HILDESHEIMER INFORMATIK-BERICHTE

ISSN 0941-3014

Arend Rensink

Denotational, Causal, and Operational Determinism in Event Structures

29/95 (September 1995)

UNIVERSITÄT HILDESHEIM

Dieser Bericht ist herausgegeben vom

Institut für Informatik

Postfach 10 13 63
31113 Hildesheim

(This page intentionally left blank)

Denotational, Causal, and Operational Determinism in Event Structures

Arend Rensink
Institut für Informatik, University of Hildesheim*

Abstract

Determinism is a theoretically and practically important concept in labelled transition systems and trees. We study its generalisation to event structures. It turns out that the result depends on what characterising property of tree determinism one sets out to generalise. We present three distinct notions of event structure determinism, and show that none of them shares all the pleasant properties of the one concept for trees.

1 Introduction

Consider the class of edge-labelled *trees*, i.e., labelled transition systems in which the transition relation induces a tree ordering over the states. A *path* in a tree is an alternating sequence of states and labels starting in the initial (smallest) state; a *word* is the corresponding sequence of labels only. A tree is called *deterministic* if from every state there is at most one transition with any given label. The following properties are easily seen to hold:

- A tree is deterministic if and only if each of its words corresponds to a unique path;
- Every tree can be collapsed to a deterministic tree with the same set of words, which is unique up to isomorphism.

In fact, either of these properties can be used to formulate an alternative, equivalent definition of the property of determinism in trees. Under a suitable notion of tree morphism, these properties are combined in the following category theoretic result (which is in fact relatively robust with respect to the choice of morphism):

- Deterministic trees form a reflective subcategory of trees, where the underlying functor is language-preserving.

*Postfach 101363, D–31113 Hildesheim; email: rensink@informatik.uni-hildesheim.de. Research partially supported by the HCM Network "EXPRESS" (Expressiveness of Languages for Concurrency)

Whereas trees have been used very successfully to model the (in general) *nondeterministic* behaviour of systems, to capture at the same time the nondeterministic *and concurrent* aspects of system behaviour, a widely accepted model is that of *event structures*, introduced originally to model Petri net unfoldings (cf. Nielsen, Plotkin and Winskel [5]). That is, trees model the concurrent execution of actions by representing all their linear orderings, and thus do not truly capture the inherent concurrency. The "words" of event structures, on the other hand, are not sequences but *partially ordered multisets* (pomsets) of labels (called *concurrent words* in the sequel); consequently, sequential and concurrent executions are distinguished.

It might be expected that the notion of determinism can be extended easily from trees to event structures; in particular, that its various characterisations discussed above generalise smoothly. As it turns out, however, this is not the case. Rather, one may distinguish *three* kinds of determinism, resulting from the three alternative definitions referred to above; the category theoretical result does not hold fully with respect to any of the resultant properties, although it can be recovered partially for subclasses of event structures. The resultant properties, in order of increasing strictness, are the following:

- For every event structure, there is a *denotationally* deterministic event structure with the same concurrent words, which is unique up to isomorphism. The concurrent words of denotationally deterministic event structures can be arbitrary.

- An event structure is called *causally* deterministic if every concurrent word uniquely corresponds to a run. The concurrent words of causally deterministic event structures are such that distinct events must either have distinct sets of causal predecessors or distinct labels.

- An event structure is called *operationally* deterministic if from every state, at most one event may occur with any given label. The concurrent words of operationally deterministic event structures are actually *auto-sequential*, meaning that equilabelled events are totally ordered; moreover, no distinct concurrent words have a common *linearisation*.

Operational determinism has been studied before in several contexts: Sassone with Nielsen and Winskel studied the categorical relation of operationally deterministic event structures to other behavioural models in the series of papers [11, 12, 6, 10], whereas Vaandrager showed in [13] that such event structures have precisely the expressive power of step sequences. We studied causally deterministic event structures in [8], presenting a complete equational theory for them. To our knowledge, denotational determinism has not been investigated before.

2 Definitions

This section defines a number of more or less standard concepts that are used in the remainder of the paper. Throughout the paper, we assume a universe $\mathbf{E}$ of events, ranged over by d, e, and a universe $\mathbf{A}$ of actions, ranged over by a, b, c.

