a natural number k stating that this phase completes the $k-$th started action. As an example, the ST trace above is coded as follows:

$$\alpha.\alpha^1.\gamma.\beta.\beta^3.\gamma.\gamma^2.\varepsilon.\varepsilon^5$$

Formally, the set of *ST labels* is $\mathcal{M}_{ST} = \mathcal{M} \cup \{\mu^k \mid \mu \in \mathcal{M}\ \wedge\ k \in \omega\}$, ranged over by η. Let us now define the notion of ST configuration and of ST transition system.

Definition 5.1 Let $\mathcal{E}$ be a les. An *ST configuration* of $\mathcal{E}$ is a pair (C, P) where $C \in \mathcal{C}_{\mathcal{E}}$ and $P \subseteq E \times \omega^{+}$ such that

1. $(C, \pi_0(P)) \in \mathcal{C}^2_{\mathcal{E}}$
2. $|\pi_1(P)| = |P|$;
3. $\forall \langle e, z\rangle \in P \,.\, |\bigcup_{e' \in R(z)} e'^{*}_{C \cup \pi_0(P)}| \ \leq z \leq \ |C \cup \pi_0(P)|$

where $R(z) = \{e' \mid \langle e', z'\rangle \in P \ \wedge \ z' \leq z\}$. The set of ST configurations of the les $\mathcal{E}$ will be denoted by $\mathcal{C}^{ST}_{\mathcal{E}}$. ■

Let us briefly comment on the definition of ST configuration. Similarly to the case of split$_2$ configuration, C denotes the past and P the presently activated events. Condition 1 makes explicit the analogies with split$_2$ configurations. In particular, every event under execution must be maximal. Condition 2 prevents partially executed events to be labelled by the same natural number; so, all the events are distinguished also via the superscript. Condition 3 states a lower and an upper bound to the number z to be associated to a partially executed event e: z must be greater than (or equal than) the cardinality of the set of all the events causing e and the sets of events causing any other partially executed events e' which started before e; z must be smaller than (or equal to) the number of the involved events.

Proposition 5.2 Let $(C, P) \in \mathcal{C}^{ST}_{\mathcal{E}}$ and $e \notin C \cup \pi_0(P)$. If $C \cup \pi_0(P) \cup \{e\} \in \mathcal{C}_{\mathcal{E}}$ then

$$(C, \, P \cup \{\langle e, \, |C \cup P| + 1\rangle\}) \in \mathcal{C}^{ST}_{\mathcal{E}}$$

Moreover, if $\langle e, k\rangle \in P$ then

$$(C \cup \{e\}, P \setminus \{\langle e, k\rangle\}) \in \mathcal{C}^{ST}_{\mathcal{E}}.$$

Proof: Obvious from Definition 5.1. ■

This proposition ensures that the pairs (C, P) reached by the ST transition relation, defined below, are ST configurations.

Definition 5.3 Let $\mathcal{G}_{ST}(\mathcal{E}) = \langle \mathcal{C}^{ST}_{\mathcal{E}}, \mathcal{M}_{ST}, \mathcal{T}^{ST}_{\mathcal{E}}, (\emptyset, \emptyset)\rangle$ be the labelled ST transition system related to the les $\mathcal{E}$, where $\mathcal{T}^{ST}_{\mathcal{E}} = \{\xrightarrow{\eta} \subseteq (\mathcal{C}^{ST}_{\mathcal{E}} \times \mathcal{C}^{ST}_{\mathcal{E}}) \mid \eta \in \mathcal{M}_{ST}\}$ is the set of transition relations. $(C, P) \xrightarrow{\eta} (C', P')$ if and only if one of the following holds:

- $C' = C$ **and** $\exists \langle e, k\rangle \,.\, [\, P' = P \cup \{\langle e, k\rangle\}$ **and** $k = |C \cup P| + 1$ **and** $\ell_{\mathcal{E}}(e) = \eta \,]$
- $\exists \langle e, k\rangle \in P \,.[\, C' = C \cup \{e\}$ **and** $P' = P \setminus \{\langle e, k\rangle\}$ **and** $\eta = (\ell_{\mathcal{E}}(e))^{k} \,]$. ■

Definition 5.4 Let $\mathcal{E}, \mathcal{F} \in \mathcal{ES}_{\mathcal{M}}$. $\mathcal{E}$ and $\mathcal{F}$ are *ST bisimilar* ($\mathcal{E} \overset{ST}{\sim} \mathcal{F}$) if and only if $\mathcal{G}_{ST}(\mathcal{E}) \sim \mathcal{G}_{ST}(\mathcal{F})$. ■

We want to stress that our formulation of ST equivalence reduces to ordinary bisimulation over a suitably labelled transition system. In the concluding section we will come back to this point.

