

# A Complete Theory of Deterministic Event Structures<sup>\*</sup>

Arend Rensink

Institut für Informatik, University of Hildesheim

**Abstract.** We present an  $\omega$ -complete algebra of a class of *deterministic event structures*, which are labelled prime event structures where the labelling function satisfies a certain distinctness condition. The operators of the algebra are summation, sequential composition and join. Each of these gives rise to a monoid; in addition a number of distributivity properties hold. Summation loosely corresponds to choice and join to parallel composition, with however some nonstandard aspects.

The space of models is a complete partial order (in fact a complete lattice) in which all operators are continuous; hence minimal fixpoints can be defined inductively. Moreover, the submodel relation can be captured within the algebra by summation ( $x \sqsubseteq y$  iff  $x + y = y$ ); therefore the effect of fixpoints can be captured by an infinitary proof rule, yielding a complete proof system for recursively defined deterministic event structures.

## 1 Introduction

It is generally recognised that *prime event structures* constitute a fundamental partial order model of behaviour, analogous to *synchronisation trees* in the field of interleaving models. In contrast to synchronisation trees however, there is no algebraic theory for prime event structures that (1) is complete for finite structures and (2) extends easily to infinite structures, especially those obtained through fixpoint constructions. In this paper we present such a theory, although not for the full class of event structures but rather for the subclass of *deterministic* ones (where the notion of determinism extends beyond that usually defined for transition systems).

### 1.1 Models

First we give some basic definitions. Throughout the paper we assume a universe of *events*  $\mathbf{E}$ , ranged over by  $d, e, f$ , and a set of *actions*  $\mathbf{A}$ , ranged over by  $a, b, c$ . The following is the standard prime event structure model with general conflict (see Winskel [22]), extended slightly to account for termination.

---

<sup>\*</sup> The research reported in this paper was partially supported by the HCM Cooperation Network “EXPRESS” (Expressiveness of Languages for Concurrency) and the Esprit Basic Research Working Group 6067 CALIBAN (Causal Calculi Based on Nets).

**Definition 1 event structures.** An *event structure* is a tuple  $\langle E, \checkmark, <, C, \ell \rangle$  where

- $E \subseteq \mathbf{E}$  is a set of *events*;
- $\checkmark \subseteq \mathbf{E}$  is a set of *termination events* such that  $E \cap \checkmark = \emptyset$ ; we also denote  $E^\checkmark = E \cup \checkmark$ ;
- $< \subseteq E \times E^\checkmark$  is an irreflexive and transitive *causal ordering relation* which is *finitary* (for all  $e \in E^\checkmark$  the set  $\{d \in E \mid d < e\}$  is finite);
- $C \subseteq \text{Fin}(E^\checkmark)$  is a *consistency predicate* on finite sets of events, which is *subset closed* ( $F \subseteq G \in C$  implies  $F \in C$ ) and such that *all events are consistent* ( $\{e\} \in C$  for all  $e \in E^\checkmark$ ), *termination events are pairwise inconsistent* (for all  $d, e \in \checkmark$ , if  $\{d, e\} \in C$  then  $d = e$ ) and *consistency propagates backwards over causality* (for all  $d, e \in E^\checkmark$ , if  $d < e \in F \in C$  then  $\{d\} \cup F \in C$ );
- $\ell: E \rightarrow \mathbf{A}$  is a *labelling function* (note that termination events are unlabelled).

A *configuration* is a consistent set  $F \in C$  which is *causally closed* ( $d < e \in F$  implies  $d \in F$ ). An *lposet configuration* is a tuple  $p = \langle E_p, \checkmark_p, <_p, \ell_p \rangle$  where  $E_p \subseteq E$  and  $\checkmark_p \subseteq \checkmark$  such that  $E_p^\checkmark$  is a configuration (note that it follows that  $|\checkmark_p| \leq 1$ ), and furthermore,  $<_p = < \cap (E_p \times E_p^\checkmark)$  and  $\ell_p = \ell \upharpoonright E_p$ .

**Notation.** The class of all event structures is denoted  $\mathbf{ES}$  and ranged over by  $\mathcal{E}$ ; the lposet configurations of  $\mathcal{E}$  are collected in  $\mathcal{P}(\mathcal{E})$ . If  $\mathcal{E}$  is a prime event structure then we use  $E_{\mathcal{E}}, \checkmark_{\mathcal{E}}, <_{\mathcal{E}}, C_{\mathcal{E}}$  and  $\ell_{\mathcal{E}}$  to denote the components of  $\mathcal{E}$ . If  $F \notin C_{\mathcal{E}}$  we say that the events in  $F$  are in *conflict*.<sup>2</sup> Furthermore, for all  $d \in E_{\mathcal{E}}^\checkmark$  we use  $\Downarrow_{\mathcal{E}} d := \{e \in E_{\mathcal{E}} \mid e <_{\mathcal{E}} d\}$  to denote the set of proper predecessors of  $d$ .

Briefly, the intuition behind prime event structures is as follows: the causal ordering relation expresses which events are necessary for others to occur, the consistency predicate which events may occur together during a system run. This intuition is formalised by the standard notion of *configurations*, which represent partial runs. The *lposet configurations*, studied in Rensink [19], are basically configurations enriched with the ordering relation and labelling function of the event structure; one may regard these again as (so-called *elementary*) event structures where the consistency predicate is omitted; the intuition is that all events are consistent.

Two prime event structures  $\mathcal{E}, \mathcal{F}$  are *isomorphic*, denoted  $\mathcal{E} \cong \mathcal{F}$ , when there exists a bijection  $\varphi: E_{\mathcal{E}}^\checkmark \rightarrow E_{\mathcal{F}}^\checkmark$  which preserves and reflects causal ordering, consistency and labelling in the standard way. Event structures are always regarded *up to isomorphism*; another way of saying this is that we are actually dealing with *isomorphism classes*, although in practice we will use suitably chosen representatives. For more details see Winskel [22]. The isomorphism classes of the lposet configurations (regarded as elementary event structures) are in fact *pomsets* with termination information; we will call them *pomset configurations*.

<sup>2</sup> An important subclass of event structures are those with *binary conflict*, see e.g. Winskel [23], in which all sets of pairwise non-conflicting events are consistent. In this paper, however, we need the more general notion of conflict.