2.1 Labelled transition systems, trees, paths and words

A *labelled transition system* is a tuple $T = \langle L, S, \to, \iota \rangle$ where L is a set of *labels* (for instance, $L = \mathbf{A}$), S is a set of *states*, $\to \subseteq S \times L \times S$ is a *transition relation* and $\iota \in S$ is the *initial state*. We write $s \xrightarrow{a} s'$ for $(s, a, s') \in \to$. A *path* in T is a sequence $s_0 a_0 s_1 \cdots a_{n-1} s_n$ for some $n \in \mathbb{N}$, where $s_0 = \iota$ and $s_i \xrightarrow{a_i} s_{i+1}$ for all $0 \leq i < n$; the sequence $a_0 \cdots a_{n-1}$ is then called a *word* of T. T is a *tree* if every $s \in S$ is the final state of precisely one path. T is called *deterministic* if $s \xrightarrow{a} s_1$ and $s \xrightarrow{a} s_2$ implies $s_1 = s_2$. Two transition systems T, U are called *isomorphic*, denoted $T \cong U$, if there is a bijection $\psi \colon S_T \to S_U$ such that $s \xrightarrow{a} s'$ iff $\phi(s) \xrightarrow{a} \phi(s')$ for all $s, s' \in S_T$, and $\phi(\iota_T) = \iota_U$.

There is a standard notion of morphism that turns the class of trees into a category $\mathbf{T}$, with as a subcategory the deterministic trees, $\mathbf{T}_{\mathrm{d}}$. On the other hand, one can define a category $\mathbf{L}$ of *languages* (i.e., prefix closed sets of sequences over $\mathbf{A}$). The following properties can be seen to hold with respect to these categories (cf. Nielsen, Sassone and Winskel [6]):

2.1 Proposition. $\mathbf{L}$ is equivalent to $\mathbf{T}_{\mathrm{d}}$.

2.2 Proposition. There is a language-preserving reflection from $\mathbf{T}$ to $\mathbf{T}_{\mathrm{d}}$.

It is the existence of a like situation for event structures that we investigate in this paper. Note that the condition of language preservation in the latter proposition was not taken as essential in [6], and indeed does not generally hold in the framework presented there. It is open for discussion to what degree language preservation is, or should be, an inherent property of determinisation. We return to this issue in the conclusion of the paper.

2.2 Event structures and morphisms

An *event structure* is a tuple $\mathcal{E} = \langle E, <, \mathrm{Coh}, \ell \rangle$ where $E \subseteq \mathbf{E}$ is a set of events, $< \subseteq E \times E$ an irreflexive and transitive *causal ordering* such that $\{d \in E \mid d < e\}$ is finite for all $e \in E$, $\mathrm{Coh} \subseteq \mathbf{2}^E$ is a set of finite sets of events representing a multi-ary *coherence predicate*, such that $F \subseteq G \in \mathrm{Coh}$ implies $F \in \mathrm{Coh}$ and $d < e \in F \in \mathrm{Coh}$ implies $F \cup \{d\} \in \mathrm{Coh}$, and $\ell \colon E \to \mathbf{A}$ is a *labelling function.*[1] We denote $d \# e$ for $\{d, e\} \notin \mathrm{Coh}$ and $\#^=$ for the reflexive closure of $\#$; $\leq$ will denote the reflexive closure of $<$. Finally, d and e are called *concurrent* if they are neither causally ordered nor conflicting. We use the following notation for the predecessors, resp. the proper predecessors of a set $F \subseteq E$:

$$\begin{array}{rcl} \lceil F \rceil_{\mathcal{E}} & := & \{d \in E \mid d \leq e\} \\ \llbracket F \rrbracket_{\mathcal{E}} & := & \lceil F \rceil_{\mathcal{E}} - F \ . \end{array}$$

We use $\lceil e \rceil_{\mathcal{E}}$ and $\llbracket e \rrbracket_{\mathcal{E}}$ to abbreviate $\lceil \{e\} \rceil_{\mathcal{E}}$ and $\llbracket \{e\} \rrbracket_{\mathcal{E}}$, respectively. We use $E_{\mathcal{E}}$, $<_{\mathcal{E}}$, $\mathrm{Coh}_{\mathcal{E}}$ and $\ell_{\mathcal{E}}$ to denote the components of an event structure $\mathcal{E}$, but omit indices when they are clear from the context. An function ϕ is an *isomorphism* from $\mathcal{E}$ to $\mathcal{F}$, denoted $\phi \colon \mathcal{E} \cong \mathcal{F}$, if ϕ is a bijection from $E_{\mathcal{E}}$ to $E_{\mathcal{F}}$ such that $d <_{\mathcal{E}} e \Leftrightarrow \phi(d) <_{\mathcal{F}} \phi(e)$, $F \in \mathrm{Coh}_{\mathcal{E}} \Leftrightarrow \phi(F) \in \mathrm{Coh}_{\mathcal{F}}$ and $\ell_{\mathcal{E}}(e) = \ell_{\mathcal{F}}(\phi(e))$ for all $d, e \in E_{\mathcal{E}}$ and $F \subseteq E_{\mathcal{E}}$. $\mathcal{E}$ and $\mathcal{F}$ are then called *isomorphic*,