The definition of ST bisimulation given above differs from link preserving bisimulation. The next theorem states that our definition of ST bisimulation equivalence makes the same identifications of link preserving bisimulation equivalence over $\mathcal{ES}_{\mathcal{M}}$.

Theorem 5.5 Given two les's $\mathcal{E}$ and $\mathcal{F}$, $\mathcal{E} \overset{lp}{\sim} \mathcal{F}$ if and only if $\mathcal{E} \overset{ST}{\sim} \mathcal{F}$.

Proof: ($\mathcal{E} \overset{\star}{\sim} \mathcal{F} \Rightarrow \mathcal{E} \overset{ST}{\sim} \mathcal{F}$). Let $\Re_{lp}$ be a link preserving bisimulation between $\mathcal{E}$ and $\mathcal{F}$. Take the relation $\mathcal{H} \subseteq \mathcal{C}^{ST}_{\mathcal{E}} \times \mathcal{C}^{ST}_{\mathcal{F}}$ defined as follows:

$$((C, P), (D, Q)) \in \mathcal{H} \quad \textit{if} \quad \text{there exists } f \text{ such that}$$

1. $\langle (C, \pi_0(P)), (D, \pi_0(Q)), f\rangle \in \Re_{lp}$
2. $Q = \{\langle f(e), z\rangle \mid \langle e, z\rangle \in P\}$

Let us verify that $\mathcal{H}$ is a bisimulation between $\mathcal{G}_{ST}(\mathcal{E})$ and $\mathcal{G}_{ST}(\mathcal{F})$. The pair $((\emptyset, \emptyset), (\emptyset, \emptyset))$ trivially belongs to $\mathcal{H}$. Let $((C, P), (D, Q)) \in \mathcal{H}$, we have to prove that every move of $\mathcal{G}_{ST}(\mathcal{E})$ in the configuration (C, P) is simulated by $\mathcal{G}_{ST}(\mathcal{F})$ in the configuration (D, Q) and viceversa. Hence assume that $(C, P) \xrightarrow{\mu} (C, P \cup \{\langle e', z'\rangle\})$, where, by Definition 5.3, $z' = |C \cup P| + 1$. By hypothesis there exists a triple $\langle (C, \pi_0(P)), (D, \pi_0(Q)), f\rangle \in \Re_{lp}$ with $\langle e, z\rangle \in P \Leftrightarrow \langle f(e), z\rangle \in Q$. Since $C \cup \pi_0(P) \cup \{e'\} \in \mathcal{C}_{\mathcal{E}}$, according to Definition 3.2, $(C, \pi_0(P)) \xrightarrow{\mu_1} (C, \pi_0(P) \cup \{e'\})$, thus, by item 2.*b* in Definition 4.1, there exists $e'' \in E_{\mathcal{F}}$ such that $(D, \pi_0(Q)) \xrightarrow{\mu_1} (D, \pi_0(Q) \cup \{e''\})$ and $\langle (C, \pi_0(P) \cup \{e'\}), (D, \pi_0(Q) \cup \{e''\}), f \cup \{(e', e'')\}\rangle \in \Re_{lp}$. Notice also that $(D, Q \cup \{\langle e'', z''\rangle\}) \in \mathcal{C}^{ST}_{\mathcal{F}}$ where $z'' = z'$ due to the fact that f is a bijection (i.e., $|C \cup P| = |D \cup Q|$). Therefore, by definition of $\mathcal{H}$, $((C, P \cup \{\langle e', z'\rangle\}), (D, Q \cup \{\langle e'', z''\rangle\})) \in \mathcal{H}$. iWith similar arguments, the case concerning the beginning move from $\mathcal{G}_{ST}(\mathcal{F})$. The simpler case of ending moves is omitted.

