Joint NCC & IIS Bull., Comp.Science, 5 (1996), 101–114

Equivalence notions for event structures and refinement of actions*

A. Votintseva

We consider different equivalence notions for prime event structures introduced in [6] which explicitly reflect causality, concurrency and conflict relations between occurrences of events in the structures. The intention of the paper is to establish whether or not these equivalences are preserved under refinement of actions. An operator of refinement [2] replaces actions on a given level of abstraction by more complicated processes on a lower level.

1. Introduction

The formalism of event structures has been proposed to describe and study the behaviour of distributed systems. An event structure consists of a set of event occurrences partially ordered by a causality relation. In addition, the structure contains a conflict relation between the events. Two events that are neither causally related nor in conflict are said to be concurrent. Thus the event structure model permits us to explicitly talk about the three basic relations – causality, conflict, and concurrency – between events of distributed systems.

Over the past several years various bisimulation equivalences [2, 5] have been defined on event structures. It is known that variants of forth and back bisimulations [3] capture intuition concerning causality and (implicitly) concurrency but not conflict between event occurrences in the structures. Attempting to get around this lack, we have introduced [6] a number of variants of bisimulations which explicitly reflect all the relations between events. As a particular case, we consider a number of bisimulations which are defined on the domain of local configurations of event structures.

In this paper, we investigate whether or not the introduced bisimulations are preserved under refinement. We give examples showing that variants of interleaving, step and pomset bisimulations are not preserved under refinement. It has turned out that the local bisimulations, being coarser than the corresponding history preserving bisimulations and incomparable with interleaving, step and pomset bisimulations, are also preserved under refinement.

*This work is suppoted in part by the Volkswagen Foundation (grant No I/70 564) and the Russian Foundation of Basic Research (grant No 96-01-01655).

The paper is organized as follows. Section 2 introduces the basic framework, labeled prime event structures, and related notions. Section 3 defines the notion of action refinement. Sections 4, 5 and 6 suggest a number of interleaving, step and pomset (respectively) bisimulation equivalences which reflect not only causality and concurrency but also conflict between event occurrences. We show that all these equivalences are not preserved under action refinement. Sections 7 and 8 consider the variants of stronger equivalences, namely history preserving and back bisimulations which are shown to be invariant under refinement of actions. Section 9 introduces the notions of local bisimulation which are defined on the domain of local configurations of event structures. In the section, we show that all variants of local bisimulations are preserved under the operation of action refinement. Finally, some concluding remarks are made in Section 10.

2. Event structures

In this paper we consider the systems that are capable of performing actions from a given set *Act* of action names. We will use event structures (more precisely, labeled prime event structures [4]) as a fundamental model for computational processes. We will not distinguish external and internal actions here.

Definition 2.1. A (labeled) *event structure* over an alphabet *Act* is a 4-tuple $\mathcal{E} = (E, <, \#, l)$, where

- E is a countable set of events,
- $< \subseteq E \times E$ is an irreflexive partial order (the *causality relation*) satisfying the *principle of finite causes*:
  $$\forall e \in E \bullet \{d \in E \mid d < e\} \text{ is finite},$$
- $\# \subseteq E \times E$ is a symmetric and irreflexive relation (the *conflict relation*) satisfying the *principle of conflict heredity*:
  $$\forall e_1, e_2, e_3 \in E \bullet e_1 < e_2 \;\&\; e_1 \# e_3 \Rightarrow e_2 \# e_3,$$
- $l : E \to Act$ is a labeling function.

Through the paper, we assume *Act* to be a fixed set of action names (labels). The components of an event structure $\mathcal{E}$ are denoted by $E_\mathcal{E}$, $<_\mathcal{E}$, $\#_\mathcal{E}$ and $l_\mathcal{E}$. If it is clear from the context, the index $\mathcal{E}$ is omitted. For an event structure $\mathcal{E}$, we let: $id = \{(e,e) \mid e \in E\}$; $\leq \; = \; < \cup \; id$; $\leq^2 \subseteq \leq$ (transitivity); $\smile \; = (E \times E) \setminus (\leq \cup \leq^{-1} \cup \#)$ (concurrency); $co = \; \smile \cup \; id$.

In graphic representations only immediate conflicts — not the inherited ones — are pictured. The <-relation is represented by arcs omitting those derivable by transitivity. Following these conventions, a trivial example of an event structure is shown in Figure 1, where $E = \{e_1, e_2, e_3, e_4\}$, $< \; = \{(e_1, e_3), (e_1, e_4), (e_2, e_3), (e_2, e_4)\}$, $\# = \{(e_3, e_4), (e_4, e_3)\}$ and $l(e_1) = a$, $l(e_2) = b$, $l(e_3) = a$, $l(e_4) = b$.

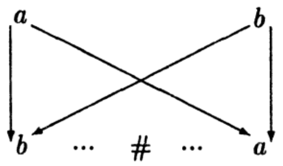


Figure 1

We will frequently give algebraic expressions (see [1]) for our examples, to make them easier to understand. The algebraic syntax includes the primitive constructs: sequential composition (;), parallel composition ($\|$), and sum (+). The operation ; ($\|$, +, respectively) may be easily 'interpreted' by indicating that all events in one component are in the <-relation ($\smile$-relation, #-relation, respectively) with all events in the other.

The event structures $\mathcal{E}$ and $\mathcal{F}$ are *isomorphic* ($\mathcal{E} \cong \mathcal{F}$) iff there exists a bijection between their sets of events preserving <, # and labeling.

The states of an event structure are called configurations. An event can occur in a configuration only if all the events in its past have occurred. Two events that are in conflict can never both occur in the same stretch of behaviour. Before formalizing the notion of a configuration it will be convenient to adopt the following notation. Let $\mathcal{E}$ be an event structure and $C \subseteq E_\mathcal{E}$. Then $\downarrow C = \{e \in E_\mathcal{E} \mid \exists e' \in C \circ e \leq_\mathcal{E} e'\}$. For $e \in E_\mathcal{E}$, we will write $\downarrow e$ instead of $\downarrow \{e\}$. C is said to be a *configuration* of $\mathcal{E}$ iff $C = \downarrow C$ (left-closed) and $\#_\mathcal{E} \cap (C \times C) = \emptyset$ (conflict-free). Let $\mathcal{C}(\mathcal{E})$ denote the set of all configurations of $\mathcal{E}$. It is clear that $\downarrow e$ is a configuration of $\mathcal{E}$ for all $e \in E_\mathcal{E}$. We now define $\mathcal{LC}(\mathcal{E}) = \{\downarrow e \mid e \in E_\mathcal{E}\}$ to be the set of *local configurations* of $\mathcal{E}$. Let $\mathcal{LC}_0(\mathcal{E})$ denote the set $(\mathcal{LC}(\mathcal{E}) \cup \{\emptyset\})$.