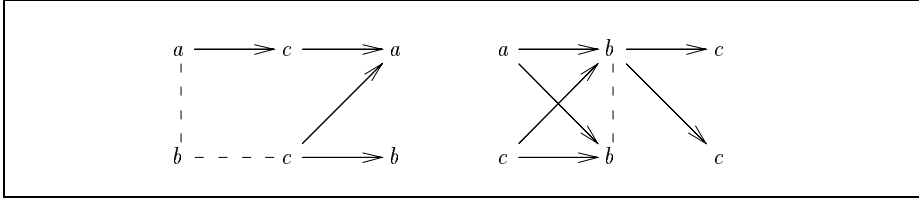


Fig. 1. Event structure examples

**Definition 2.** An event structure  $\mathcal{E}$  is called *deterministic* if for all  $d, e \in E$ :

$$(\Downarrow_{\mathcal{E}} d = \Downarrow_{\mathcal{E}} e) \wedge (d, e \in \sqrt{\mathcal{E}} \vee \ell_{\mathcal{E}}(d) = \ell_{\mathcal{E}}(e)) \implies (d = e) .$$

Hence we call an event structure deterministic if distinct events either have different causal predecessors or distinct labels. This immediately extends to pomsets as well. The class of all deterministic event structures is denoted **DES**.<sup>3</sup>

It should be remarked that the above notion of determinism is less restrictive than one might think. In particular, the resulting models are more general than the deterministic behaviour models studied in [21]. If the ordering relation  $<$  is in fact *sequential*, in the sense that all consistent events are ordered, then determinism in the above sense precisely coincides with the usual notion of determinism on trees. However, consider the left hand event structure in Fig. 1. (Event names are omitted from the picture; the causal ordering is represented by arrows and conflicting events are connected by dotted lines, where for the purpose of this example we assume that conflict is in fact binary.) The left hand structure is easily seen to be deterministic in the sense of Def. 2, but the transition system generated by the corresponding set of configurations is *not* deterministic in the usual sense: for instance, from the configuration consisting of the initial  $a$ -event, there are two different  $c$ -labelled events that are enabled.

On the other hand, the right hand event structure in Fig. 1 is in two places nondeterministic. Namely, the two conflicting  $b$ -events have the same set of predecessors; and indeed this is the traditional form of nondeterminism, where from a given state there are two conflicting ways to perform a certain action. Equationally, such a circumstance is ruled out by requiring  $x; (y+z) = x; y+x; z$ . But also, the two concurrent  $c$ -events have the same set of predecessors and hence also violate the condition in Def. 2; hence certain forms of *auto-concurrency* (concurrent occurrences of the same action) are also ruled out in deterministic event structures —albeit not all forms, as can be seen from the  $c$ -events on the left hand side. In our proof system below, such cases of nondeterminism are ruled out by the analogous axiom  $x; (y \sqcup z) = x; y \sqcup x; z$  (where  $\sqcup$  roughly corresponds with parallel composition).

Another characterisation of determinism is that the *partial runs* of a system completely determine the behaviour. This holds for deterministic event structures as well, if we take as partial runs *pomset configurations*: deterministic event

<sup>3</sup> Under an appropriate notion of morphism, **DES** is a reflective subcategory of **ES**. Details have to be omitted here; see [20] for a similar result on pomsets.

**Table 2.** Theory Ax over signature  $\Sigma = \{\delta, \varepsilon, +, ;, \sqcup\}$

$\delta + x = x$	(1)
$(x + y) + z = x + (y + z)$	(2)
$x + y = y + x$	(3)
$x + x = x$	(4)
$\varepsilon; x = x$	(5)
$x; \varepsilon = x$	(6)
$(x; y); z = x; (y; z)$	(7)
$\varepsilon \sqcup x = x$	(8)
$(x \sqcup y) \sqcup z = x \sqcup (y \sqcup z)$	(9)
$x \sqcup y = y \sqcup x$	(10)
$\delta; x = \delta$	(11)
$x; \delta = x \sqcup \delta$	(12)
$(x + y); z = x; z + y; z$	(13)
$x; (y + z) = x; y + x; z$	(14)
$x \sqcup (y + z) = (x \sqcup y) + (x \sqcup z)$	(15)
$x; (y \sqcup z) = x; y \sqcup x; z$	(16)

structures are basically the same objects as prefix closed sets of deterministic pomsets. (Deterministic pomsets are studied in detail in Rensink [20].)

**Proposition 3.**  $\mathcal{P}$  is a bijection between (isomorphism classes of) deterministic event structures and prefix closed sets of deterministic pomsets.

It should be noted that many event structure variations known from the literature can be regarded as deterministic event structures in which the labelling function maps not to actions but to an intermediate layer of *abstract events*, which are in turn labelled by actions. Alternatively, one may treat the labels of deterministic event structures as such abstract events, and add another layer of labels on top of them; this would allow one to get around the restriction of determinism, although at the price of adding a layer of complexity and almost certainly losing completeness.

## 1.2 Language and Theory

Consider the signature  $\Sigma = \{\delta, \varepsilon, +, ;, \sqcup\}$ , with approximately the following intuition:  $\delta$  denotes *deadlock* and  $\varepsilon$  *successful termination*;  $\mathcal{E} + \mathcal{F}$  expresses the *choice* between  $\mathcal{E}$  and  $\mathcal{F}$ , although isomorphic initial parts are merged together (so determinism is preserved);  $\mathcal{E}; \mathcal{F}$  is the *sequential composition* and  $\mathcal{E} \sqcup \mathcal{F}$  the *join* of two event structures, where the latter is a kind of parallel composition except that isomorphic initial parts are again merged rather than put in parallel. Over this signature we consider the equational theory Ax presented in Table 2. It is seen that each of the operators of  $\Sigma$  gives rise to a monoid, which for summation and join is commutative and idempotent ( $x \sqcup x = x$  is derivable); the neutral elements of sequential composition and join coincide on the constant for

successful termination; the neutral element of summation is the deadlock constant, which is moreover left cancellative for sequential composition whereas its effect as a right operand of sequential composition equals its effect as an operand of join. Furthermore, there is a number of distributivity properties.

If one compares Ax with the standard axiom system of ACP (see Bergstra and Klop [4], Baeten and Weijland [3]) then the difference is that we have Axiom 12 and the distributivity properties in Axioms 14–16. As discussed above, the latter capture the notion of determinism. (It should be remarked, however, that these axioms are crucial to the completeness result of this paper; one cannot simply drop them and so obtain a complete theory of arbitrary event structures.)

