Causal ambiguity and partial orders in event structures

Rom Langerak, Ed Brinksma * Joost-Pieter Katoen†
University of Twente Friedrich-Alexander University

December 1996

Abstract

Event structure models often have some constraint which ensures that for each system run it is clear what are the causal predecessors of an event (i.e. there is no causal ambiguity). In this contribution we study what happens if we remove such constraints. We define five different partial order semantics that are intentional in the sense that they refer to syntactic aspects of the model. We also define an observational partial order semantics, that derives a partial order from just the event traces.

It appears that this corresponds to the so-called early intentional semantics; the other intentional semantics cannot be observationally characterized. We study the equivalences induced by the different partial order definitions, and their interrelations.

1 Introduction

Prominent models for non-interleaving semantics are the *event structure* models. Event structures have as their basic objects labelled events together with relations representing causality and conflict. Originally event structures were used for giving a semantics to Petri nets [Win80]. They have been also used as a semantics for process algebraic languages like CCS [BC94], CSP [LG91] and LOTOS [Lan92]. Several different types of event structures exist: we mention prime event structures [Win80, Win89], stable event structures [Win89], flow event structures [BC94], and bundle event structures [Lan93, Lan92].

All these models are causally disambiguous, by which we mean the following: if an event has happened, there is exactly one set of causal predecessors of the event, i.e. there is never any ambiguity in deciding which are the causes of an event.
This is an important technical property, especially if one wants to relate an event structure model to the more fundamental model of partially ordered sets (or *posets*). Posets can be

*{langerak,brinksma}@cs.utwente.nl, Department of Computer Science, PO Box 217, 7500 AE Enschede, The Netherlands

†katoen@informatik.uni-erlangen.de, Institut für Informatik VII, Martensstrasse 3, D-91058 Erlangen, Germany

used as the underlying semantics of many different models; for an elaborate motivation of the importance of posets we refer to [Ren93]. Absence of causal ambiguity implies that there is exactly one poset corresponding to a system run.
Posets can be defined in two alternative ways: by referring to the causality representation in the model (we call this *intentional*), or by just referring to the system runs (we call this *observational*). Having corresponding intentional and observational characterizations of the posets is important for relating event structures to other models, where an explicit representation of causality may be absent.

In e.g. stable or bundle event structures the absence of causal ambiguity (this property is called *stability* in [Win89]) is due to a constraint on the model, which roughly says that if there are alternative causes for an event, then these causes should somehow be in conflict.

For certain application areas (e.g. business redesign) it can be argued that this constraint is too restrictive [Fer94]. Therefore the problem this paper addresses is the following: is it possible to define a partial order semantics for an event structure model with causal ambiguity ?

The organization of the paper is as follows. In section 2 we present some event structure models and their relation to posets. Section 3 sketches the problem of causal ambiguity. In section 4 we give five intentional poset definitions, and in section 5 we show that exactly one of them (the so–called early causality) has an observational characterization. In section 6 we look at the induced equivalence relations, and section 7 is for conclusions.

2 Event structures

Event structure models have as their basic ingredient events labelled with actions; an event models the occurrence of its action. Different events can have the same action label, implying that they model different occurrences of the action. Action labels do not play a role in this paper but are important when the model is used e.g. as a semantics for a language. We are in general not interested in the event identities as such (so implicitly we work modulo an event renaming morphism), as the events just serve to identify or distinguish action occurrences. Often we will denote an event by its action label, if no confusion arises.

Two events in a system are said to be in *conflict* if there is no system run in which both events happen. In this paper we will restrict ourselves to the representation of conflict by a binary relation between events. In that case the main difference between the models lies in the way they represent causality.

In prime event structures causality is modelled by a partial order on the set of events. This model is mathematically very elegant and convenient. The drawback is that as a consequence each event has a unique enabling, so if an action can be caused in alternative ways we need to model the action by different events, harmful to the conciseness of models. In addition it may be rather complicated to define some operations on prime event structures, especially parallel synchronization [Vaa89].

For these reasons other models like stable, flow and bundle event structures model causality in a different way. Flow event structures model causality by a flow relation that (contrary to prime event structures) need not be transitive, thereby making it possible for an event to have alternative enablings. However, also for flow event structures parallel synchronization is a bit problematic as it is technically dependent on self-conflicting events (as we argued in [Lan92]).

For this reason in this paper we concentrate on stable and bundle event structures. Both have a constraint in order to exclude causal ambiguity, the removal of which is the theme of this paper. We present both models in a bit more detail. Since concepts, like well-foundedness [Win89], that address problems with infinite sets of events are orthogonal to the issues of this paper and need not bother us here, we conveniently restrict ourselves to finite sets of events.

2.1 Bundle event structures

In bundle event structures [Lan93, Lan92], causality is represented by *bundles*: a bundle is a pair (X, e) with X a set of events and e an event. The set of all bundles is denoted by $\mapsto$ and we denote a bundle (X, e) by $X \mapsto e$.

The meaning of a bundle $X \mapsto e$ is that X is a set of causal conditions for e, in the sense that if e happens, one of the events in X has to have happened before. If several bundles point to e, for each bundle set an event should have happened.

In addition, we demand that for each bundle $X \mapsto e$, all the events in X are in mutual conflict with each other. In this way, if e has happened, exactly one event from X has happened before, so there is no doubt about which are the causal predecessors of e. In the next section we see what happens if we remove this condition.

The definition of bundle event structures:

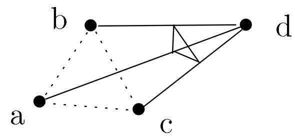
Definition 2.1 A *bundle event structure* $\mathcal{E}$ is a 4-tuple
$\mathcal{E} = (E, \#, \mapsto, l)$ with :

- E a set of *events*
- $\# \subseteq E \times E$, the symmetric and irreflexive *conflict* relation
- $\mapsto \subseteq 2^E \times E$, the *bundle* set
- $l : E \to Act$, the *labelling* function

such that the following property holds:
P1: $X \mapsto e \Longrightarrow \forall e_1, e_2 \in X : (\ e_1 \neq e_2 \Longrightarrow e_1 \# e_2\ )$ □

We represent a bundle event structure graphically in the following way. Events are drawn as dots; near the dot we sometimes give the event name and/or the action. Conflicts are indicated by dotted lines. A bundle $X \mapsto e$ is indicated by drawing an arrow from each element of X to e and connecting all the arrows by small lines.