($\mathcal{E} \overset{ST}{\sim} \mathcal{F} \Rightarrow \mathcal{E} \overset{lp}{\sim} \mathcal{F}$). Let $\mathcal{H}$ be a bisimulation between $\mathcal{G}_{ST}(\mathcal{E})$ and $\mathcal{G}_{ST}(\mathcal{F})$ and consider the following relation $\Re_{lp} \subseteq \mathcal{C}^2_{\mathcal{E}} \times \mathcal{C}^2_{\mathcal{F}} \times \wp(E_{\mathcal{E}} \times E_{\mathcal{F}})$:

a. $\langle (\emptyset, \emptyset), (\emptyset, \emptyset), \emptyset\rangle \in \Re_{lp}$;

b. $\langle (C, P), (D, Q), f\rangle \in \Re_{lp}$ whenever the following conditions are satisfied:

1. there exists $((C, P^+), (D, Q^+)) \in \mathcal{H}$ such that $\pi_0(P^+) = P$ and $\pi_0(Q^+) = Q$;
2. if $f(e) = e'$ then $\exists z. \langle e, z\rangle \in P^+ \Leftrightarrow \langle e', z\rangle \in Q^+$;
3. for every C', P' with $C' \cup P' = C$ and $(C', P') \in \mathcal{C}^2_{\mathcal{E}}$, $\langle (C', P'), (f(C'), f(Q')), f\lceil C\rangle \in \Re_{lp}$.

We have to prove that $\Re_{lp}$ is a link preserving bisimulation. Item 1 is immediately satisfied. Assume $\langle (C, P), (D, Q), f\rangle \in \Re_{lp}$. Part of item 2.*a* (that concerning the present) in Definition 4.1 is verified by item *b*.2 above. We show by induction over the cardinality of C that f is bijective over the past. The case $C = \emptyset$ is trivial because, by item *b*.2, $D = f(C) = \emptyset$. Assume that f is a bijection when the cardinality of C is n. The case $|C| = n + 1$ can be proved as follows: take $e \in Max(C)$. Then $(C \setminus \{e\}, \{e\}) \in \mathcal{C}^2_{\mathcal{E}}$. Thus, by item *b*.3, $\langle (C \setminus \{e\}, \{e\}), (f(C \setminus \{e\}), \{f(e)\}), f\lceil C\rangle \in \Re_{lp}$. Now, by applying inductive hypothesis, we obtain that f is a bijection between $C \setminus \{e\}$ and $f(C \setminus \{e\})$.

Let us prove that item 2.*b* in Definition 4.1 is satisfied. Let $(C, P) \xrightarrow{\mu_1} (C, P \cup \{e'\})$ and $((C, P^+), (D, Q^+)) \in \mathcal{H}$ satisfying items *b*.1, *b*.2 and *b*.3 above. By Definition 5.3, $(C, P^+) \xrightarrow{\mu} (C, P^+ \cup \{\langle e', z'\rangle\})$, where $z' = |C \cup P^+| + 1$. Thus, since $\mathcal{H}$ is a bisimulation between $\mathcal{G}_{ST}(\mathcal{E})$ and $\mathcal{G}_{ST}(\mathcal{F})$, there exists e'' such that $(D, Q^+) \xrightarrow{\mu} (D, Q^+ \cup \{\langle e'', z'\rangle\})$ and $((C, P^+ \cup \{\langle e', z'\rangle\}), (D, Q^+ \cup \{\langle e'', z'\rangle\})) \in \mathcal{H}$. Let $f' = f \cup \{(e', e'')\}$. It is simple to verify that $\langle (C, P \cup \{e'\}), (D, Q \cup \{e''\}), f'\rangle \in \Re_{lp}$. The other cases are omitted because they may be proved in a similar way. ■

6 Concluding Remarks

The non atomic semantics based on the so-called ST-idea has been investigated in full detail by presenting several alternative definitions which produce the same identifications over stable event structures. All of them, apart from the one we propose in Section 5, make use of external information, such as suitable bijections on the involved events which preserves some interesting properties. Our proposal, instead, is purely extensional in the sense that the observable effect of a transition is completely represented through its label. This improvement might appear rather slight, or even irrelevant. On the contrary, it is fundamental for practical purposes. Indeed, ST bisimulation is reduced to a variant of ordinary bisimulation so that all the mathematical results which comes from its theory can be applied also to this case. An immediate consequence is that it is rather easy to define operational semantics for Process Description Languages in a purely inductive way (following the approach of Plotkin [17]) in such a way that the labels over the transitions directly express the ST information of the causal link. This is what we have done for Milner's CCS in [13] and for Hoare's TCSP in [?]. Furthermore, the fact that ST bisimulation equivalence is a congruence over the language may be easily proved resorting to standard techniques [?], as done in [?]. Finally, also axiomatizations for ST congruence may be given quite easily following a technique developed by Darondeau and Degano [3], even if at the price of introducing an unbounded number of auxiliary operators for the expansion theorem.