For $C' \subseteq C \in \mathcal{C}(\mathcal{E})$ we define the following: C' is called a *step* if $\forall e_1, e_2 \in C' \circ \neg(e_1 <_\mathcal{E} e_2)$; the *restriction* of $\mathcal{E}$ to C' is defined as $\mathcal{E} \lceil C' = (C', <_\mathcal{E} \cap (C' \times C'), \#_\mathcal{E} \cap (C' \times C'), l_\mathcal{E} |_{C'})$. We denote by C' not only the set itself, but also the labeled partial order it induces by restricting $<_\mathcal{E}$ and $l_\mathcal{E}$ to C'. It will (hopefully) be clear from the context what we mean. We use $pom_\mathcal{E}(C) = \{(\mathcal{E} \lceil (C'' \setminus C))/ \cong \mid C \subseteq C'' \in \mathcal{C}(\mathcal{E})\}$ to denote the set of *pomsets* of C.

Definition 2.2. Let $\mathcal{E}$ be an event structure and $C, C' \in \mathcal{C}(\mathcal{E})$. Then

$$C \longrightarrow_\mathcal{E} C' \overset{def}{\Longleftrightarrow} C \subseteq C'.$$

We use $\longmapsto_\mathcal{E}$ to denote $\longrightarrow_\mathcal{E} |_{\mathcal{LC}_0(\mathcal{E})}$.

- $C \xrightarrow{p}_{\mathcal{E}} C' \stackrel{def}{\iff} C \rightarrow_{\mathcal{E}} C'$ and $C' \setminus C = p$, where $p \in pom_{\mathcal{E}}(C)$.
  We use $\stackrel{p}{\longmapsto}_{\mathcal{E}}$ to denote $\xrightarrow{p}_{\mathcal{E}}|_{\mathcal{LC}_0^2(\mathcal{E})}$.
- $C \uparrow_{\mathcal{E}} C' \stackrel{def}{\iff} \exists C'' \in \mathcal{C}(\mathcal{E}) \circ (C \rightarrow_{\mathcal{E}} C''\ \&\ C' \rightarrow_{\mathcal{E}} C'')$.
  We use $\underline{\uparrow}_{\mathcal{E}}$ to denote $\uparrow_{\mathcal{E}}|_{\mathcal{LC}_0^2(\mathcal{E})}$.
- $C \not\uparrow_{\mathcal{E}} C' \stackrel{def}{\iff} \neg(C \uparrow_{\mathcal{E}} C')$.
  We use $\underline{\not\uparrow}_{\mathcal{E}}$ to denote $\not\uparrow_{\mathcal{E}}|_{\mathcal{LC}_0^2(\mathcal{E})}$.
- $C \uparrow'_{\mathcal{E}} C' \stackrel{def}{\iff} \neg(C \not\uparrow_{\mathcal{E}} C' \vee C \rightarrow_{\mathcal{E}} C' \vee C' \rightarrow_{\mathcal{E}} C)$.
  We use $\underline{\uparrow}'_{\mathcal{E}}$ to denote $\uparrow'_{\mathcal{E}}|_{\mathcal{LC}_0^2(\mathcal{E})}$.

Lemma 2.1. [6] *Let $\mathcal{E}$ be an event structure, $C, C' \in \mathcal{C}(\mathcal{E})$ and $\downarrow d, \downarrow d' \in \mathcal{LC}(\mathcal{E})$. Then*

(i) $C \uparrow_{\mathcal{E}} C' \iff C \cup C' \in \mathcal{C}(\mathcal{E})$;

(ii) $C \not\uparrow_{\mathcal{E}} C' \iff \exists e \in C\ \exists e' \in C' \circ e \#_{\mathcal{E}} e'$;

(iii) $C \uparrow'_{\mathcal{E}} C' \iff \forall e \in C \setminus C' \neq \emptyset\ \forall e' \in C' \setminus C \neq \emptyset \circ e \smile_{\mathcal{E}} e'$;

(iv) $\downarrow d \mapsto_{\mathcal{E}} \downarrow d' \iff d \leq_{\mathcal{E}} d'$;

(v) $\downarrow d\ \underline{\not\uparrow}_{\mathcal{E}} \downarrow d' \iff d \#_{\mathcal{E}} d'$;

(vi) $\downarrow d\ \underline{\uparrow}'_{\mathcal{E}} \downarrow d' \iff d \smile_{\mathcal{E}} d'$.

An event structure $\mathcal{E}$ is called an *event structure without autoconcurrency*, if $\forall e, e' \in E_{\mathcal{E}} \circ ((e\ co_{\mathcal{E}}\ e'\ \&\ l_{\mathcal{E}}(e) = l_{\mathcal{E}}(e')) \Rightarrow e = e')$. In the following, we will consider only event structures without autoconcurrency and will simply call them event structures.

3. Refinement of actions

One of the most important features of equivalence notions is its preservation by refinement of actions. We use the definition of refinement from [2]. This operator allows one to design systems in a top-down style, changing the level of abstraction by interpreting actions on a higher level by more complicated processes on a lower level. A refinement function will be a function RF specifying, for each action a, a finite, conflict-free and nonempty event structure $RF(a)$ which is to be substituted for a. Interesting refinements will mostly refine only certain actions, hence replace most actions by themselves. However, for uniformity we consider all actions to be refined. Given an event structure $\mathcal{E}$ and a refinement function RF, we construct the refined event structure $RF(\mathcal{E})$ as follows. Each event e labeled by a is replaced by a disjoint copy, $\mathcal{E}_e$, of $RF(a)$. The causality and conflict structure is inherited from $\mathcal{E}$: every event which was causally before e is causally before all events of $\mathcal{E}_e$, all events which causally followed e will causally follow all the events of $\mathcal{E}_e$, and all events in conflict with e are in conflict with all the events of $\mathcal{E}_e$.