The following picture is an example of a bundle event structure, with a bundle $\{a, b, c\} \mapsto d$
:



The bundle here means that for d to happen, either a, b or c should have happened already.

The concept of a system run for a bundle event structure is captured by the notion of an *event trace*, which is a conflict–free sequence of events, where each event is preceded by its causal predecessors:

Definition 2.2 Let $\mathcal{E} = (E, \#, \mapsto, l)$ be a bundle event structure. An *event trace* is a sequence of distinct events $e_1, \ldots, e_n$, with $e_1, \ldots, e_n \in E$, satisfying:

- $\{e_1, \ldots, e_n\}$ is conflict-free, i.e. $\forall e_i, e_j : \neg(e_i \# e_j)$.
- $X \mapsto e_i \Longrightarrow \{e_1, \ldots, e_{i-1}\} \cap X \neq \emptyset$

□

Notation: Let $\sigma = e_1 \ldots e_n$ be an event trace, then $\widehat{\sigma} = \{e_1, \ldots, e_n\}$ is the set of events in σ.
With the help of event traces we can define a semantics for bundle event structures in terms of (labelled) partial orders, abbreviated *posets* (not to be confused with *pomsets*, which are equivalence classes of posets modulo event renaming morphisms [Pra86]). Posets form a natural and attractive basic semantics for comparing true concurrency models [Ren93].
The next definition and theorem show how to obtain posets from event traces:

Definition 2.3 Let σ be an event trace of $\mathcal{E}$, with $\widehat{\sigma} = T$. We define the *precedence relation* $\prec_T \subseteq T \times T$ by $e \prec_T e'$ iff $\exists X \subseteq E : (\ e \in X \wedge X \mapsto e'\ )$. The relation $\leq_T$ is defined as $\leq_T = \prec_T^*$, i.e. the reflexive and transitive closure of $\prec_T$. □

Theorem 2.4 $\leq_T$ is a partial order over T.

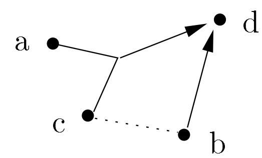
Proof : see [Lan92] □

Let $\mathcal{E}$ be a bundle event structure, then the set of posets we get by applying definition 2.3 to all event traces of $\mathcal{E}$ is denoted by $P(\mathcal{E})$, where P stands for posets.
Other results concerning bundle event structures are transformation laws preserving poset equivalence, a variant with asymmetric conflict for modelling the LOTOS disrupt operator, and a cpo fixed point semantics for recursive processes [Lan92]. Extensions to the model include time [KLLB96a], probabilities [KLL94], and stochastic information [BKLL95], enabling the model to be used for performance modelling [KLLB96b]; see [Kat96] for an overview. A current research interest is the modelling of recursive processes with the help of graph grammars.

2.2 Stable event structures

The first event structure model that was defined in order to allow for multiple enablings is the model of *stable* event structures [Win89]. There causality is represented by a set $\vdash$ of *enablings*, which are pairs (X, e), with X a set of events and e an event, denoted by $X \vdash e$. The interpretation is that e can happen if for some enabling $X \vdash e$ all the events in X have happened already.

Example 2.5 In stable event structure



event d has enablings $\{b\} \vdash d$ and $\{a, c\} \vdash d$, meaning that d can happen after b or after a and c. □

In addition there is a constraint (called the *stability* constraint, from which the model takes its name) demanding that if there are alternative different enablings, these enablings should have some conflict between their events. In this way the set of causal predecessors is always unique, so this constraint prevents causal ambiguity. In the next section we remove this constraint.
There are several slightly different definitions of stable event structures in the literature [Win89]; our definition is a slight variation of the definition in [BC94] [1]

Definition 2.6 A *stable event structure* is a structure $\mathcal{E} = (E, \#, \vdash, l)$ where

- E a set of events
- $\# \subseteq E \times E$ the irreflexive, symmetric conflict relation
- $\vdash \subseteq 2^E \times E$ the enabling relation
- $l : E \rightarrow Act$ the labelling function

such that the following property holds:
P2: $(F \vdash e \ \wedge \ G \vdash e) \implies (F \neq G \Longrightarrow F \cup G$ is not conflict-free) (stability) □

Also for stable event structures there is a definition of event trace:

Definition 2.7 Let $\mathcal{E} = (E, \#, \vdash, l)$ be a stable event structure. An *event trace* is sequence of distinct events $e_1, \ldots, e_n$, with $e_1, \ldots, e_n \in E$, satisfying:

[1] The difference is that there it is also demanded that each enabling $X \vdash e$ is consistent, i.e. $X \cup \{e\}$ is conflict–free. Inconsistent enablings can always be removed (as they have no semantic effect) and then all definitions coincide.

- $\{e_1, \ldots, e_n\}$ is conflict-free, i.e. $\forall e_i, e_j : \neg(e_i \# e_j)$.
- $\forall i \; : \; \exists F \subseteq \{e_1, \ldots, e_{i-1}\} \; : \; F \vdash e_i$

$\square$

We get posets in a similar way as in definition 2.3, only the definition of precedence has to be adapted:

Definition 2.8 Let σ be an event trace of $\mathcal{E}$, with $\widehat{\sigma} = T$. We define the *precedence relation* $\prec_T \subseteq T \times T$ by $e \prec_T e'$ iff $\exists F \subseteq T : (\; e \in F \wedge F \vdash e' \;)$. The relation $\leq_T$ is defined as $\leq_T = \prec_T^*$, i.e. the reflexive and transitive closure of $\prec_T$. $\square$

Theorem 2.9 $\leq_T$ is a partial order over T.