References

[1] L. Aceto and M. Hennessy. Adding action refinement to a finite process algebra. In J. Leach Albert, B. Monien, and M. Rodríguez, editors, *Proceedings* 18^{th} *ICALP,* Madrid, volume 510 of *Lecture Notes in Computer Science*, pages 506–519. Springer-Verlag, 1991. A full version of this paper appeared as Report 6/90, Computer Science, University of Sussex, 1990. To appear in *Information and Computation.*

[2] J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors. *REX School/Workshop on Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency,* Noordwijkerhout, volume 354 of *Lecture Notes in Computer Science.* Springer-Verlag, 1989.

[3] P. Darondeau and P. Degano. Causal trees. Publication Interne No. 442, IRISA, Rennes Cedex, France, 1988. An extended abstract appeared in: Proceedings ICALP 89, Stresa (G. Ausiello, M. Dezani-Ciancaglini & S. Ronchi Della Rocca), LNCS 372, Springer-Verlag, pp. 234–248.

[4] P. Darondeau and P. Degano. Causal trees: interleaving + causality. In I. Guessarian, editor, *Semantics of Systems of Concurrent Processes, Proceedings LITP Spring School on Theoretical Computer Science,* La Roche Posay, France, volume 469 of *Lecture Notes in Computer Science*, pages 239–255. Springer-Verlag, 1990.

[5] R. De Nicola. Extensional equivalences for transition systems. *Acta Informatica*, 24:211–237, 1987.

[6] P. Degano, R. De Nicola, and U. Montanari. Partial orderings descriptions and observations of nondeterministic concurrent processes. In Bakker et al. [2], pages 438–466.

[7] R.J. van Glabbeek. The linear time – branching time spectrum. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90,* Amsterdam, volume 458 of *Lecture Notes in Computer Science*, pages 278–297. Springer-Verlag, 1990.

[8] R.J. van Glabbeek. The refinement theorem for ST-bisimulation semantics. In M. Broy and C.B. Jones, editors, *Proceedings IFIP TC2 Working Conference on Programming Concepts and Methods,* Sea of Gallilea, Israel. North-Holland, 1990.

[9] R.J. van Glabbeek and U. Goltz. Equivalence notions for concurrent systems and refinement of actions. Arbeitspapiere der GMD 366, Gesellschaft für Mathematik und Datenverarbeitung, Sankt Augustin, 1989.

An extended abstract appeared in: Proceedings 14^{th} Symposium on Mathematical Foundations of Computer Science, Porąbka-Kozubnik, Poland, August/September 1989 (A. Kreczmar and G. Mirkowska), LNCS 379, Springer-Verlag, pp. 237–248.

[10] R.J. van Glabbeek and F.W. Vaandrager. Petri net models for algebraic theories of concurrency. In J.W. de Bakker, A.J. Nijman, and P.C. Treleaven, editors, *Proceedings PARLE conference,* Eindhoven*, Vol. II (Parallel Languages)*, volume 259 of *Lecture Notes in Computer Science*, pages 224–242. Springer-Verlag, 1987.

[11] R.J. van Glabbeek and F.W. Vaandrager. The difference between splitting in n and $n+1$, 1991. In preparation.

[12] R. Gorrieri. *Refinement, atomicity and transactions for process description languages.* Dissertazione per il titolo di dottore di ricerca in informatica (terzo ciclo), Dipartimento di Informatica, Università di Pisa, 1990.

[13] R. Gorrieri and C. Laneve. The limit of split_n bisimulations for ccs agents. In *Symposium on Mathematical Foundations of Computer Science*, volume 520 of *Lecture Notes in Computer Science*, pages 170 – 180. Springer-Verlag, 1991.

[14] M. Hennessy. Axiomatising finite concurrent processes. *SIAM Journal on Computing*, 17(5):997–1017, 1988.

[15] R. Milner. *Communication and Concurrency.* Prentice-Hall International, Englewood Cliffs, 1989.

[16] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, 5^{th} *GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[17] G.D. Plotkin. A structural approach to operational semantics. Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[18] W. Vogler. Bisimulation and action refinement. SFB-Bericht 342/10/90, TUM, 1990.

[19] G. Winskel. An introduction to event structures. In Bakker et al. [2], pages 364–397.