### 1.3 Results

The main theorem of this paper, presented in Sect. 2, is that Ax is complete for isomorphism of finite structures in **DES**, and  $\omega$ -complete if **A** is large enough. That is, not only can all finite deterministic event structures be denoted, but also all terms denoting isomorphic event structures are provably equal in Ax.

Furthermore, it is straightforward to generalise the theory to infinite structures. This is worked out in Sect. 3. **DES** is in fact a complete lattice under the submodel relation defined by  $x \sqsubseteq y \Leftrightarrow x + y = y$ . The distributivity properties in Axioms 13–15 already imply that sequential composition and join are monotonic with respect to  $\sqsubseteq$ ; the corresponding semantic operators are in fact continuous. It follows that all context-free equations in  $\Sigma$  have inductively defined least fixpoint solutions. This makes it feasible to add a least fixpoint constructor  $\mu x. t$  to the language (with  $x \in \mathbf{V}$  an arbitrary variable occurring free in the term  $t$ ) and thereby allow the recursive definition of infinite behaviour.

A simple *guardedness* condition on terms characterises those recursive equations with a *unique* fixpoint solution. This provides us with an additional proof rule for such guarded terms  $t$ :

$$\frac{x = t[x/y]}{x = \mu y. t}$$

Finally, the fact that the submodel relation can be characterised within the language allows us some degree of inductive reasoning as well. In particular, for arbitrary  $t$  one may infer:

$$\frac{t_x^i \sqsubseteq y \quad \text{for all } i \in \mathbb{N}}{\mu x. t \sqsubseteq y}$$

(where the  $t_x^i$  denote inductive approximations of  $\mu x. t$ ). The resulting proof system is then  $\omega$ -complete for  $\Sigma$  with recursion.

## 2 Soundness and Completeness

In this section we discuss the first of our main claims: that the theory in Table 2 is complete for isomorphism of deterministic event structures. For this purpose,

we first define a  $\Sigma$ -algebra on deterministic event structures. The constants  $\delta$  and  $\varepsilon$  and actions  $s \in \mathbf{A}$  are modelled by the following deterministic event structures:

$$\begin{aligned}\mathcal{E}_\delta &:= \langle \emptyset, \emptyset, \emptyset, \{\emptyset\}, \emptyset \rangle \\ \mathcal{E}_\varepsilon &:= \langle \emptyset, \{e\}, \emptyset, \{\emptyset, \{e\}\}, \emptyset \rangle \\ \mathcal{E}_a &:= \langle \{d\}, \{e\}, \{(d, e)\}, \{\emptyset, \{d\}, \{e\}, \{d, e\}\}, \{(d, a)\} \rangle \quad (a \in \mathbf{A}).\end{aligned}$$

To define the operators we need some additional notions. If  $\rho \subseteq \mathbf{E}^2$  is a relation over events, then the *domain* of  $\rho$  is given by  $\text{dom } \rho = \{e \in \mathbf{E} \mid \exists d \in \mathbf{E}. (e, d) \in \rho\}$  and the *codomain* by  $\text{cod } \rho = \{e \in \mathbf{E} \mid \exists d \in \mathbf{E}. (d, e) \in \rho\}$ . We denote  $F \rho G$  whenever  $F \subseteq \text{dom } \rho$  and  $G = \{e \mid \exists d \in F. (d, e) \in \rho\}$ . A *prefix relation* between  $\mathcal{E}$  and  $\mathcal{F}$  is a relation  $\rho \subseteq E_{\mathcal{E}} \times E_{\mathcal{F}}$  such that

$$\rho = \{(d, e) \mid ((d, e) \in (\check{\mathcal{E}} \times \check{\mathcal{F}}) \vee \ell_{\mathcal{E}}(d) = \ell_{\mathcal{F}}(e)) \wedge (\Downarrow_{\mathcal{E}} d) \rho (\Downarrow_{\mathcal{F}} e)\} .$$

By induction on the depth, it is straightforward to establish that between arbitrary  $\mathcal{E}, \mathcal{F} \in \mathbf{DES}$  there is a unique prefix relation  $\rho$ , which is *one-to-one* ( $((d, e), (d', e')) \in \rho$  implies  $e = e'$ ) and *injective* ( $((d, e), (d', e')) \in \rho$  implies  $d = d'$ ). In the following, we use a special symbol  $* \notin \mathbf{E}$ , and we assume that  $\mathbf{E}$  is closed under pairing, i.e.,  $(\mathbf{E} \cup \{*\})^2 \subseteq \mathbf{E}$ . For arbitrary  $E \subseteq \mathbf{E}$  and  $i = 1, 2$  we use  $\pi_i(E) = \{e_i \mid (e_1, e_2) \in E\}$ .

**Definition 4 summation.** Let  $\mathcal{E}, \mathcal{F} \in \mathbf{DES}$  and let  $\rho$  be the prefix relation between them. The *sum* of  $\mathcal{E}$  and  $\mathcal{F}$  is given by  $\mathcal{E} + \mathcal{F} = \langle E, \check{\mathcal{E}}, <, C, \ell \rangle$  where

$$\begin{aligned}E &= ((E_{\mathcal{E}} \setminus \text{dom } \rho) \times \{*\}) \cup (\{*\} \times (E_{\mathcal{F}} \setminus \text{cod } \rho)) \cup (\rho \cap (E_{\mathcal{E}} \times E_{\mathcal{F}})) \\ \check{\mathcal{E}} &= ((\check{\mathcal{E}} \setminus \text{dom } \rho) \times \{*\}) \cup (\{*\} \times (\check{\mathcal{F}} \setminus \text{cod } \rho)) \cup (\rho \cap (\check{\mathcal{E}} \times \check{\mathcal{F}})) \\ < &= \{((d, e), (d', e')) \in E \times E^\vee \mid (d <_{\mathcal{E}} d' \vee d' = *) \wedge (e <_{\mathcal{F}} e' \vee e' = *)\} \\ C &= \{F \subseteq E^\vee \mid \pi_1(F) \in C_{\mathcal{E}} \vee \pi_2(F) \in C_{\mathcal{F}}\} \\ \ell &= \{((d, e), a) \in E \times \mathbf{A} \mid a = \ell_{\mathcal{E}}(d) \vee a = \ell_{\mathcal{F}}(e)\} .\end{aligned}$$

This basically corresponds to the usual definition of choice on event structures, except that where usually one requires  $E_{\mathcal{E}} \cap E_{\mathcal{F}} = \emptyset$ , here we *merge* parts of the operands  $\mathcal{E}$  and  $\mathcal{F}$ , viz. those parts that are related by  $\rho$ . If  $\rho = \emptyset$  then the definition is in fact precisely that of choice. The following is the natural definition of sequential composition on event structures, where a copy of the second operand is created and appended at every exit point of the first operand.