Proof : It is quite easy to adapt the proof for bundle event structures given in [Lan92]. $\square$

Again we denote the set of posets we get by applying definition 2.8 to all event traces of stable event structure $\mathcal{E}$ by $P(\mathcal{E})$.

Each bundle event structure can be transformed into a stable event structure that is equivalent w.r.t. the set of event traces or posets. However, stable event structures are more expressive: for example, there is no bundle event structure with the same set of event traces as the stable event structure in example 2.5.

2.3 Observational partial orders

We have called the above definitions of partial order (obtained from an event trace) *intentional*, as opposed to *observational*, because they refer to aspects of the model: bundles in the case of bundle event structures, and enablings in the case of stable event structures. The bundles or enablings are not observable as such. Therefore the question arises how to relate these partial orders to systems where the only observations that can be made are the event traces. As an answer to this question we give a definition of partial orders from event traces that is only based on event traces and does not need to take recourse to bundles or enablings. We call this definition observational, even though a rather strong notion of observation is assumed, namely the ability to observe events (so the occurrence of actions, instead of just actions).

It is easy to prove that each event trace is a linearization of the partial order we get by definition 2.3 respectively 2.8. This provides the basic intuition for the observational poset definition, which works as follows.
Let σ be an event trace of a bundle or stable event structure $\mathcal{E}$, with set of events $\widehat{\sigma} = T$. Now consider all event traces of $\mathcal{E}$ with the same events as σ and suppose $\{\sigma' \mid \widehat{\sigma'} = T\} = \{\sigma_1, \ldots, \sigma_m\}$.
We associate with each event trace σ_i an ordering $\leq_i$ on its events, which is simply the order of the events in the event trace, so if $\sigma_i = e_{i1} \ldots e_{in}$ then $\leq_i$ is defined by $e_{i1} \leq_i e_{i2} \leq_i \ldots \leq_i e_{in}$.

Now define $\leq_T$ by $\leq_T = \leq_1 \cap \leq_2 \cap \ldots \cap \leq_m$. It is not hard to see that $\leq_T$ is a partial order over T, so $(T, \leq_T)$ is a partially ordered set or poset.
Let $\mathcal{E}$ be a bundle or stable event structure, then the set of posets we get by applying the above definition to all event traces of $\mathcal{E}$ is denoted by $OP(\mathcal{E})$, where OP stands for *observational* posets.

The following theorem states that the intentional and observational definitions coincide:

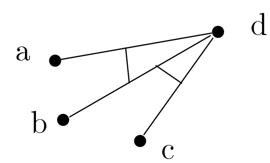
Theorem 2.10 Let $\mathcal{E}$ be a bundle or stable event structure, then $P(\mathcal{E}) = OP(\mathcal{E})$, i.e. the intentional posets are equal to the observational posets.

Proof : See the proof of corol. 7.5.4. in [Lan92] for bundle event structures, which can easily be adapted for stable event structures. □

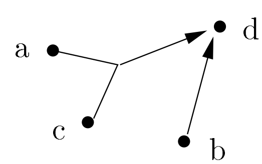
The correspondence between the intentional and the observational definition makes it possible to relate bundle/stable event structures to other models that can be defined to generate event traces, e.g. Petri nets or process algebras [Lan92].

3 The problem of causal ambiguity

In the previous section we saw how to define in a rather straightforward way a partial order semantics for bundle and stable event structures, both in an intentional and in an observational way. Each event trace gives rise to a unique partial order.
Crucial for these definitions are the constraints P1 and P2 (see definition 2.1 respectively definition 2.6), that say that from each bundle only one event can happen, respectively that only one enabling can occur. If we would not have constraint P1, then the following would be a (bundle) event structure:



with bundles $\{a, b\} \mapsto d$ and $\{b, c\} \mapsto d$, corresponding to the following "stable" event structure (if constraint P2 would be removed):



Suppose we would take event trace $abcd$ and would ask what partial order corresponds to this event trace. What are the causal predecessors of d ? With the stability constraint this question always has a unique answer, but now there are several candidates: $\{a, c\}$, $\{b\}$, $\{a, b\}$, $\{b, c\}$ and $\{a, b, c\}$ are all candidate sets of causal predecessors of d. We therefore have to adapt our definition of how to obtain a partial order from an event trace, and in the next section we will see that there are several ways of doing so.

Also the observational definition of the previous section does not work anymore. If we try the recipe given there for the above event structures, we obtain 14 event traces with events $\{a, b, c, d\}$; the intersection of these linear orders is a poset with just the identity as the ordering relation, which surely does not capture the causality information of the event structure.
Providing intentional and observational partial order definitions for event structures without the stability constraints P1 or P2 is the theme of the following sections.

Bundle event structures without constraint P1 have been baptized *dual* event structures in [Kat96]. Stable event structures without constraint P2 have been called just "event structures" in [Win89]; however, this term is also used for an even more general type of event structures. We therefore use the term *instable* event structures for "stable" event structures without the stability constraint.
So the definition of dual and instable event structures becomes:

Definition 3.1 A *dual* event structure $\mathcal{E} = (E, \#, \mapsto, l)$ or *instable* event structure $\mathcal{E} = (E, \#, \vdash, l)$:

- E a set of *events*
- $\# \subseteq E \times E$, the symmetric and irreflexive *conflict* relation
- $\mapsto \subseteq 2^E \times E$, the *bundle* set, respectively
  $\vdash \subseteq 2^E \times E$, the *enabling* relation
- $l : E \to Act$, the *labelling* function

□

Each instable event structure can be transformed into a dual event structure and vice versa. This can be seen by considering all the enablings of an event e in an instable event structure as a disjunction of conjunctions: if there are enablings $\{e_{11}, \ldots, e_{1m}\}$, ..., $\{e_{k1}, \ldots, e_{kn}\}$ for event e, then e can happen if $(e_{11} \wedge \ldots \wedge e_{1m}) \vee \ldots \vee (e_{k1} \wedge \ldots \wedge e_{kn})$ have happened.