Definition 3.1. Let $\mathcal{E}$ be an event structure and RF be a *refinement function* (for $\mathcal{E}$) which associates a finite, conflict-free, non-empty event structure $RF(a)$ with each action $a \in Act$. Then the *refinement* of $\mathcal{E}$ by RF is the event structure $RF(\mathcal{E}) = (E, <, \#, l)$ defined as follows:

- $E_{RF(\mathcal{E})} = \{(e, e') \mid e \in E_{\mathcal{E}},\ e' \in E_{RF(l_{\mathcal{E}}(e))}\}$,
- $(e, e') <_{RF(\mathcal{E})} (d, d') \iff (e <_{\mathcal{E}} d) \vee (e = d \;\&\; e' <_{RF(l_{\mathcal{E}}(e))} d')$,
- $(e, e') \#_{RF(\mathcal{E})} (d, d') \iff (e \#_{\mathcal{E}} d)$,
- $l_{RF(\mathcal{E})}(e, e') = l_{RF(l_{\mathcal{E}}(e))}(e')$.

Lemma 3.1. [2] *Let $\mathcal{E}$ be an event structure and RF be a refinement function.*

i) *$\widetilde{C} \subseteq E_{RF(\mathcal{E})}$ is a configuration of $RF(\mathcal{E})$ iff*
$\widetilde{C} = \bigcup_{e \in C} \{e\} \times C_e$, where
C is a configuration of $\mathcal{E}$,
C_e is a configuration of $RF(l_{\mathcal{E}}(e))$ for $e \in C$,
$C_e = E_{RF(l(e))}$, if e is not maximal in C with respect to $\leq_{\mathcal{E}}$.
$\widetilde{C}$ is a refinement *of C,*

ii) *If $\widetilde{C} \rightarrow_{RF(\mathcal{E})} \widetilde{C}'$, then $pr_1(\widetilde{C}) \rightarrow_{\mathcal{E}} pr_1(\widetilde{C}')$ (pr_1 denotes a projection to the first component).*

4. Interleaving semantics

In this section, we investigate bisimulation notions in the context of event structures. In order to get equivalences which nicely fit the model under consideration, we introduce some new variants of bisimulations, explicitly expressing all the relations between occurrences of events in the structures. The aim of this section is to consider whether or not these bisimulations are preserved by refinement.

Definition 4.1. Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, $\mathcal{B} \subseteq \mathcal{C}(\mathcal{E}) \times \mathcal{C}(\mathcal{F})$ and $\mathfrak{I} \in \{a, c\}^*$. Then

(i) $\mathcal{B}$ is an *i-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff the following holds:
$(\emptyset, \emptyset) \in \mathcal{B}$,
if $(C, D) \in \mathcal{B}$ and $C \xrightarrow{a}_{\mathcal{E}} C'$,
then there is D' such that $D \xrightarrow{a}_{\mathcal{F}} D'$ and $(C', D') \in \mathcal{B}$.
and vice versa.

(ii) $\mathcal{B}$ is an *ia-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is an i-bisimulation between $\mathcal{E}$ and $\mathcal{F}$ and for all $(C, D) \in \mathcal{B}$ the following holds:

– if $C \not\uparrow_{\mathcal{E}} C'$, then there is D' such that $D \not\uparrow_{\mathcal{F}} D'$ and $(C', D') \in \mathcal{B}$,

– and vice versa.

(iii) $\mathcal{B}$ is an *ic-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is an i-bisimulation between $\mathcal{E}$ and $\mathcal{F}$ and for all $(C, D) \in \mathcal{B}$ the following holds:

– if $C \uparrow'_{\mathcal{E}} C'$, then there is D' such that $D \uparrow'_{\mathcal{F}} D'$ and $(C', D') \in \mathcal{B}$,

– and vice versa.

$\mathcal{E}$ and $\mathcal{F}$ are *iβ-bisimilar*, denoted $\mathcal{E} \approx_{i\beta} \mathcal{F}$, if there exists an $i\beta$-bisimulation, i.e. a relation $\mathcal{B}$ which is an $i\gamma$-bisimulation for all $\gamma \in \beta$.

Proposition 4.1. *Let $\beta \in \{a, c\}^*$. Then $\approx_{i\beta}$ is not preserved under refinement.*

Proof. Let us first consider the event structures $\mathcal{E}$ and $\mathcal{F}$:

The composed event structures $\mathcal{E}_1 = \mathcal{E} + \mathcal{F}$ and $\mathcal{F}_1 = \mathcal{F} + \mathcal{F}$ are $i\beta$-bisimilar. However, when refining the action b by the sequence $b_1 \rightarrow b_2$, we get $\mathcal{E}_1 \not\approx_{i\beta} \mathcal{F}_1$.

5. Step semantics

A more discriminating view of concurrent systems than that offered by interleaving semantics is obtained by modelling concurrency as either arbitrary interleaving or simultaneous execution. The word *step* originates from Petri net theory where it denotes a set (or multiset) of concurrently executable transitions. Step semantics give a more precise account of concurrency than interleaving semantics, e.g., the systems $a||b$ and $a; b + b; a$ are distinguished. We will formalize some step equivalence notions and then discuss an example which shows that these equivalences are not also preserved by refinement.

Step semantics are defined by generalizing the single action transitions $C \xrightarrow{a} C'$ to transitions of the form $C \xrightarrow{A} C'$, where A is a multiset over

Act, representing actions which occur concurrently. Using this kind of transitions, we get different variants of ***step bisimulation*** being straightforward generalizations of the corresponding interleaving equivalences.

Definition 5.1. Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, $\mathcal{B} \subseteq \mathcal{C}(\mathcal{E}) \times \mathcal{C}(\mathcal{F})$ and $\beta \in \{a,c\}^*$. Then

(i) $\mathcal{B}$ is an *s-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff the following holds:
- $(\emptyset,\emptyset) \in \mathcal{B}$,
- if $(C,D) \in \mathcal{B}$ and $C \xrightarrow{p}_{\mathcal{E}} C'$ such that p is a step, then there are D' and q such that $D \xrightarrow{q}_{\mathcal{F}} D'$, $(C',D') \in \mathcal{B}$ and $p \cong q$,
- and vice versa.