**Definition 5 sequential composition.** Let  $\mathcal{E}, \mathcal{F} \in \mathbf{DES}$ . The *sequential composition* of  $\mathcal{E}$  and  $\mathcal{F}$  is given by  $\mathcal{E}; \mathcal{F} = \langle E, \check{\mathcal{E}}, <, C, \ell \rangle$  where

$$\begin{aligned}E &= (E_{\mathcal{E}} \times \{*\}) \cup (\check{\mathcal{E}} \times E_{\mathcal{F}}) \\ \check{\mathcal{E}} &= \check{\mathcal{E}} \times \check{\mathcal{F}} \\ < &= \{((d, e), (d', e')) \in E \times E^\vee \mid d <_{\mathcal{E}} d' \vee (d = d' \wedge e <_{\mathcal{F}} e')\} \\ C &= \{F \subseteq E^\vee \mid \pi_1(F) \in C_{\mathcal{E}} \wedge \pi_2(F) \setminus \{*\} \in C_{\mathcal{F}}\} \\ \ell &= \{((d, e), a) \in E \times \mathbf{A} \mid a = \ell_{\mathcal{E}}(d) \vee a = \ell_{\mathcal{F}}(e)\} .\end{aligned}$$

Finally, we define the *join* of two event structures, which again merges the  $\rho$ -related parts. The difference with summation is twofold: in the join of two structures, two events are conflicting if its projection is conflicting in *either* of its operands rather than *both* its operands as for choice; and termination events are synchronised. If  $\rho = \emptyset$  then join coincides with the standard definition of *parallel composition* (without synchronisation) of event structures.

**Definition 6 join.** Let  $\mathcal{E}, \mathcal{F} \in \mathbf{DES}$  and let  $\rho$  be the unique prefix relation between them. The *join* of  $\mathcal{E}$  and  $\mathcal{F}$  is given by  $\mathcal{E} \sqcup \mathcal{F} = \langle E, \checkmark, <, C, \ell \rangle$  where

$$\begin{aligned} E &= ((E_{\mathcal{E}} \setminus \text{dom } \rho) \times \{*\}) \cup (\{*\} \times (E_{\mathcal{F}} \setminus \text{cod } \rho)) \cup (\rho \cap (E_{\mathcal{E}} \times E_{\mathcal{F}})) \\ \checkmark &= \checkmark_{\mathcal{E}} \times \checkmark_{\mathcal{F}} \\ < &= \{((d, e), (d', e')) \in E \times E' \mid (d <_{\mathcal{E}} d' \vee d' = *) \wedge (e <_{\mathcal{F}} e' \vee e' = *)\} \\ C &= \{F \cup G \cup H \mid F, G \subseteq E \wedge H \subseteq \checkmark \wedge \pi_1(F \cup H) \in C_{\mathcal{E}} \wedge \pi_2(G \cup H) \in C_{\mathcal{F}}\} \\ \ell &= \{((d, e), a) \in E \times \mathbf{A} \mid a = \ell_{\mathcal{E}}(d) \vee a = \ell_{\mathcal{F}}(e)\} . \end{aligned}$$

Note that these operations extend to lposets if we ignore the consistency predicate. The following properties may provide more insight into these operations, and will be important in the next section:

$$\mathcal{P}(\mathcal{E} + \mathcal{F}) = \mathcal{P}(\mathcal{E}) \cup \mathcal{P}(\mathcal{F}) \quad (17)$$

$$\mathcal{P}(\mathcal{E}; \mathcal{F}) = \{p; q \mid p \in \mathcal{P}(\mathcal{E}), q \in \mathcal{P}(\mathcal{F})\} \quad (18)$$

$$\mathcal{P}(\mathcal{E} \sqcup \mathcal{F}) = \{p \sqcup q \mid p \in \mathcal{P}(\mathcal{E}), q \in \mathcal{P}(\mathcal{F})\} . \quad (19)$$

In fact, due to Prop. 3, these equations in themselves completely characterise the operations. This establishes the interpretation of  $\Sigma$ . It is now important to distinguish carefully between objects (event structures) and their denotations (terms of  $\Sigma$ ).  $T_{\Sigma}(\mathbf{V})$ , ranged over by  $s, t$ , will denote the set of  $\Sigma$ -terms on generators  $\mathbf{A}$  and variables  $\mathbf{V}$ , and  $T_{\Sigma}$  the corresponding set of *ground terms*, i.e. terms without variables. A *substitution* is a function  $\sigma: \mathbf{V} \rightarrow T_{\Sigma}(\mathbf{V})$ , usually denoted by a list  $s/x, t/y$  etc. of the non-identity substitutions.  $\sigma$  is called *ground* if its images are ground terms. Substitutions inductively give rise to functions  $\sigma: T_{\Sigma}(\mathbf{V}) \rightarrow T_{\Sigma}(\mathbf{V})$  (note the overloading of the symbol  $\sigma$ ); applications of the latter are postfix denoted, e.g.  $t\sigma$ . The *semantics* of  $\Sigma$  is expressed by a function  $\llbracket \_ \rrbracket: T_{\Sigma} \rightarrow \mathbf{DES}$  defined inductively on the structure of ground terms.

*Example 1.*

1. The left hand event structure in Fig. 1 is generated by  $((c; b \sqcup (a; c \sqcup c); a) + b); \delta$ ;
2. The term  $(a \sqcup b) + (b \sqcup c) + (a \sqcup c)$  yields an event structure  $\mathcal{E}$  where conflict is not binary: if  $\ell_{\mathcal{E}}(e_x) = x$  then  $\{e_a, e_b\}, \{e_b, e_c\}, \{e_a, e_c\} \in C_{\mathcal{E}}$  but  $\{e_a, e_b, e_c\} \notin C_{\mathcal{E}}$ .

The following *soundness* theorem states that the semantics is well-behaved in that it maps to the intended class of models (deterministic event structures) and preserves provable equality as event structure isomorphism; in other words, that **DES** is indeed a model of Ax.