Bundles however can be considered as a conjuction of disjunctions: if event e has bundles $\{e_{11}, \ldots, e_{1m}\}$, ..., $\{e_{k1}, \ldots, e_{kn}\}$ pointing to it, then e can happen after $(e_{11} \vee \ldots \vee e_{1m}) \wedge \ldots \wedge (e_{k1} \vee \ldots \vee e_{kn})$.

We can transform a conjunction of disjunctions into a disjunction of conjunctions and vice versa; this means that we can transform dual into instable event structures and vice versa, so the two models are equally expressive (e.g. in terms of event traces). It is in fact quite remarkable that when we add the reasonably-looking constraints P1 and P2 to the models, obtaining respectively bundle and stable event structures, these models have different expressive powers.

In the rest of this paper we use dual event structures as our descriptive vehicle. The above shows that we could have just as well used instable event structures.

4 Intentional partial order definitions

In this section we present several definitions of causality in possibly causally ambiguous situations. What definition is appropriate depends on considerations coming from the application area. In this respect the situation is very similar to the field of implementation relations [vG90], where many different implementation relations exist, each with its own (often observational) justification. In fact in section 6 we show how these different causality notions give rise to different partial order equivalences, and study their interrelations. In section 5 we show that only one of the notions in this section has an observational characterization in terms of event traces similar to what we saw in section 2 (cf. Theorem 2.10).

By a *cause* of e in σ we mean a set of causal predecessors of e, that is a set of events that enable e to happen. Each of the notions in this section gives an answer to the following question: suppose we have a dual event structure $\mathcal{E}$, with an event trace σ, and an event e in σ, what are the possible causes C in σ of e ? We do not demand that C is always unique, i.e. in principle we allow a set $\{C_i\}$ of possible causes as an answer to our question (some notions lead to a unique C though).

We can define partial orders on $\widehat{\sigma}$ in the following way: for each e in σ, choose a cause C_e. Now define for all $e, e' \in \widehat{\sigma}$: $e' \prec e$ iff $e' \in C_e$ and define the ordering relation on $\widehat{\sigma}$ to be the transitive and reflexive closure of $\prec$. If each cause C_e occurs before e in σ (and all notions we consider have this property, in agreement with the common sense idea that causes have to occur before effects) it is easy to see that this definition leads indeed to a partial order.

4.1 Liberal causality

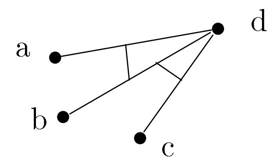
The least restrictive notion of causality, which we call the *liberal* one, is the one saying that each set of events from bundles pointing to e that satisfies all bundles is a cause.

Definition 4.1 *Liberal:* Let σ be an event trace of $\mathcal{E}$, e an event in this trace, and all bundles pointing to e given by $X_1 \mapsto e, \ldots, X_n \mapsto e$.
A set C is a cause of e in σ iff the following conditions hold:

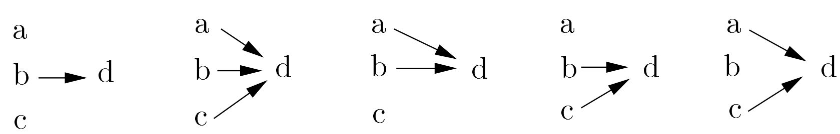
- each $e' \in C$ occurs before e in σ
- $C \subseteq X_1 \cup \ldots \cup X_n$
- for all i: $X_i \cap C \neq \emptyset$

The set of posets obtained in this way from σ is denoted by $P_{lib}(\sigma)$ □

Example 4.2 Consider event trace *abcd* of event structure



Then $P_{lib}(abcd)$ consists of the posets



□

4.2 Bundle satisfaction causality

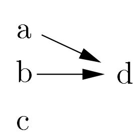
This causality notion is based on the idea that for an e in σ each bundle pointing to e is satisfied by exactly one event in a cause of e. This means that for all bundles pointing to e, each bundle can be mapped to an event in a cause C such that all events in C are being mapped upon, so the presence of each event e' in C should be justified by some bundle $X \mapsto e$, with $e' \in X$, that is associated to e'.

Definition 4.3 *Bundle satisfaction:* Let σ be an event trace of $\mathcal{E}$, e an event in this trace, and all bundles pointing to e given by $X_1 \mapsto e, \ldots, X_n \mapsto e$.
A set C is a cause of e in σ iff the following conditions hold:

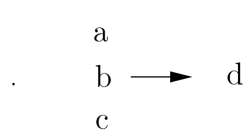
- each $e' \in C$ occurs before e in σ
- There is a surjective mapping $f : \{X_i\} \to C$ such that $f(X_i) \in X_i$

The set of posets obtained in this way from σ is denoted by $P_{bsat}(\sigma)$ □

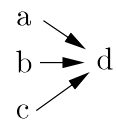
Example 4.4 Let $\mathcal{E}$ be the same dual event structure as in example 4.2. Now we allow e.g.



(where a satisfies bundle $\{a, b\} \mapsto d$ and b satisfies bundle $\{b, c\} \mapsto d$) and

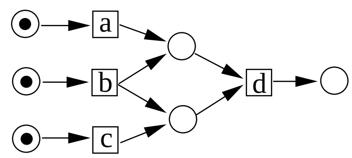


(where b satisfies both bundles $\{a, b\} \mapsto d$ and $\{b, c\} \mapsto d$). Notice that we do allow more events from one bundle, or several bundles satisfied by the same event.



is not allowed as a poset, as d has three causal predecessors and there are only two bundles to be satisfied. □

This notion of causality seems to have a Petri net like intuitive motivation, as illustrated by the following example. This Petri net corresponds to the dual event structure in the previous two examples (except that in the Petri net the d might fire twice, which could be prevented by adding another condition) :



Suppose a, b and c have all fired. Then the conditions of d are both filled with two tokens. If now d fires, it uses two tokens, with the following possibilities: either both tokens from b, or one token from b and one from a, or one from b and one from c, or one from a and one from c. These four possibilities correspond to the four possible causes of d according to definition 4.3, viz. $\{b\}$, $\{a, b\}$, $\{b, c\}$ and $\{a, c\}$. The idea of one token (where each token "remembers" where it was produced) per condition corresponds to the one event per bundle idea of the bundle satisfaction definition of causality.