(ii) $\mathcal{B}$ is an *sa-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is an s-bisimulation between $\mathcal{E}$ and $\mathcal{F}$ and for all $(C,D) \in \mathcal{B}$ the following holds:
- if $C \not\uparrow_{\mathcal{E}} C'$, then there is D' such that $D \not\uparrow_{\mathcal{F}} D'$ and $(C',D') \in \mathcal{B}$,
- and vice versa.

(iii) $\mathcal{B}$ is an *sc-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is an s-bisimulation between $\mathcal{E}$ and $\mathcal{F}$, and for all $(C,D) \in \mathcal{B}$ the following holds:
- if $C \uparrow'_{\mathcal{E}} C'$, then there is D' such that $D \uparrow'_{\mathcal{F}} D'$ and $(C',D') \in \mathcal{B}$,
- and vice versa.

$\mathcal{E}$ and $\mathcal{F}$ are *$s\beta$-bisimilar*, denoted $\mathcal{E} \approx_{s\beta} \mathcal{F}$, if there exists an $s\beta$-bisimulation, i.e. a relation $\mathcal{B}$ which is an $s\gamma$-bisimulation for all $\gamma \in \beta$.

Proposition 5.1. *Let $\beta \in \{a,c\}^*$. Then $\approx_{s\beta}$ is not preserved under refinement.*

Proof. Let us consider the event structures $\mathcal{E}_2 = \mathcal{E} + \mathcal{F} + (a||b)$ and $\mathcal{F}_2 = \mathcal{F} + \mathcal{F} + (a||b)$ (where $\mathcal{E}$ and $\mathcal{F}$ are from Proposition 4.1) which are $s\beta$-bisimilar. However, when refining the action b by the sequence $b_1 \rightarrow b_2$, we get $\mathcal{E}_1 \not\approx_{s\beta} \mathcal{F}_1$.

6. Partial order semantics

In [7] it was suggested to generalize the idea of bisimulation by considering transitions labeled by pomsets. So we consider now transitions $C \xrightarrow{u} C'$, where u is a pomset over *Act*.

Definition 6.1. Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, $\mathcal{B} \subseteq \mathcal{C}(\mathcal{E}) \times \mathcal{C}(\mathcal{F})$ and $\beta \in \{a,c\}^*$. Then

(i) $\mathcal{B}$ is a *p-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff the following holds:
   - $(\emptyset,\emptyset) \in \mathcal{B}$,
   - if $(C,D) \in \mathcal{B}$ and $C \xrightarrow{p}_{\mathcal{E}} C'$,
     then there are D' and q such that
     $D \xrightarrow{q}_{\mathcal{F}} D'$, $(C',D') \in \mathcal{B}$ and $p \cong q$,
   - and vice versa.

(ii) $\mathcal{B}$ is a *pa-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is a p-bisimulation between $\mathcal{E}$ and $\mathcal{F}$ and for all $(C,D) \in \mathcal{B}$ the following holds:
   - if $C \not\uparrow_{\mathcal{E}} C'$, then there is D' such that $D \not\uparrow_{\mathcal{F}} D'$ and $(C',D') \in \mathcal{B}$,
   - and vice versa.

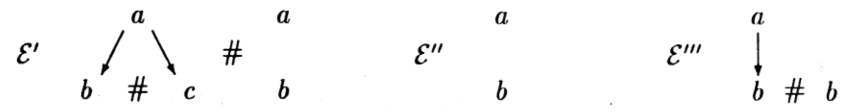
(iii) $\mathcal{B}$ is a *pc-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is a p-bisimulation between $\mathcal{E}$ and $\mathcal{F}$, and for all $(C,D) \in \mathcal{B}$ the following holds:
   - if $C \uparrow'_{\mathcal{E}} C'$, then there is D' such that $D \uparrow'_{\mathcal{F}} D'$ and $(C',D') \in \mathcal{B}$,
   - and vice versa.

$\mathcal{E}$ and $\mathcal{F}$ are *$p\beta$-bisimilar*, denoted $\mathcal{E} \approx_{p\beta} \mathcal{F}$, if there exists a $p\beta$-bisimulation, i.e. a relation $\mathcal{B}$ which is a $p\gamma$-bisimulation for all $\gamma \in \beta$.

These equivalences are evidently stronger than the corresponding step bisimulations: $\mathcal{E} \approx_{p\beta} \mathcal{F}$ implies $\mathcal{E} \approx_{s\beta} \mathcal{F}$; moreover, $\mathcal{E}_1 \approx_{sac} \mathcal{F}_1$ and $\mathcal{E}_1 \not\approx_{pac} \mathcal{F}_1$.

Proposition 6.1. *Let $\beta \in \{a,c\}^*$. Then $\approx_{p\beta}$ is not preserved under refinement.*

Proof. Let us first consider the event strucrures $\mathcal{E}'$, $\mathcal{E}''$ and $\mathcal{E}'''$:



The composed structures $\mathcal{E}_2 = \mathcal{E}' + \mathcal{E}''$ and $\mathcal{F}_2 = \mathcal{E}' + \mathcal{E}'''$ are $p\beta$-bisimular, but when refining the action a by the sequence $a_1 \to a_2$ we get $\mathcal{E}_2 \not\approx_{p\beta} \mathcal{F}_2$.

7. History preserving bisimulations

Another equivalence notion based on the idea of bisimulation with partial orders that might be preserved by refinement has been suggested in [8]. It has turned out that this notion coincides with the NMS partial ordering equivalence suggested earlier in [9]. We give here the definitions of *history* *preserving equivalences* in terms of event structures. We will show that these equivalences are preserved by refinement.