**Theorem 7 soundness.** For arbitrary  $s, t \in T_{\mathcal{Y}}$ ,  $\llbracket t \rrbracket \in \mathbf{DES}$  and  $\text{Ax} \vdash s = t$  implies  $\llbracket s \rrbracket = \llbracket t \rrbracket$ .

Next, we state that all the objects of our model can be denoted. For the proof, the following meta-notation is convenient:  $\sum T$  for finite sets  $T \subseteq T_{\mathcal{Y}}$  stands for the sum of all  $t \in T$ , where  $\sum \emptyset = \delta$  and  $\sum \{t\} = t$ . This meta-notation is well-defined up to provable equality of terms, due to the fact that  $+$  is commutative, associative and idempotent with identity  $\delta$  (Axioms 1-4). Similarly,  $\bigsqcup T$  for finite  $T$  stands for the *join* of all  $t \in T$ , where  $\bigsqcup \emptyset = \varepsilon$ . Now we inductively define functions  $R, S: \mathbf{DES} \rightarrow T_{\mathcal{Y}}$  as follows:

$$\begin{aligned} R(\mathcal{E}) &:= \sum_{p \in \mathcal{P}(\mathcal{E}), |\surd_p|=0} S(p); \delta + \sum_{p \in \mathcal{P}(\mathcal{E}), |\surd_p|=1} S(p) \\ S(\mathcal{E}) &:= \bigsqcup_{e \in E_{\mathcal{E}}} S(\mathcal{E} \upharpoonright \downarrow_{\mathcal{E}} e); \ell_{\mathcal{E}}(e) . \end{aligned}$$

$S: \mathbf{DES} \rightarrow T_{\mathcal{Y}}$  is only defined on elementary event structures (where  $E_{\mathcal{E}} \in C_{\mathcal{E}}$ ) such that  $\llbracket S(\mathcal{E}); \delta \rrbracket = \mathcal{E}$ . Hence  $R$  decomposes  $\mathcal{E}$  into the sum of all its lposet configurations, which in turn are constructed as the join of their elements. The following theorem states that this yields denotations for all finite structures.

**Theorem 8 no junk.**  $\mathcal{E} = \llbracket R(\mathcal{E}) \rrbracket$  for all finite  $\mathcal{E} \in \mathbf{DES}$ .

Finally, we state that  $\text{Ax}$  is strong enough to prove all equalities that hold in the model.

**Theorem 9 no confusion.** For all  $s, t \in T_{\mathcal{Y}}$ , if  $\llbracket s \rrbracket = \llbracket t \rrbracket$  then  $\text{Ax} \vdash s = t$ .

As usual, this theorem is proved by rewriting terms to *normal forms*.

**Definition 10 normal forms.** A term  $t \in T_{\mathcal{Y}}$  is in *elementary normal form* if it is of the form  $t = (\bigsqcup T); a$ , where  $T$  is a *closed set of elementary normal form terms* in the sense that if  $(\bigsqcup T'); a' \in T$  then  $T' \subseteq T$ . Furthermore,  $t$  is in *normal form* if it is of the form

$$t = \sum_{m \in M} (\bigsqcup T_m); \delta + \sum_{n \in N} (\bigsqcup T_n) .$$

where  $N \subseteq M$  and for all  $m \in M$ ,  $T_m$  is a closed set of elementary form terms such that if  $T \subseteq T_m$  is a closed set of elementary form terms then  $T = T_k$  for some  $k \in M$ .

A brief explanation is called for. An elementary normal form  $(\bigsqcup T); a$  correspond to *topped deterministic pomsets*, i.e., with a greatest element: in fact, the subterm  $\bigsqcup T$  yields the pomset minus its greatest element, whereas  $a$  is the label of the top element. The closure of  $T$  is necessary to obtain uniqueness. It is a fact that every finite deterministic pomset can be obtained as the finite join of such “topped” ones; see Rensink [20]. Normal forms consist of a set of deterministic pomsets (first component), a subset of which is terminated (second component).

The function  $R$  used in the proof of Th. 8 in fact yields normal forms; moreover,  $R$  is left inverse of the semantic mapping on normal forms. It follows that there is at most one normal form term describing a given event structure.



**Lemma 11 normal forms are unique.**  $R(\llbracket t \rrbracket) = t$  for all normal forms  $t$ .

It follows that syntactically different normal form terms yield different pomsets, which is one of the two crucial properties of normal forms. The second crucial property is that every term can be rewritten up to provable equality to a normal form term. To see that this holds, we define a function  $norm: T_{\Sigma} \rightarrow T_{\Sigma}$  which constructs normal forms. Assume that  $norm(t_i) = \sum_{m \in M_i} (\bigsqcup T_m); \delta + \sum_{n \in N_i} (\bigsqcup T_n)$  for  $i = 1, 2$ ; then  $norm$  is defined inductively as follows

$$\begin{aligned}
norm(\delta) &:= \sum \{\varepsilon; \delta\} + \sum \emptyset \\
norm(\varepsilon) &:= \sum \{\varepsilon; \delta\} + \sum \{\varepsilon\} \\
norm(a) &:= \sum \{(\bigsqcup \{\varepsilon; a\}); \delta\} + \sum \{\bigsqcup \{\varepsilon; a\}\} \\
norm(t_1 + t_2) &:= \sum_{m \in M_1 \cup M_2} (\bigsqcup T_m); \delta + \sum_{n \in N_1 \cup N_2} (\bigsqcup T_n) \\
norm(t_1; t_2) &:= \sum_{m_1 \in M_1, n_1 \in N_1} (\bigsqcup T_{m_1} \cup \{norm'((\bigsqcup T_{n_1}); s) \mid s \in T_{m_2}\}); \delta \\
&\quad + \sum_{n_2 \in N_2} (\bigsqcup \{norm'((\bigsqcup T_{n_1}); s) \mid s \in T_{n_2}\}) \\
norm(t_1 \sqcup t_2) &:= \sum_{m_i \in M_i} (\bigsqcup T_{m_1} \cup T_{m_2}); \delta + \sum_{n_i \in N_i} (\bigsqcup T_{n_1} \cup T_{n_2})
\end{aligned}$$

Here,  $norm': T_{\Sigma} \rightarrow T_{\Sigma}$  is an auxiliary function inductively defined on terms of the form  $t; s$  where  $t = \bigsqcup T_t$  ( $T_t$  a closed set of elementary normal forms) and  $s = (\bigsqcup T_s); a$  is an elementary normal form:

$$norm'(t; s) = (\bigsqcup \{norm'(t; s') \mid s' \in T_s\}); a$$

It can be proved by induction on the term structure that for such  $t$  and  $s$ ,  $norm'$  always yields an elementary normal form term such that  $Ax \vdash t; s = norm'(t; s)$ , and that for all  $t \in T_{\Sigma}$ ,  $norm(t)$  yields a normal form term provably equal to  $t$ .