Clearly each C satisfying definition 4.3 also satisfies definition 4.1, so for all event traces σ, $P_{bsat}(\sigma) \subseteq P_{lib}(\sigma)$.

4.3 Minimal causality

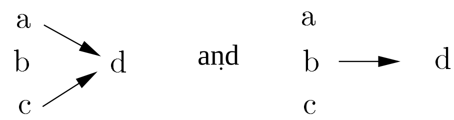
The next causality definition is based on the idea that each cause should be minimal, in the sense that there is no subset which is also a cause.

Definition 4.5 *Minimal:* Let σ be an event trace of $\mathcal{E}$, e an event in this trace, and all bundles pointing to e given by $X_1 \mapsto e, \ldots, X_n \mapsto e$.
A set C is a cause of e in σ iff the following conditions hold:

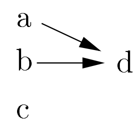
- each $e' \in C$ occurs before e in σ
- for all i: $X_i \cap C \neq \emptyset$
- there is no proper subset of C satisfying the previous two conditions

The set of posets obtained in this way from σ is denoted by $P_{min}(\sigma)$ □

Example 4.6 Let $\mathcal{E}$ be the same dual event structure as in example 4.2. Now the only posets for trace $abcd$ are



E.g.



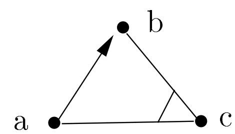
is not allowed anymore as $\{a, b\}$ is not minimal: also the subset $\{b\}$ would be sufficient for d to be enabled. □

Again it is easy to see that each C satisfying definition 4.5 also satisfies definition 4.3, so for all event traces σ, $P_{min}(\sigma) \subseteq P_{bsat}(\sigma)$.

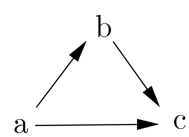
4.4 Early causality

If one is trying to remove "superfluous" events from the causes, at first sight the minimal definition given above seems hard to improve upon. However, look at the following example.

Example 4.7 Consider trace abc from event structure



then $\{a\}$ is a minimal cause of b, and $\{b\}$ is a minimal cause of c, so we have a poset

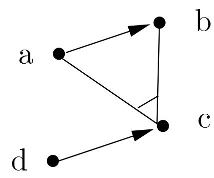


(with $a \leq c$ because of transitivity). However, if b happens, a has happened already, and a is enough to let c happen. So in a sense the causality relation between b and c is superfluous. □

In order to remove this superfluousness, we would like to demand that a cause is somehow the "earliest".

If e_i is an event in trace $\sigma = e_1 \ldots e_n$, we call i its index in σ. A first attempt to define "earlier" would be to say that C is earlier than C' in σ iff the maximum index in σ of the events in C is smaller than the maximum index of the events in C'. However, this will not do, as illustrated by this example:

Consider trace $abdc$ from event structure



then we think $\{a, d\}$ intuitively is an earlier set of predecessors of c than $\{b, d\}$, but this is not captured by the above attempt to define "earlier"".

This suggests that we should compare the sets C and C' on the events that they have not in common, so:

Definition 4.8 Let $\sigma = e_1 \ldots e_n$ be an event trace, and let $C, C' \subseteq \{e_1 \ldots e_n\}$. We say C is earlier than C', notation $C \ll C'$, iff the maximal index in σ of the events in $C \setminus C'$ is smaller than the maximal index in σ of the events in $C' \setminus C$ (we define the maximal index of $\emptyset$ to be 0). □

Lemma 4.9 Let σ be an event trace, let Id be the identity relation over all the subsets of $\widehat{\sigma}$. The relation $\ll \cup Id$ is a total order over all the subsets of $\widehat{\sigma}$.

Proof : Represent a subset C of $\widehat{\sigma}$ by a binary n-digit, where the ith digit is 1 iff $e_i \in C$, the nth digit being the most significant one. Call the resulting number $n(C)$, then it is easy to see that $C \ll C'$ iff $n(C) < n(C')$. □

Given a set of subsets of $\widehat{\sigma}$, lemma 4.9 ensures that it makes sense to talk of a unique earliest element of this set. Now we are ready for the definition of early causality:

Definition 4.10 *Early:* Let σ be an event trace of $\mathcal{E}$, e an event in this trace, and all bundles pointing to e given by $X_1 \mapsto e, \ldots, X_n \mapsto e$.
A set C is a cause of e in σ iff the following conditions hold:

- each $e' \in C$ occurs before e in σ
- for all i: $X_i \cap C \neq \emptyset$
- C is the earliest set satisfying the previous two conditions.

The set of posets obtained in this way from σ is denoted by $P_{early}(\sigma)$ □

Note that due to the uniqueness of the earliest enabling, this definition leads to a unique cause in an event trace σ, and so to a unique poset for σ.

It is easy to check that if $C \subset C'$ then $C \ll C'$; this means that each earliest cause C is also minimal, so for all event traces σ, $P_{early}(\sigma) \subseteq P_{min}(\sigma)$.

4.5 Late causality

In the last section we defined an early causality, taking always the earliest cause. One might ask if it would also be possible to ask for the latest possible cause. Think for instance of a situation where events write values into variables; then it would be natural to consider the last write as a causal predecessor of e.g. an event that reads the variable.

We define C later C' iff $C' \ll C$. Now it is not the case that latest implies minimality (on the contrary, a superset of a set C will always be later). Therefore in the definition of late causality we have to explicitly state that the cause is a minimal one, whereas for early causality this was a consequence.