[1] These are the event structures with general conflict (see Winskel [14])

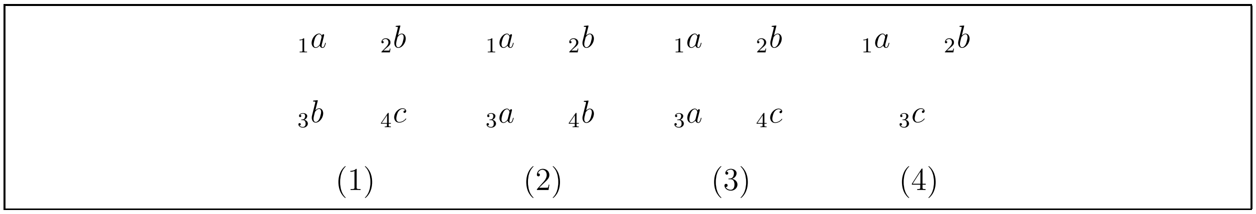


Figure 1: Some event structures. In (4), the set $\{1, 2, 3\}$ *is conflicting, but all its proper subsets are coherent.*

denoted $\mathcal{E} \cong \mathcal{F}$. The *restriction* of an event structure $\mathcal{E}$ to a set of events $F \subseteq E$ is defined by

$$\mathcal{E} \upharpoonright F \;:=\; \langle F, < \cap\, (F \times F), \mathrm{Coh} \cap \mathbf{2}^F, \ell \upharpoonright F \rangle \;\; .$$

See Figure 1 for some examples of event structures, where the arrows represent causality and the dotted lines conflict. The notation $_1a$ etc. denotes the event 1 labelled with the action a (where we assume $\mathbb{N} \subseteq \mathbf{E}$). We omit events when the structure is to be interpreted modulo isomorphism.

In order to state our results in a category theoretic setting, we define a notion of *event structure morphism.* In this, we deviate from the standard notion of Winskel [14] and Nielsen, Sassone and Winskel [6], because we want to highlight the issue of determinism in isolation, rather than regarding it in combination with concurrency. To be precise, our morphisms are more restricted than the standard ones, in that they are allowed to manipulate conflict but not causality. At the end of the paper (Section 6) we will discuss how the situation changes when the standard notion of morphism is used instead.

An event structure morphism from $\mathcal{E}$ to $\mathcal{F}$ is a pair (λ, η) (notation: $(\lambda, \eta) \colon \mathcal{E} \to \mathcal{F}$) where λ is a partial function from $\mathbf{A}$ to $\mathbf{A}$ and η a partial function from $E_\mathcal{E}$ to $E_\mathcal{F}$, such that for all $e \in E_\mathcal{E}$, $\eta(e)$ is defined iff $\lambda(\ell_\mathcal{E}(e))$ is defined, in which case $\ell_\mathcal{F}(\eta(e)) = \lambda(\ell_\mathcal{E}(e))$; moreover, η preserves and reflects sets of predecessors (i.e., $\eta(\llbracket e \rrbracket_\mathcal{E}) = \llbracket \eta(e) \rrbracket_\mathcal{F}$ for all $e \in \mathrm{dom}\,\eta$),[2] is non-injective only on conflicting events (i.e., $\eta(d) = \eta(e)$ implies $d \,\#_\mathcal{E}^{=}\, e$ for all $d, e \in \mathrm{dom}\,\eta$), and preserves coherency (i.e., $F \in \mathrm{Coh}_\mathcal{E}$ implies $f(F) \in \mathrm{Coh}_\mathcal{F}$ for all $F \subseteq E_\mathcal{E}$). Event structures and their morphisms, with identity morphisms $(id_\mathbf{A}, id_\mathcal{E})$ for all $\mathcal{E}$ and pairwise composition of morphisms, trivially give rise to a category **ES**. Note that the resulting notion of *iso*morphism coincides with the one presented explicitly above; that is, $(\lambda, \eta) \colon \mathcal{E} \to \mathcal{F}$ is an isomorphism iff $\lambda = id_\mathbf{A}$ and $\eta \colon \mathcal{E} \cong \mathcal{F}$.