**Lemma 12 normal forms exist.** For all terms  $t \in T_{\Sigma}$ ,  $norm(t)$  is in normal form and  $Ax \vdash t = norm(t)$ .

Theorem 9 is then proved as follows: if  $\llbracket s \rrbracket = \llbracket t \rrbracket$  for two terms  $s, t \in T_{\Sigma}$  then Lemma 12 and Lemma 11 imply  $Ax \vdash s = norm(s) = R(\llbracket s \rrbracket) = R(\llbracket t \rrbracket) = norm(t) = t$ .

If there are enough elements around ( $\mathbf{A}$  is large enough) then not only the completeness property of Th. 9 holds, but one which is even stronger, viz. *completeness for open terms*. This is the property that if two open terms denote the same object *under arbitrary ground substitutions* then they are provably equal *before substitution*. See e.g. [12, 13, 14] for a general discussion.

**Theorem 13 Ax is  $\omega$ -complete.** Assume  $|\mathbf{A}| = \omega$ . For all  $s, t \in T_{\Sigma}(\mathbf{V})$ , if  $\llbracket s\sigma \rrbracket = \llbracket t\sigma \rrbracket$  for all ground substitutions  $\sigma: \mathbf{V} \rightarrow T_{\Sigma}$  then  $Ax \vdash s = t$ .

The side condition  $|\mathbf{A}| = \omega$  is needed to ensure that for any pair of terms  $s, t \in T_{\Sigma}(\mathbf{V})$  there are enough “unused elements,” i.e. not occurring in  $s$  or  $t$ , to “encode” the free variables of  $t$ . For instance, if  $|\mathbf{A}| = 1$  then  $Ax$  is not  $\omega$ -complete. The deterministic event structures over a one-element action set are

in fact sequential, and hence for instance  $x; y = y; x$  holds under all ground substitutions. However, this equations are not provable in Ax (and in fact does not hold in general), hence  $\omega$ -completeness does not hold.

To prove  $\omega$ -completeness one can apply the technique described by Heering in [13] and by Lazrek, Lescanne and Thiel in [14], which uses *open* normal forms with the following properties:

- for any open term there is an open normal form that is provably equal to it;
- for any pair of different open normal forms there is a “characteristic” ground substitution that maps them to (closed) terms denoting different objects.

In our case, open normal forms are a simple variation on Def. 10 in which variables  $x$  are treated in the exact same way as actions  $a$ . Since Ax does not distinguish between variables and actions, the first step of the  $\omega$ -completeness proof is immediate. The characteristic substitution required in the second step is obtained by mapping every variable to a distinct new element.

### 3 Infinite and Recursive Behaviour

The signature  $\Sigma$  only allows to express finite behaviour. For the description of infinite behaviour there are basically two mechanisms known from the literature: a constructor for unbounded repetition (the *Kleene star*) and a constructor for (least or greatest) fixpoints with respect to some ordering. We investigate the latter. It should be noted that this section mostly relies on standard theory, the applicability of which follows from the developments above.

The ordering on which the fixpoints will be based is generated by summation:

$$x \sqsubseteq y :\Leftrightarrow x + y = y \ .$$

In terms of deterministic event structures, this implies that  $\mathcal{E} \sqsubseteq \mathcal{F}$  iff the prefix relation  $\rho$  between  $\mathcal{E}$  and  $\mathcal{F}$  is total on  $E_{\mathcal{E}}^{\setminus}$ . Taking (17) into account, it follows that for arbitrary deterministic event structures  $\mathcal{E}, \mathcal{F}$

$$\mathcal{E} \sqsubseteq \mathcal{F} \iff \mathcal{P}(\mathcal{E}) \subseteq \mathcal{P}(\mathcal{F}) \ .$$

Standard order theory implies that the space of nonempty prefix closed sets of deterministic pomsets is a complete lattice under the subset relation, with unions as suprema, intersections as infima, and the singleton set consisting of the empty pomset as the bottom element. Due to Prop. 3 it follows that **DES**, too, is a complete lattice under  $\sqsubseteq$  as defined above, with bottom element  $\mathcal{E}_\delta$ . The supremum of  $\{\mathcal{E}_i\}_{i \in I}$  can be regarded as the infinite sum  $\sum_{i \in I} \mathcal{E}_i$ . The properties (17)–(19) then imply that all operators of  $\Sigma$  are continuous w.r.t.  $\sqsubseteq$ . As a consequence, arbitrary open terms, when regarded as functions over **DES**, are continuous in all variables and hence give rise to inductively definable least (and greatest) fixpoints. The *approximations of  $t$  in  $x$*  are defined as usual:  $t_x^0 := \delta$  and  $t_x^{i+1} := t[t_x^i/x]$  for all  $i \in \mathbb{N}$ .

**Theorem 14 least fixpoints.** For all  $t \in T_{\Sigma}(\{x\})$ ,  $\mathcal{E} = \sum_{i \in \mathbb{N}} \llbracket t_x^i \rrbracket$  is the  $\sqsubseteq$ -smallest deterministic event structure such that  $\llbracket t \rrbracket(\mathcal{E}) = \mathcal{E}$  (where  $\llbracket t \rrbracket$  is interpreted as a function in  $x$ ).

To be able to denote such behaviours (without having to introduce infinitary sums) one can use the extended signature  $\Sigma_{\mu} = \Sigma \cup \{\mu x\}_{x \in \mathbf{V}}$  where the  $\mu x$  are second-order fixpoint constructors whose effect on event structures is defined by

$$\llbracket \mu x . t \rrbracket := \sum_{i \in \mathbb{N}} \llbracket t_x^i \rrbracket . \quad (20)$$

To reason equationally about fixpoint terms, one may first of all add the axiom of unfolding:

$$\mu x . t = t[\mu x . t/x] . \quad (21)$$

The validity follows immediately from (20) in combination with the definition of the approximations  $t_x^i$ . However, the resulting proof system is not complete; for instance the (valid) equation

$$a; (\mu x . b; a; x) = \mu y . a; b; y \quad (22)$$

cannot be proved using unfolding. There are two (standard) ways in which the power of the proof system can be extended further. The first relies on *uniqueness of fixpoint solutions*; it has the advantage that reasoning remains finite, but it does not restore completeness.