Definition 4.11 *Late:* Let σ be an event trace of $\mathcal{E}$, e an event in this trace, and all bundles pointing to e given by $X_1 \mapsto e, \ldots, X_n \mapsto e$.
A set C is a cause of e in σ iff the following conditions hold:

- each $e' \in C$ occurs before e in σ
- for all i: $X_i \cap C \neq \emptyset$
- there is no proper subset of C satisfying the previous two conditions
- C is the latest set satisfying the previous three conditions

The set of posets obtained in this way from σ is denoted by $P_{late}(\sigma)$ □

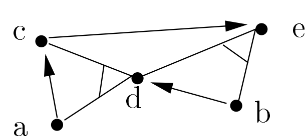
Each C satisfying definition 4.11 trivially satisfies definition 4.5, so for all event traces σ, $P_{late}(\sigma) \subseteq P_{min}(\sigma)$.

4.6 Comparisons

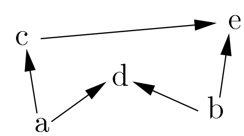
We saw that for each event trace σ, $P_{late}(\sigma), P_{early}(\sigma) \subseteq P_{min}(\sigma) \subseteq P_{bsat}(\sigma) \subseteq P_{lib}(\sigma)$.

We can extend the definition of P_x to dual event structures by having $P_x(\mathcal{E})$ denote the posets of all event traces of event structure $\mathcal{E}$.
The subset relations for the posets of a single event trace carry over to the subset relations for the posets of a dual event structure. The question is whether these relations can be strict. For $\mathcal{E}$ in example 4.2 we have seen that $P_{min}(\mathcal{E}) \subset P_{bsat}(\mathcal{E}) \subset P_{lib}(\mathcal{E})$. For $\mathcal{E}$ in example 4.7 we have that $P_{early}(\mathcal{E}) \subset P_{min}(\mathcal{E})$.
The most difficult case is the late causality, there we have the next somewhat more involved example:

Example 4.12 Let $\mathcal{E}$ be the following dual event structure:



We invite the reader to check that



is a minimal poset for e.g. event trace *abcde*, but cannot be a late poset for any event trace of $\mathcal{E}$. □

So we conclude that all inclusion relations may be strict, i.e. there are no coinciding causality notions.

5 An observational partial order definition

We would like to have also for the dual or instable event structure an observational definition of partial order like the one in section 2.3 (cf. definition 2.3). As illustrated in section 3, we cannot use the technique of reconstructing the posets from their linearizations (the event traces) as we end up with posets that have too little ordering and do not model the causality in a satisfactory way. We therefore try another recipe.

The idea of this definition is the following: for an event e in σ, we look at all event traces with the same events as σ. We then look at the set of predecessors of e in some event trace (we call such a set a *securing* for e). From all these securings we now take the earliest securing for e in σ and define $e' \leq e$ for all e' in this earliest securing.

Definition 5.1 Let σ be an event trace of a dual event structure $\mathcal{E}$, and e an event in σ.

- let $[\sigma]$ be the set of all event traces of $\mathcal{E}$ with events $\widehat{\sigma}$
- the securings of e are defined as $\{\widehat{\sigma_1} | \exists \sigma_2 : \sigma_1 e \sigma_2 \in [\sigma]\}$
- take the earliest securing S in σ and define $e' \leq e$ iff $e' \in S \cup \{e\}$

□

The nice result is that $\leq$ as defined by the observational definition 5.1 is exactly the unique partial order as defined by the intentional one of *early* causality. We prove this using the following lemma:

Lemma 5.2 Let e be an event in event trace σ. Let $C = \{e_1, \ldots, e_n\}$ be the earliest cause of e, and let $S_1, \ldots, S_n$ be the earliest securings of $e_1, \ldots, e_n$. Then:
$S = \bigcup S_i \cup C$ is the earliest securing of e in σ.

Proof : Let S' be an arbitrary securing of e in σ. Then S' contains a cause C' for e.

- Suppose $C' = C$. Then all securings in S' of $e_1, \ldots, e_n$ are equal or later than the corresponding securings in S (as they were earliest), so S is earlier or equal to S'.
- Suppose $C' \neq C$, then the maximal index m of events in C' is higher than for C. Now the events in S' that occur after e_m have to be in securings of elements in $C' \cup C$ and therefore this set is later or equal than the corresponding securings in S'. Since C' is later than C, this shows that S' is later than S.

So we have proven that S is earlier or equal than S'; since S' was arbitrary this shows that S is the earliest securing. □

Let $\leq$ be the ordering defined by definition 5.1, then we write $OP(\sigma)$ (for *observational poset*) for $(\widehat{\sigma}, \leq)$.

Theorem 5.3 Let σ be an event trace of dual event structure $\mathcal{E}$. Then:
$OP(\sigma) = P_{early}(\sigma)$.

Proof : By induction on the length i of prefixes σ_i of σ.

i = 1: Trivial.

Step: Suppose the order of $OP(\sigma_i)$ is $\leq^i$ and of $P_{early}(\sigma_i)$ is $\leq^i_{early}$. Let C be the earliest cause of e_{i+1} in σ. Then we prove:

$\leq^{i+1}$

= (Definition 5.1)

$\leq^i \cup \{(e, e_{i+1}) | e \in S\} \cup \{(e_{i+1}, e_{i+1})\}$

= (induction hypothesis and Lemma 5.2)

$\leq^i \cup \{(e, e_{i+1}) | e \in C)\} \cup \{(e, e_{i+1}) | e \in S_j, 0 < j <= i\} \cup \{(e_{i+1}, e_{i+1})\}$

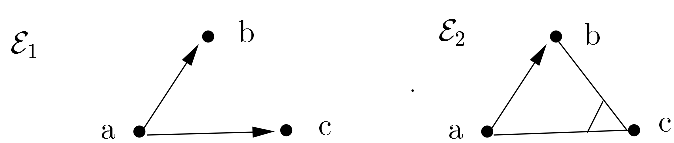
= (C is the earliest cause of e_{i+1} and Definition 4.10)

$\leq^{i+1}_{early}$

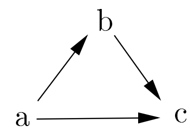
□

So early causality can also be characterized in an observational way. Is it possible to find a characterization for any of the other intentional causality concepts ? The answer is no, as can be learned from the following example.