2.3 Partially ordered sets and multisets

A *labelled partially ordered set* (lposet) is a finite event structure without conflict; i.e., a triple $p = \langle E, <, \ell \rangle$ (where the conflicting sets are omitted altogether). The notion of isomorphism is inherited from event structures. An lposet p is a *prefix* of an lposet q, denoted $p \preceq q$, if $E_p \subseteq E_q$ is left-closed according to $<_q$ ($d <_q e \in E_p$ implies $d \in E_p$) and $p = q \upharpoonright E_p$. p *augments* or *smoothens* q, denoted $p \sqsubseteq q$, if they have the same sets of events ($E_p = E_q$), p has more ordering ($<_p \supseteq <_q$) and the labelling functions coincide ($\ell_p = \ell_q$). An lposet p is

[2] In contrast, standard morphisms satisfy $\eta(\llbracket e \rrbracket) \supseteq \llbracket \eta(e) \rrbracket$.

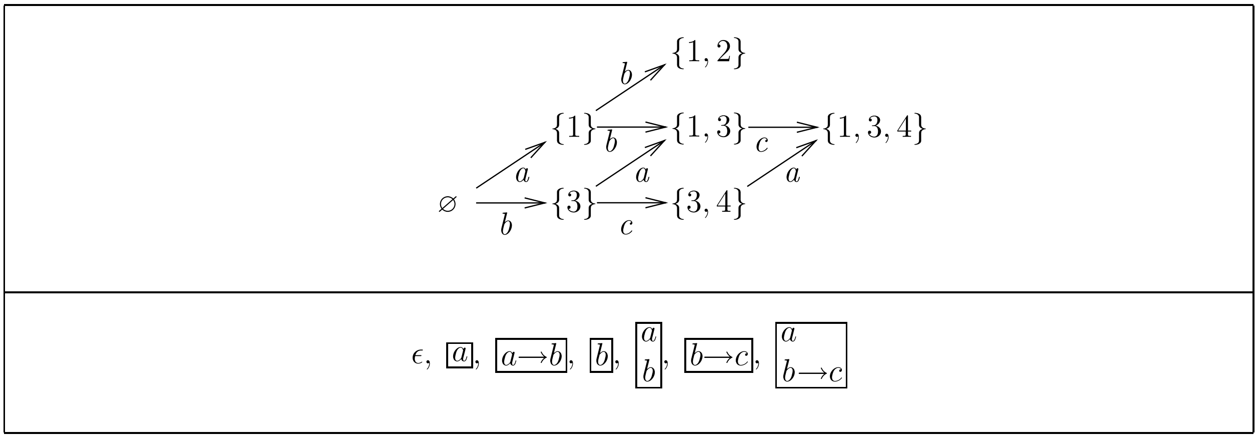


Figure 2: Transitions and concurrent language of structure (1) of Fig. 1

called *topped* if it has a greatest (top) element (i.e., $e \in E_p$ such that $d \leq e$ for all $d \in E_p$); we write $\top_p$ for the greatest element (e.g., $\top_p = e$ in the above case).

A *partially ordered multiset* (pomset) is an isomorphism class of lposets $[p]_{\cong} = \{q \mid q \cong p\}$; we usually denote $[p]_{\cong}$ by $[p]$. The concepts of lposet prefix and augmentation are lifted to pomsets: $[p] \preceq [q]$ if $p \cong p' \preceq q$ and $[p] \sqsubseteq [q]$ iff $p \cong p' \sqsubseteq q$ for some lposet p'.

A *concurrent language* $\mathcal{L}$ is a prefix closed sets of pomsets (i.e., such that $[p] \preceq [q] \in \mathcal{L}$ implies $[p] \in \mathcal{L}$). Concurrent languages give rise to a category **CL** where morphisms are partial functions λ from $\mathbf{A}$ to $\mathbf{A}$, which are extended to functions $\hat{\lambda}$ from pomsets to pomsets by defining $\hat{\lambda}([p]) = [q]$ with

$$\begin{array}{rcl} E_q & = & \{e \in E_p \mid \lambda \text{ defined on } \ell_p(e)\} \\ <_q & = & <_p \cap (E_q \times E_q) \\ \ell_q & = & \lambda \circ (\ell_p \upharpoonright E_q) \ . \end{array}$$

(Note that $[q]$ is well-defined modulo the choice of representative p.) Then λ is a morphism from $\mathcal{L}$ to $\mathcal{M}$ (notation: $\lambda\colon \mathcal{L} \to \mathcal{M}$) iff $\hat{\lambda}(\mathcal{L}) \subseteq \mathcal{M}$.