**Definition 15 guardedness.** Let  $t, u \in T_{\Sigma_{\mu}}(\mathbf{V})$  with  $x \in \mathbf{V}$ .

- $\delta$  and  $a$  are guarded in  $x$  ( $a \in \mathbf{A}$  arbitrary);
- $\varepsilon$  and  $y$  are not guarded in  $x$  ( $y \in \mathbf{V}$  arbitrary);
- $t + u$  and  $t \sqcup u$  are guarded in  $x$  iff both  $t$  and  $u$  are;
- $t; u$  is guarded in  $x$  iff  $t$  is;
- $\mu y . t$  is guarded in  $x$  iff  $t$  is guarded in  $x$  and  $y$ .

One can prove that if  $t$  is guarded in  $x$ , then  $\llbracket t \rrbracket(\mathcal{E})$  is completely determined up to depth  $n$  by the fragment of  $\mathcal{E}$  up to depth  $n - 1$ . But then the minimal fixpoint  $\mu x . t$  and the *maximal* fixpoint  $\nu x . t$  (which is defined in an analogous fashion by approximating from *above*, i.e., starting with the top element of **DES**) are identical up to arbitrary depth, hence identical; hence the fixpoint equation  $x = t[x/y]$  has  $x = \mu y . t$  as its unique solution. But then the following rule holds:

$$\frac{x = t[x/y] \quad (y \text{ guarded in } t)}{x = \mu y . t} \quad (23)$$

Indeed, this suffices to prove (22) above: (21) together with Axiom 7 implies  $a; (\mu x . b; a; x) = a; b; (a; (\mu x . b; a; x)) = a; b; (a; (\mu x . b; a; x))$ , and hence (23) is applicable for  $t = a; b; y$ . Unfortunately, the proof system continues to be incomplete. To obtain completeness for  $T_{\Sigma_{\mu}}$  we have to allow infinitary reasoning. One way is to introduce infinitary *terms*, in particular the infinite sums shown above. This in fact gets us more than we wanted, since in principle *every* infinite deterministic event structure can then be denoted, rather than just the recursively

defined ones we are considering. A more restricted idea is to express the fact that  $\mu x. t$  is the  $\sqsubseteq$ -infimum of its approximations, in the form of the following infinitary proof rule:

$$\frac{s[t_x^i/y] \sqsubseteq z \quad \text{for all } i \in \mathbb{N}}{s[\mu x. t/y] \sqsubseteq z} \quad (24)$$

Here  $s$  and  $t$  are arbitrary, i.e., not necessarily guarded. The validity of (24) is immediate, since summation distributes over all other operands, and hence  $s[\mu x. t/y] = s[\sum_{i \in \mathbb{N}} t_x^i/y] = \sum_{i \in \mathbb{N}} s[t_x^i/y]$ . (Note: the dual statement  $s[t_x^i/y] \sqsubseteq s[\mu x. t/y]$  can be derived from (21) by induction on  $i$ .) The resulting proof system  $\text{Ax}_\mu$ , consisting of  $\text{Ax}$  and (24), is  $(\omega)$ -complete for equality of terms in  $T_{\Sigma_\mu}$ .

**Theorem 16**  $\text{Ax}_\mu$  is sound and complete. *For all  $s, t \in T_{\Sigma_\mu}$ ,  $\llbracket s \rrbracket = \llbracket t \rrbracket$  if and only if  $\text{Ax}_\mu \vdash s = t$ .*

**Proof sketch for completeness.** To prove  $s = t$ , prove  $s \sqsubseteq t$  and  $t \sqsubseteq s$  separately by approximating the outermost fixpoint operators in respectively  $s$  and  $t$ , proving the inequalities for the resulting terms, and applying (24). To see that such inequality proofs  $s \sqsubseteq t$  can always be carried through, bear in mind that for finite  $s$ , i.e. not containing fixpoints,  $s \sqsubseteq t$  iff  $s \sqsubseteq t'$  for some  $t'$  obtained by substituting high enough approximations for all the fixpoint subterms of  $t$ . By induction on the number of fixpoints in  $s$  it then follows that  $s \sqsubseteq t$  is provable for arbitrary  $s$ .  $\square$

The next theorem states  $\omega$ -completeness. This is independent of Th. 16 since it inherits the assumption in Th. 13 that there are enough labels in  $\mathbf{A}$ ; however, the proof strategy is identical.

**Theorem 17**  $\text{Ax}_\mu$  is  $\omega$ -complete. *Assume  $|\mathbf{A}| = \omega$ . For all  $s, t \in T_{\Sigma_\mu}(\mathbf{V})$ , if  $\llbracket s\sigma \rrbracket = \llbracket t\sigma \rrbracket$  for all ground substitutions  $\sigma: \mathbf{V} \rightarrow T_{\Sigma_\mu}$  then  $\text{Ax}_\mu \vdash s = t$ .*

## 4 Concluding Remarks

We briefly review the results we have achieved and try to put them into a somewhat wider context. In addition, we compare our work with several existing results.

### 4.1 Evaluation

The results we have achieved, consisting of an  $\omega$ -complete theory for a particular class of event-based models including recursively defined ones, are rather strong but will be hard to extend or transfer to other classes of models. They are strong because they represent about the best one can expect in equational reasoning; in fact we do not know of other  $\omega$ -completeness results for similar (event-based or causality-based) models. They will be hard to extend because the operators we have used are quite specific for the type of model we have studied. Especially the restriction to *deterministic* models is rather severe but also quite essential to our

setup. In the introduction we have already suggested that by adding a layer of labelling one can in a sense get around this restriction, but probably only at the price of losing completeness. A similar remark can be made concerning the fact that our models have general, rather than binary, conflict: to restrict to binary conflict does not at all seem easy.