Example 5.4 The dual event structures



have the same event traces. $\mathcal{E}_2$ has for trace abc the poset



under liberal, bundle satisfaction, minimal and late causality, but this is not a poset of $\mathcal{E}_1$.
□

Any observational definition of causality would have the same result for $\mathcal{E}_1$ and $\mathcal{E}_2$ above as they have the same traces. Since the other intentional causality concepts lead to different posets for $\mathcal{E}_1$ and $\mathcal{E}_2$ this shows that these intentional concepts cannot be observationally characterized.

So the result is that the early causality concept is the only one that can be observationally characterized.

6 Partial order equivalence relations

The causality notions defined in the previous sections induce equivalence relations in the following way:

Definition 6.1 Let $\mathcal{E}_1$, $\mathcal{E}_2$ be dual event structures. We define $\mathcal{E}_1 \approx_x \mathcal{E}_2$ iff $P_x(\mathcal{E}_1) = P_x(\mathcal{E}_2)$, where $x \in \{lib, bsat, min, early, late\}$. □

Now an obvious question is the relation between the different equivalence relations. First of all, we note that due to theorem 5.3, $\approx_{early}$ is equal to event trace equivalence (since equal event traces lead to the same observational posets so to the same early posets, and vice versa).
We first prove the following lemma:

Lemma 6.2 Let $\mathcal{E}$ be a dual event structure. For $x \in \{lib, bsat, min, early, late\}$ the following holds:

1. each linearization of a poset from $P_x(\mathcal{E})$ is an event trace from $\mathcal{E}$

2. each event trace σ from $\mathcal{E}$ is a linearization of each poset in $P_x(\sigma)$

Proof :

1. similar to 7.5.1. in [Lan92]

2. because the ordering of each poset in $P_x(\sigma)$ is contained in the sequence ordering of σ, i.e. $e_i \leq e_j \Longrightarrow i \leq j$,

□

Corollary 6.3 $\mathcal{E}_1 \approx_x \mathcal{E}_2 \implies \mathcal{E}_1 \approx_{early} \mathcal{E}_2$
for $x \in \{lib, bsat, min, late\}$.

Proof : As a consequence of lemma 6.2, $\approx_x$ implies event trace equivalence which equals $\approx_{early}$ due to Theorem 5.3. □

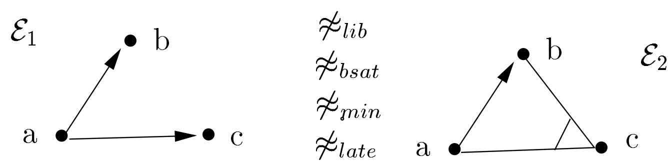
There is one other implication we can prove.

Theorem 6.4 $\mathcal{E}_1 \approx_{bsat} \mathcal{E}_2 \implies \mathcal{E}_1 \approx_{lib} \mathcal{E}_2$.

Proof : We sketch the proof without filling in the details.
Each poset from P_{lib} can be constructed from a poset from P_{bsat} in the following way. Take a poset from P_{bsat} and add to this poset zero or more pairs $e \leq e'$ from other posets in P_{bsat} that are directly ordered, i.e. there is no $e'' \neq e, e'$ such that $e \leq e'' \leq e'$. Since it can be proven that each directly ordered pair $e \leq e'$ of some poset in P_{lib} is contained in some poset of P_{bsat} (which need not hold for the other notions !), in this way we can obtain all and only posets from P_{lib}. So if $\mathcal{E}_1 \approx_{bsat} \mathcal{E}_2$ we get the same sets of liberal posets, i.e. $\mathcal{E}_1 \approx_{lib} \mathcal{E}_2$
□

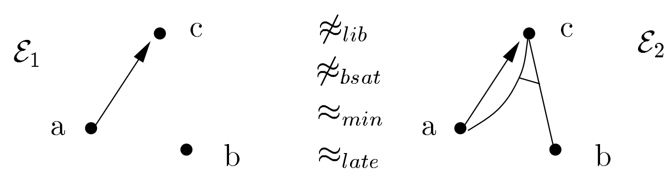
The two above implications are strict (i.e. the reverse does not hold). Moreover, no other implications hold. This can be seen from the following examples, where each pair of dual event structures is event trace equivalent and so early equivalent:

Example 6.5



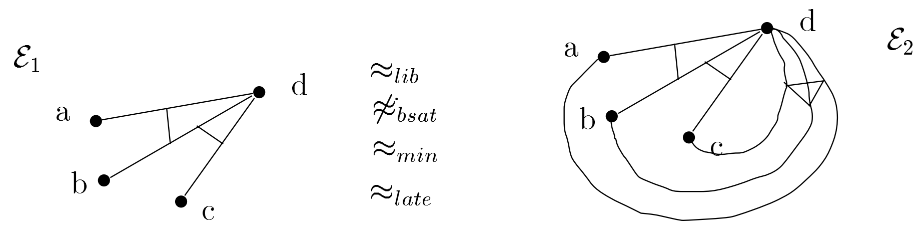
as $\mathcal{E}_2$ can have $b \leq c$ and $\mathcal{E}_1$ can not. □

Example 6.6

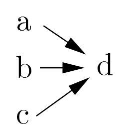


as $\mathcal{E}_2$ can have $b \leq c$ in liberal and bundle satisfaction posets and $\mathcal{E}_1$ can not. For minimal and late causality, b will not be in a cause for c as a is sufficient. □