Definition 7.1. Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, $\mathcal{B} \subseteq \mathcal{C}(\mathcal{E}) \times \mathcal{C}(\mathcal{F})$, and $\beta \in \{a, c\}^*$. Then

(i) $\mathcal{B}$ is an *h-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff the following holds:
- $(\emptyset, \emptyset) \in \mathcal{B}$,
- if $(C, D) \in \mathcal{B}$, then
  - $\mathcal{E} \lceil C \cong \mathcal{F} \lceil D$,
  - if $C \xrightarrow{p}_{\mathcal{E}} C'$, then there are D' and q such that $D \xrightarrow{q}_{\mathcal{F}} D'$, $(C', D') \in \mathcal{B}$ and $p \cong q$,
- and vice versa.

(ii) $\mathcal{B}$ is an *ha-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is an h-bisimulation between $\mathcal{E}$ and $\mathcal{F}$ and for all $(C, D) \in \mathcal{B}$ the following holds:
- if $C \not\gamma_{\mathcal{E}} C'$, then there is D' such that $D \not\gamma_{\mathcal{F}} D'$ and $(C', D') \in \mathcal{B}$,
- and vice versa.

(iii) $\mathcal{B}$ is an *hc-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is an h-bisimulation between $\mathcal{E}$ and $\mathcal{F}$, and for all $(C, D) \in \mathcal{B}$ the following holds:
- if $C \upharpoonleft'_{\mathcal{E}} C'$, then there is D' such that $D \upharpoonleft'_{\mathcal{F}} D'$ and $(C', D') \in \mathcal{B}$,
- and vice versa.

$\mathcal{E}$ and $\mathcal{F}$ are *$h\beta$-bisimilar*, denoted $\mathcal{E} \approx_{h\beta} \mathcal{F}$, if there exists an $h\beta$-bisimulation, i.e. a relation $\mathcal{B}$ which is an $h\gamma$-bisimulation for all $\gamma \in \beta$.

Note that the isomorphism requirement guarantees that the labels of the events in $C' \setminus C$ correspond to those in $D' \setminus D$. In [6] it has been shown that $\mathcal{E} \approx_{h\beta} \mathcal{F}$ implies $\mathcal{E} \approx_{p\beta} \mathcal{F}$.

Proposition 7.1. *Let $\beta \in \{a, c\}^*$. Then $\approx_{h\beta}$ is preserved under refinement.*

Proof. Assume $\mathcal{E}$ and $\mathcal{F}$ to be event structures, RF be a refinement function and $\mathcal{B}$ be an $h\beta$-bisimulation between $\mathcal{E}$ and $\mathcal{F}$. We make some preparations. Define $\tilde{\mathcal{B}} = \{(\tilde{C}, \tilde{D}) \in \mathcal{C}(RF(\mathcal{E})) \times \mathcal{C}(RF(\mathcal{F}))|\ \exists (C, D) \in \mathcal{B}\ \exists f : C \to D \bullet pr_1(\tilde{C}) = C,\ pr_1(\tilde{D}) = D$, and f is an isomorphism satisfying $\forall e \in C \bullet C_e = D_{f(e)}\}$. It has been shown in [2] that $\tilde{\mathcal{B}}$ is an h-bisimulation between $RF(\mathcal{E})$ and $RF(\mathcal{F})$. For $D \in \mathcal{C}(\mathcal{F})$ such that $(C, D) \in \mathcal{B}$, we define $ref_{\tilde{C}}(D) = \bigcup_{d \in D} \{d\} \times D_d$ with $D_d = C_{f^{-1}(d)}$ for all $d \in D$ and some isomorphism f from C onto D. Due to Definition 3.1 and Lemma 3.1(i), it is easy to see that $ref_{\widetilde{C}}(D) \in \mathcal{C}(RF(\mathcal{F}))$. By construction of $\widetilde{\mathcal{B}}$, we get $(\widetilde{C}, ref_{\widetilde{C}}(D)) \in \widetilde{\mathcal{B}}$.

We now show that $\widetilde{\mathcal{B}}$ is an $h\beta$-bisimulation between $\mathcal{E}$ and $\mathcal{F}$. Assume $(\widetilde{C}, \widetilde{D}) \in \widetilde{\mathcal{B}}$. Then $(C, D) \in \mathcal{B}$, by construction of $\widetilde{\mathcal{B}}$. We consider two cases.

$\beta = a$. Assume $\widetilde{C} \not\uparrow_{RF(\mathcal{E})} \widetilde{C}'$. Due to Lemma 2.1(ii), there are $(e,g) \in \widetilde{C}$ and $(e',g') \in \widetilde{C}'$ such that $(e,g) \#_{RF(\mathcal{E})} (e',g')$. This implies $e \in C$, $e' \in C'$ and $e \#_{\mathcal{E}} e'$ by Definition 3.1 and Lemma 3.1(i). Hence $C \not\uparrow_{\mathcal{E}} C'$ once again by Lemma 2.1(ii). Since $(C,D) \in \mathcal{B}$, there exists D' such that $D \not\uparrow_{\mathcal{F}} D'$ and $(C',D') \in \mathcal{B}$ by Definition 7.1(ii). Take $\widetilde{D}' = ref_{\widetilde{C}'}(D')$. Then $(\widetilde{C}', \widetilde{D}') \in \widetilde{\mathcal{B}}$. Due to Lemma 2.1(ii), there are $d \in D$ and $d' \in D'$ such that $d \#_{\mathcal{F}} d'$. By Definition 3.1, $(d,g) \#_{RF(\mathcal{F})} (d',g')$ for all $g \in E_{RF(l_{\mathcal{F}}(d))}$ and $g' \in E_{RF(l_{\mathcal{F}}(d'))}$. Due to Lemma 3.1(i), we have $(d,g) \in \widetilde{D}$ and $(d',g') \in \widetilde{D}'$ for some $g \in E_{RF(l_{\mathcal{F}}(d))}$ and $g' \in E_{RF(l_{\mathcal{F}}(d'))}$. Once again from Lemma 2.1(ii), it follows that $\widetilde{D} \not\uparrow_{RF(\mathcal{F})} \widetilde{D}'$.