Because of the above, we regard the work presented in this paper in a sense as an “end point,” presenting somehow the maximum that can be achieved using this particular approach. However, the algebra itself lends itself to further study. For instance, one may try to characterise other, weaker equivalences such as history preserving bisimulation (cf. Rabinovich and Trakhtenbrot [18], Van Glabbeek and Goltz [10]) by adding axioms to  $Ax_\mu$ ; or, regarding our labels  $\mathbf{A}$  as abstract events rather than actions, one may study the differences between various types of event-based models such as (for example) flow or stable event structures (cf. [6, 22]) in terms of  $Ax_\mu$ .

## 4.2 Comparison

Let us review a few comparable results in the literature. Quite close to this paper is the theory of *series-parallel pomsets*; cf. Grabowski [11], Pratt [17], Gischer [9], Aceto [2]. The syntactic difference with our theory is minimal: they do not have Axiom 16 and additionally have  $x;\delta = \delta$ . However, the resulting class of pomsets is quite different, and their theory deals with *augmentation* closed rather than *prefix* closed sets of pomsets. Indeed, outside the class of deterministic pomsets the prefix relation is not so well-behaved, as shown in Rensink [20], and consequently the connection to event structures expressed in Prop. 3 is lost immediately. With respect to infinite models, [9] and [17] consider regular expressions but not generally recursive ones.

Worth mentioning is also Boudol and Castellani [5], who present a theory of (nondeterministic) event structures inspired by the work on series-parallel pomsets. They omit all distributivity properties. The class of event structures that can then be denoted is restricted in quite a different sense than ours: certain order structures cannot be generated, a limitation which is inherited directly from the series-parallel pomsets. Infinite behaviour is not considered.

Another approach constitute the *Mazurkiewicz traces*, investigated e.g. by Mazurkiewicz in [15] and by Aaldersberg and Rozenberg in [1]. These in fact form a subclass of deterministic pomsets; however, the operation of *partially commutative concatenation* on traces is really incomparable with any of ours: it combines aspects of sequential composition and join. The relation of Mazurkiewicz traces to event structures is discussed exhaustively in Nielsen, Sassone and Winskel [16]. Infinite traces are considered especially in the form of *regular languages*.

Finally, consider the treatment of deadlock and successful termination. In [5], there is a single constant combining properties of both; in [9], deadlock is right cancellative for sequential composition. See also De Nicola and Labelle [8] for a discussion of this point. Our theory was motivated by the wish to obtain monotonicity (in fact, continuity) of sequential composition – hence deadlock and

successful termination are distinguished – and to remain, in some sense, operational – hence sequential composition is not right cancellative. This is compatible with the interleaving theory of ACP presented e.g. in Baeten and Weijland [3], albeit Axiom 12 is not a basic axiom there (but is inductively derivable).

## References

1. I. J. Aalbersberg and G. Rozenberg. Theory of traces. *Theoretical Comput. Sci.*, 60:1–82, 1988.
2. L. Aceto. Full abstraction for series-parallel pomsets. In S. Abramsky and T. S. E. Maibaum, editors, *TAPSOFT '91, Volume 1*, vol. 493 of *Lecture Notes in Computer Science*, pp. 1–25. Springer-Verlag, 1991.
3. J. C. M. Baeten and W. P. Weijland. *Process Algebra*. Cambridge University Press, 1990.
4. J. A. Bergstra and J. W. Klop. Algebra of communicating processes with abstraction. *Theoretical Comput. Sci.*, 37(1):77–121, 1985.
5. G. Boudol and I. Castellani. A non-interleaving semantics for CCS based on proved transitions. *Fund. Informaticae*, XI(4):433–452, Dec. 1988.
6. G. Boudol and I. Castellani. Permutations of transitions: An event structure semantics for CCS and SCCS. In de Bakker et al. [7], pp. 411–427.
7. 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*, vol. 354 of *Lecture Notes in Computer Science*. Springer-Verlag, 1989.
8. R. De Nicola and A. Labella. A completeness theorem for nondeterministic Kleene algebras. In I. Prívvara, B. Rován, and P. Ružička, editors, *Mathematical Foundations of Computer Science 1994*, vol. 841 of *Lecture Notes in Computer Science*, pp. 536–545. Springer-Verlag, 1994.
9. J. L. Gischer. The equational theory of pomsets. *Theoretical Comput. Sci.*, 61:199–224, 1988.
10. R. van Glabbeek and U. Goltz. Equivalences and refinement. In I. Guessarian, editor, *Semantics of Systems of Concurrent Processes*, vol. 469 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.
11. J. Grabowski. On partial languages. *Fund. Informaticae*, IV(2):427–498, 1981.
12. J. F. Groote. *Process Algebra and Structured Operational Semantics*. PhD thesis, University of Amsterdam, 1991.
13. J. Heering. Partial evaluation and  $\omega$ -completeness of algebraic specifications. *Theoretical Comput. Sci.*, 43:149–167, 1986.
14. A. Lazrek, P. Lescanne, and J.-J. Thiel. Tools for proving inductive equalities, relative completeness, and  $\omega$ -completeness. *Information and Computation*, 84:47–70, 1990.
15. A. Mazurkiewicz. Basic notions of trace theory. In de Bakker et al. [7], pp. 285–363.
16. M. Nielsen, V. Sassone, and G. Winskel. Relationships between models for concurrency. In J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *A Decade of Concurrency*, vol. 803 of *Lecture Notes in Computer Science*, pp. 425–476. Springer-Verlag, 1994.
17. V. R. Pratt. Modeling concurrency with partial orders. *International Journal of Parallel Programming*, 15(1):33–71, 1986.

18. A. Rabinovich and B. A. Trakhtenbrot. Behaviour structure and nets. *Fund. Informaticae*, XI(4):357–404, Dec. 1988.
19. A. Rensink. Posets for configurations! In W. R. Cleaveland, editor, *Concur '92*, vol. 630 of *Lecture Notes in Computer Science*, pp. 269–285. Springer-Verlag, 1992.
20. A. Rensink. Deterministic pomsets. Hildesheimer Informatik-Berichte 94/30, Institut für Informatik, Universität Hildesheim, Nov. 1994.
21. V. Sassone, M. Nielsen, and G. Winskel. Deterministic behavioural models for concurrency. In S. M. Borzyszkowski and S. Sokolowski, editors, *Mathematical Foundations of Computer Science*, vol. 711 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.
22. G. Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Applications and Relationships to Other Models of Concurrency*, vol. 255 of *Lecture Notes in Computer Science*, pp. 325–392. Springer-Verlag, 1987.
23. G. Winskel. An introduction to event structures. In de Bakker et al. [7], pp. 364–397.