Example 6.7

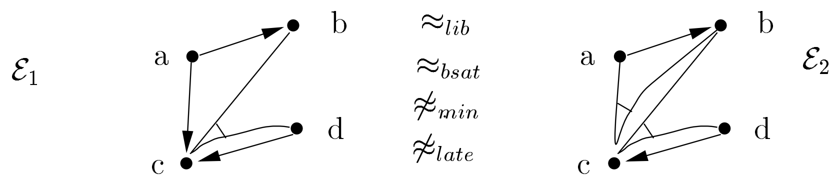


as the extra bundle $\{a, b, c\} \mapsto d$ has no influence on liberal, minimal and late causes, but $P_{bsat}(\mathcal{E}_2)$ has poset

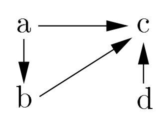


and $P_{bsat}(\mathcal{E}_1)$ has not. □

Example 6.8



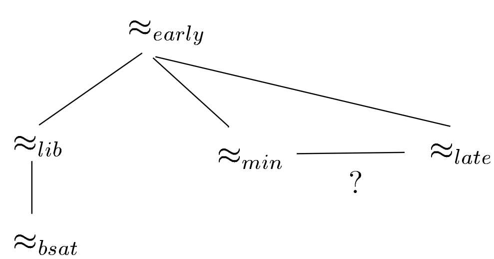
For minimal and late causality, $\mathcal{E}_2$ has poset



as $\{b, d\}$ is a minimal cause for c in e.g. trace $abdc$, which does not hold for $\mathcal{E}_1$.

□

The only relationship we have not been able to clear up is between $\approx_{min}$ and $\approx_{late}$ (note that it is not the case that $P_{min}(\mathcal{E}) = P_{late}(\mathcal{E})$, as illustrated by example 4.12). We have not been able to produce an example of their difference, nor have we been able to prove that such an example does not exist.
If we leave that relation as an open question, we can resume our findings in the following diagram:



7 Conclusion

We have shown that it is possible to give a partial order semantics for a causally ambiguous event structure model. We have presented five intentional causality concepts (that make use of the way causality is represented in the model): liberal, bundle satisfaction, minimal, early and late causality. We have given an observational characterization (that makes use of just event traces) of one of them, namely the *early* causality, and have shown that for the other notions no observational characterization can be given.
Especially the fact that late causality, which at first sight seems a symmetric counterpart to early causality, cannot be observationally characterized is something that we did not expect beforehand.

We studied the induced equivalence relations and found that all equivalences imply early equivalence (which is equal to event trace equivalence), and that bundle satisfaction equivalence implies liberal equivalence.
We gave examples showing that apart from these implications the different equivalences are incomparable, except for the relation between minimal and late equivalence: the relation between these equivalences is an open question.

Another problem for further study would be to look at transformation laws preserving the various equivalences, in a similar way as has been done in [Lan92] for event trace equivalence.

Acknowledgements

This paper owes much to discussions with Rob van Glabbeek and Arend Rensink. Thanks to the anonymous referees for many suggestions for improvement.

References

[BC94] G. Boudol and I. Castellani. Flow models of distributed computations: three equivalent semantics for CCS. *Information and Computation*, 114:247–314, 1994.

[BKLL95] E. Brinksma, J.-P. Katoen, R. Langerak, and D. Latella. A stochastic causality-based process algebra. *The Computer Journal*, 38(7):552–565, 1995.

[dBdRR89] J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors. *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, volume 354 of *Lecture Notes in Computer Science*. Springer-Verlag, 1989.

[Fer94] L. Ferreira Pires. *Architectural Notes: A Framework for Distributed Systems Development*. PhD thesis, University of Twente, 1994.

[Kat96] J-P. Katoen. *Quantitative and Qualitative Extensions of Event Structures*. CTIT Ph. D-thesis series no. 96-09, University of Twente, 1996.

[KLL94] J.-P. Katoen, R. Langerak, and D. Latella. Modelling systems by probabilistic process algebra: An event structures approach. In R.L. Tenney, P.D. Amer, and M.Ü. Uyar, editors, *Formal Description Techniques VI*, volume C–22 of *IFIP Transactions*, pages 253–268. North-Holland, 1994.

[KLLB96a] J-P. Katoen, D. Latella, R. Langerak, and E. Brinksma. On specifying real-time systems in a causality-based setting. In B. Jonsson and J. Parrow, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 1135 of *Lecture Notes in Computer Science*, pages 385–405, Uppsala, Sweden, 1996. Springer-Verlag.

[KLLB96b] J-P. Katoen, D. Latella, R. Langerak, and E. Brinksma. Stochastic simulation of event structures. In M. Ribaudo, editor, *Proceedings of the 4th Workshop on Process Algebra and Performance Modelling*, Torino, Italy, 1996. Università di Torino.

[Lan92] R. Langerak. *Transformations and Semantics for LOTOS*. PhD thesis, University of Twente, 1992.

[Lan93] R. Langerak. Bundle event structures: a non-interleaving semantics for LOTOS. In M. Diaz and R. Groz, editors, *Formal Description Techniques V*, volume C–10 of *IFIP Transactions*, pages 331–346. North-Holland, 1993.

[LG91] R. Loogen and U. Goltz. Modelling nondeterministic concurrent processes with event structures. *Fundamentae Informaticae*, 14:39–74, 1991.

[Pra86] V.R. Pratt. Modeling concurrency with partial orders. *International Journal of Parallel Programming*, 15(1):33–71, 1986.

[Ren93] A. Rensink. *Models and Methods for Action Refinement*. PhD thesis, University of Twente, 1993.

[Vaa89] F.W. Vaandrager. A simple definition for parallel composition of prime event structures. Report CS-R8903, Centre for Mathematics and Computer Science, 1989.

[vG90] Rob van Glabbeek. *Comparative Concurrency Semantics and Refinement of Actions*. PhD thesis, Free University of Amsterdam, 1990.

[Win80] G. Winskel. *Events in Computation*. PhD thesis, University of Edinburgh, 1980. (also available as Technical Report CST-10-80).

[Win89] G. Winskel. An introduction to event structures. In de Bakker et al. [dBdRR89], pages 364–397.