$\beta = c$. Assume $\widetilde{C} \uparrow'_{RF(\mathcal{E})} \widetilde{C}'$. Due to Lemma 3.1(i) and Definition 2.2, the three cases are admissible:

- $C \xrightarrow{p}_{\mathcal{E}} C'$. By Definition 7.1(i), there are D' and q such that $D \xrightarrow{q}_{\mathcal{F}} D'$, $p \cong q$ and $(C',D') \in \mathcal{B}$.
- $C' \xrightarrow{p}_{\mathcal{E}} C$. Due to Definition 7.1(i), there are D' and q such that $D' \xrightarrow{q}_{\mathcal{F}} D$, $p \cong q$ and $(C',D') \in \mathcal{B}$.
- $C \uparrow'_{\mathcal{E}} C'$. By Definition 7.1(iii), there exists D' such that $D \uparrow'_{\mathcal{F}} D'$ and $(C',D') \in \mathcal{B}$.

Let $\widetilde{D}' = ref_{\widetilde{C}'}(D')$. Then $(\widetilde{C}', \widetilde{D}') \in \widetilde{\mathcal{B}}$. By Lemma 2.1(iii), we have $(e,g) \smile_{RF(\mathcal{E})} (e',g')$ for all $(e,g) \in (\widetilde{C} \setminus \widetilde{C}')$ and $(e',g') \in (\widetilde{C}' \setminus \widetilde{C})$. Take arbitrary $(e,g) \in (\widetilde{C} \setminus \widetilde{C}')$ and $(e',g') \in (\widetilde{C}' \setminus \widetilde{C})$. Let $f : C \to D$ and $f' : C' \to D'$ be isomorphisms. By Lemma 3.1(i) and construction of $\widetilde{\mathcal{B}}$, we have $(f(e),g) \in (\widetilde{D} \setminus \widetilde{D}')$ and $(f(e'),g') \in (\widetilde{D}' \setminus \widetilde{D})$. It is sufficient to show that $(f(e),g) \smile_{RF(\mathcal{F})} (f(e'),g')$. Suppose a contrary, i.e. $\neg((f(e),g) \smile_{RF(\mathcal{F})} (f(e'),g'))$. Clearly, $(f(e),g) \neq_{RF(\mathcal{F})} (f(e'),g')$. Then three cases remain to be considered.

- $(f(e),g) \#_{RF(\mathcal{F})} (f(e'),g')$. This implies $f(e) \in D$, $f(e') \in D'$ and $f(e) \#_{\mathcal{F}} f(e')$ due to Definition 3.1 and Lemma 3.1(i). By Lemma 2.1(ii), we have $D \not\uparrow_{\mathcal{F}} D'$ contradicting Definition 2.2.
- $(f(e),g) <_{RF(\mathcal{F})} (f(e'),g')$. This contradicts $\widetilde{D}' \in \mathcal{C}(RF(\mathcal{F})$.
- $(f(e'),g') <_{RF(\mathcal{F})} (f(e),g)$. This contradicts $\widetilde{D} \in \mathcal{C}(RF(\mathcal{F})$.

Therefore $\widetilde{D} \uparrow'_{RF(\mathcal{F})} \widetilde{D}'$ due to Lemma 2.1(iii).

Thus $\approx_{h\beta}$ is preserved under refinement, due to Definition 7.1.

8. Back bisimulations

In this section we introduce back variants of the bisimulations defined above. It was shown in [6] that interleaving, step, pomset and history preserving back bisimulations coincide. So $\approx_{habc}$ is the strongest equivalence considered so far (except for event structure isomorphism, of course).

Definition 8.1. Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, $\mathcal{B} \subseteq \mathcal{C}(\mathcal{E}) \times \mathcal{C}(\mathcal{F})$, $\alpha \in \{i,s,p,h\}$ and $\beta \in \{a,b,c\}^*$. Then $\mathcal{B}$ is an *αb-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is an α-bisimulation between $\mathcal{E}$ and $\mathcal{F}$ and for all $(C,D) \in \mathcal{B}$ the following holds:

- If $C' \xrightarrow{p}_{\mathcal{E}} C$ such that
  - p has at most one element, if $\alpha = i$,
  - p is a step, if $\alpha = s$,

  then there are D' and q such that

  $D' \xrightarrow{q}_{\mathcal{F}} D$, $(C',D') \in \mathcal{B}$ and $p \cong q$,
- and vice versa.

$\mathcal{E}$ and $\mathcal{F}$ are *αβ-bisimilar*, denoted $\mathcal{E} \approx_{\alpha\beta} \mathcal{F}$, if there exists an $\alpha\beta$-bisimulation, i.e. a relation $\mathcal{B}$ which is an $\alpha\gamma$-bisimulation for all $\gamma \in \beta$. □

Proposition 8.1. [6] Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, $\alpha, \alpha' \in \{i,s,p,h\}$ and $\beta \in \{a,c\}^*$. Then

$$\mathcal{E} \approx_{\alpha\beta b} \mathcal{F} \Longleftrightarrow \mathcal{E} \approx_{\alpha'\beta b} \mathcal{F}.$$

Proposition 8.2. *Let $\alpha \in \{i,s,p,h\}$ and $\beta \in \{a,c\}^*$. Then $\approx_{\alpha b\beta}$ is preserved under refinement.*

Proof. Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, RF be a refinement function and $\mathcal{B}$ be an $hb\beta$-bisimulation between $\mathcal{E}$ and $\mathcal{F}$. Define $\widetilde{\mathcal{B}} = \{(\widetilde{C},\widetilde{D}) \in \mathcal{C}(RF(\mathcal{E}))\times\mathcal{C}(RF(\mathcal{F}))|\, \exists (C,D) \in \mathcal{B}\, \exists f : C \to D \bullet pr_1(\widetilde{C}) = C, pr_1(\widetilde{D}) = D$, and f is an isomorphism satisfying $\forall e \in C \bullet C_e = C_{f(e)}\}$. By Proposition 7.1, $\widetilde{\mathcal{B}}$ is an $h\beta$-bisimulation between $RF(\mathcal{E})$ and $RF(\mathcal{F})$.

We have to show that $\widetilde{\mathcal{B}}$ is an hb-bisimulation between $\mathcal{E}$ and $\mathcal{F}$. Let $(\widetilde{C}.\widetilde{D}) \in \widetilde{\mathcal{B}}$.

Assume $\widetilde{C}' \xrightarrow{p}_{RF(\mathcal{E})} \widetilde{C}$. It is necessary to show that there exist $\widetilde{D}'$ and q such that $\widetilde{D}' \xrightarrow{q}_{RF(\mathcal{F})} \widetilde{D}$ and $(\widetilde{C}',\widetilde{D}') \in \widetilde{\mathcal{B}}$. Since $C = pr_1(\widetilde{C})$ and $C' = pr_1(\widetilde{C}')$, we have $C' \to_{\mathcal{E}} C$, by Lemma 3.1(ii). Let $C \backslash C' = p'$. Then $C' \xrightarrow{p'}_{\mathcal{E}} C$ according to Definition 2.2. Since $(C,D) \in \mathcal{B}$, there exist D' and q' such that $D' \xrightarrow{q'}_{\mathcal{F}} D$ and $(C'.D') \in \mathcal{B}$, due to Definition 8.1. Take $\widetilde{D}' = \bigcup_{d\in D'}\{d\} \times D'_d$, where $D'_d = C'_{f'^{-1}(d)}$ for all $d \in D'$ and some isomorphism $f' : C' \to D'$.

Then $\widetilde{D}' \in \mathcal{C}(RF(\mathcal{F}))$ and $(\widetilde{C}', \widetilde{D}') \in \widetilde{\mathcal{B}}$. Since $\widetilde{C}' \xrightarrow{p}_{RF(\mathcal{E})} \widetilde{C}$, we have $C'_e \subseteq C_e$ for all $e \in C'$ due to Definition 2.2 and Lemma 3.1(i). Hence $D'_d \subseteq D_d$ for all $d \in D'$, from construction of $\widetilde{\mathcal{B}}$. Moreover, since $D' \xrightarrow{q'}_{\mathcal{F}} D$, we have $D' \subseteq D$ by Definition 2.2. Thus $\widetilde{D}' \subseteq \widetilde{D}$ with $\widetilde{D} \setminus \widetilde{D}' = q$. This implies $\widetilde{D}' \xrightarrow{q}_{RF(\mathcal{F})} \widetilde{D}$, once again by Definition 2.2.

Thus $\approx_{hb\beta}$ is preserved under refinement by Definition 7.1. Due to Proposition 8.1, it follows that $\approx_{\alpha b\beta}$ is also preserved under refinement.

9. Local bisimulations

We now introduce a number of bisimulations which are directly defined on the domain of local configurations of the event structures. As it has been shown in [6], these notions are useful for discovering a match for the equivalence induced by the logic L_1 in [10].

Definition 9.1. Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, $\mathcal{B} \subseteq \mathcal{LC}_0(\mathcal{E}) \times \mathcal{LC}_0(\mathcal{F})$ and $\beta \in \{a, b, c\}^*$. Then

(i) $\mathcal{B}$ is a *local bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff the following holds:

- $(\emptyset, \emptyset) \in \mathcal{B}$,
- if $(C, D) \in \mathcal{B}$ and $C \overset{p}{\mapsto}_{\mathcal{E}} C'$,

  then there are D' and q such that

  $D \overset{q}{\mapsto}_{\mathcal{F}} D'$, $p \cong q$ and $(C', D') \in \mathcal{B}$,
- and vice versa.

(ii) $\mathcal{B}$ is a *local b-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is a local bisimulation between $\mathcal{E}$ and $\mathcal{F}$, and for all $(C, D) \in \mathcal{B}$ the following holds:

- if $C' \overset{p}{\mapsto}_{\mathcal{E}} C$, then there are D' and q such that

  $D' \overset{q}{\mapsto}_{\mathcal{F}} D$, $p \cong q$ and $(C', D') \in \mathcal{B}$,
- and vice versa.

(iii) $\mathcal{B}$ is a *local a-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is a local bisimulation between $\mathcal{E}$ and $\mathcal{F}$, and for all $(C, D) \in \mathcal{B}$ the following holds:

- if $C \not\uparrow_{\mathcal{E}} C'$, then there is D' such that $D \not\uparrow_{\mathcal{F}} D'$ and $(C', D') \in \mathcal{B}$,
- and vice versa.

(iv) $\mathcal{B}$ is a *local c-bisimulation* between $\mathcal{E}$ and $\mathcal{F}$ iff $\mathcal{B}$ is a local bisimulation between $\mathcal{E}$ and $\mathcal{F}$, and for all $(C, D) \in \mathcal{B}$ the following holds:

- if $C \uparrow'_{\mathcal{E}} C'$, then there is D' such that $D \uparrow'_{\mathcal{F}} D'$ and $(C', D') \in \mathcal{B}$.
- and vice versa.

$\mathcal{E}$ and $\mathcal{F}$ are *locally β-bisimilar*, denoted $\mathcal{E} \approx_{l\beta} \mathcal{F}$, if there exists a local β-bisimulation, i.e. a relation $\mathcal{B}$ which is a local γ-bisimulation for all $\gamma \in \beta$.

Before formulating the main result of the section, we point out the properties of local bisimulations.

Lemma 9.1. [6] *Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, $\mathcal{B}$ be the minimal l-bisimulation between $\mathcal{E}$ and $\mathcal{F}$, and $(C, D) \in \mathcal{B}$. Then $\mathcal{E}\lceil C \cong \mathcal{F}\lceil D$.*

Lemma 9.2. [6] *Let $\mathcal{E}$ and $\mathcal{F}$ be event structures and $\beta \in \{a, c\}^*$. Then*

$$\mathcal{E} \approx_{l\beta} \mathcal{F} \iff \mathcal{E} \approx_{lb\beta} \mathcal{F}.$$

Proposition 9.1. *Let $\beta \in \{a, b, c\}^*$. Then $\approx_{l\beta}$ is preserved under refinement.*

Proof. Let $\mathcal{E}$ and $\mathcal{F}$ be event structures, RF be a refinement function and $\mathcal{B}$ be the minimal $labc$-bisimulation between $\mathcal{E}$ and $\mathcal{F}$. Let $\tilde{\mathcal{B}} = \{(\tilde{C}, \tilde{D}) \in \mathcal{LC}_0(RF(\mathcal{E})) \times \mathcal{LC}_0(RF(\mathcal{F})) | \; \exists (C, D) \in \mathcal{B} \; \exists f : C \to D \circ pr_1(\tilde{C}) = C$, $pr_1(\tilde{D}) = D$ and f is an isomorphism satisfying $\forall e \in C \circ C_e = C_{f(e)}\}$.

We first show that $\tilde{\mathcal{B}}$ is an l-bisimulation. Obviously, $(\emptyset, \emptyset) \in \tilde{\mathcal{B}}$. Let $(\tilde{C}, \tilde{D}) \in \tilde{\mathcal{B}}$ and $\tilde{C} \xrightarrow{p}_{RF(\mathcal{E})} \tilde{C}'$. By construction of $\tilde{\mathcal{B}}$, we have $C = pr_1(\tilde{C}) \in \mathcal{LC}(\mathcal{E})$ and $D = pr_1(\tilde{D}) \in \mathcal{LC}(\mathcal{F})$. Then $C' = pr_1(\tilde{C}') \in \mathcal{LC}(\mathcal{E})$ due to Lemma 3.1 and Definition 3.1, and $C \overset{p'}{\mapsto}_{\mathcal{E}} C'$ due to Definition 2.2. Since $(C, D) \in \mathcal{B}$, there exists $D' \in \mathcal{LC}(\mathcal{F})$ such that $D \overset{q'}{\mapsto}_{\mathcal{F}} D'$ and $(C', D') \in \mathcal{B}$ by Definition 9.1(i). Let $\tilde{D}' = \bigcup_{d \in D'} \{d\} \times D'_d$, where $D'_d = C'_{f'^{-1}(d)}$ for all $d \in D'$ and some isomorphism $f' : C' \to D'$. From Definition 3.1 and Lemma 3.1, it is easy to see that $\tilde{D}' \in \mathcal{LC}(RF(\mathcal{F}))$. By construction of $\tilde{\mathcal{B}}$, we get $(\tilde{C}', \tilde{D}') \in \tilde{\mathcal{B}}$. It remains to show that $\tilde{D} \overset{q}{\mapsto}_{RF(\mathcal{F})} \tilde{D}'$ and $p \cong q$. Due to Definition 2.2, we have $D \subseteq D'$. From Lemma 3.1, it follows that $C_d \subseteq C'_d$ for all $d \in D$. This means that $\tilde{D} \subseteq \tilde{D}'$ with $\tilde{D}' \setminus \tilde{D} = q$. Hence $\tilde{D} \overset{q}{\mapsto}_{RF(\mathcal{F})} \tilde{D}'$, once again by Definition 2.2. Since $(C, D), (C', D') \in \mathcal{B}$ and $\mathcal{B}$ is the minimal $labc$-bisimulation, $\mathcal{E}\lceil C \cong \mathcal{F}\lceil D$ and $\mathcal{E}\lceil C' \cong \mathcal{F}\lceil D'$ by Lemma 9.1. Thus $RF(\mathcal{E})\lceil \tilde{C} \cong RF(\mathcal{F})\lceil \tilde{D}$ and $RF(\mathcal{E})\lceil \tilde{C}' \cong RF(\mathcal{F})\lceil \tilde{D}'$. Hence $p \cong q$.

Thus $\tilde{\mathcal{B}}$ is an l-bisimulation between $RF(\mathcal{E})$ and $RF(\mathcal{F})$. Then $\tilde{\mathcal{B}}$ is also an lb-bisimulation due to Lemma 9.2. Reasoning analogously to the proof of Proposition 7.1, it can be shown that $\tilde{\mathcal{B}}$ is an la- and lc-bisimulation between $RF(\mathcal{E})$ and $RF(\mathcal{F})$. This means that $\approx_{l\beta}$ is preserved under refinement.

10. Concluding remarks

In this paper we have established what kinds of bisimulations, introduced in [6], are preserved under refinement. In order to get equivalence notions nicely adapted to peculiarities of event structures, we have introduced a number of variants of bisimulation notions which reflect all the relations between event occurrences in the structures. We have shown that "forth" bisimulations based on interleaving of atomic actions, steps or pomsets are not preserved when changing the level of atomicity of actions. However, we have managed to show that bisimulations, keeping causal structures of runs (perhaps implicitly) under consideration, are indeed preserved by refinement of actions.

It is clear that the main difference between the logical equivalence and any behavioural one known from the literature is that the former uses only local configurations, whereas the latter as a rule considers all configurations of event structures. Therefore a number of variants of bisimulations which are defined on the domain of local configurations of event structures were put forward. In this paper we have shown that all variants of local bisimulations are preserved under refinement of action.

References

[1] G. Boudol, I. Castellani, *Concurrency and atomicity*, Theoretical Computer Science, **59**, 1988, 25–84.

[2] R. Glabbeek, U. Goltz, *Equivalence notions for concurrent systems and refinement of actions*, Lectures Notes in Computer Science, **379**, 1989, 237–248.

[3] R. deNicola, U. Montanari, F. Vaandrager, *Back and forth bisimulations*, Lectures Notes in Computer Science, **458**, 1990.

[4] M. Nielsen, G. Plotkin, G. Winskel, *Petri nets, event structures and domains*, Theoretical Computer Science, **13**, 1981, 85–108

[5] W. Vogler, *Modular construction and partial order semantics of Petri nets*, Lecture Notes in Computer Science, **625**, 1992.

[6] I.B. Virbitskaite, A.V. Votintseva, *Comparing logical and behavioural equivalences for event structures*, Hildesheimer Informatik-Berichte, **27**, 1996.

[7] G. Boudol, I. Castellani, *On the semantics of concurrency: partial orders and transition systems*, Proc. TAPSOFT 87, **I**, Lecture Notes in Computer Science, **249**, 1987, 123–137.

[8] R. Devillers, *On the definition of a bisimulation notion based on partial words*, Petri Net Newsletter, No. 29, 1988, 16–19.

[9] P. Degano, R. De Nicola, U. Montanari, *A distributed operational semantics for CCS based on condition/event systems*, Acta Informatica, **26**, 1988, 59–91.

[10] M. Mukund, P.S. Thiagarajan, *An Axiomatization of Event Structures*, Lecture Notes in Computer Science, **405**, 1989, 143–160.