2.4 Configurations, event transitions and concurrent languages

A *configuration* of an event structure $\mathcal{E}$ is a coherent (and therefore finite) set $F \in \mathrm{Coh}$ which is left-closed according to $<$ ($d < e \in F$ implies $d \in F$). The configurations of $\mathcal{E}$ are collected in $\mathcal{C}(\mathcal{E})$. $\mathcal{E}$ thus naturally gives rise to the tree $es.t(\mathcal{E}) = \langle \mathbf{A}, \mathcal{C}(\mathcal{E}), \rightarrow, \varnothing \rangle$ where for all $F, G \in \mathcal{C}(\mathcal{E})$, $F \xrightarrow{a} G$ iff $G = F \cup \{e\}$ for some $e \notin F$ such that $\ell_{\mathcal{E}}(e) = a$. It is not difficult to prove that $\mathcal{E} \cong \mathcal{F}$ iff $es.t(\mathcal{E}) \cong es.t(\mathcal{F})$. A *concurrent word* of $\mathcal{E}$ is a pomset $[p]$ such that E_p is a configuration of $\mathcal{E}$ and $p = \mathcal{E} \upharpoonright E_p$. The concurrent words of $\mathcal{E}$ are collected in $es.cl(\mathcal{E})$. It is clear that $es.cl(\mathcal{E})$ is prefix-closed, hence a concurrent language. Figure 2 shows an example.

An event structure morphism $(\lambda, \eta)\colon \mathcal{E} \to \mathcal{F}$ implies concurrent language inclusion after λ-renaming. This is due to the fact that event structure morphisms, as we have defined them, completely preserve the causal structure of configurations. More precisely, if (λ, η) is a morphism from $\mathcal{E}$ to $\mathcal{F}$ then $\hat{\lambda}(es.cl(\mathcal{E})) \subseteq es.cl(\mathcal{F})$. Hence we have the following:

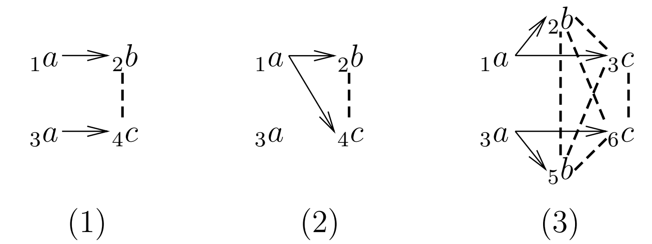


Figure 3: Event structures with the same concurrent languages

2.3 Proposition. The mapping $es.cl\colon \mathbf{ES} \to \mathbf{CL}$ gives rise to a functor, with arrow part $(\lambda, \eta) \mapsto \lambda$.

3 Denotational determinism

We come to the first of our notions of event structure determinism. It is based on the idea that denotationally, a deterministic model is completely determined (up to isomorphism) by its concurrent language. We will show that for any event structure there is a denotationally deterministic event structure, unique up to isomorphism, with the same concurrent language. However, due to the possible presence of equilabelled events which are *causally indistinguishable*, in the sense of having the same set of proper predecessors, the construction of the denotationally deterministic event structure is not always straightforward. Consider the event structures in Figure 3. They have the same concurrent language, namely $\left\{\epsilon, \boxed{a}, \boxed{a\to b}, \boxed{a\to c}, \boxed{\begin{matrix}a\\a\end{matrix}}, \boxed{\begin{matrix}a\to b\\a\end{matrix}}, \boxed{\begin{matrix}a\\a\to c\end{matrix}}\right\}$; however, their choice structure is different. Neither (1) nor (2) is in any way deterministic, since in either case, when an a occurs, the choice of event (either 1 or 3) affects the possible continuations. Structure (3) does not share this characteristic, and indeed it satisfies the criteria we will formulate below for denotational determinism. In fact, structure (3) *determinises* the other two (where determinisation is the operation of constructing a deterministic event structure with the same concurrent language).

By the same token, even an event structure that contains no conflict may be nondeterministic, and to determinise it, conflict may have to be introduced; see structure (2) in Figure 4. In contrast, if equilabelled events have different causal predecessors, such as 1 and 7 resp. 4 and 7 of structure (4) in Figure 4, or have isomorphic continuations, such as 1 and 4 in the same structure, then this does not violate denotational determinism.

Since the property of having both equal labels and equal sets of predecessors plays an important role in the following, we introduce a special relation between events, called *causal indistinguishability*.

$$d \sim e \;:\Leftrightarrow\; \ell(d) = \ell(e) \wedge [\![d]\!] = [\![e]\!]$$

Basically, an event structure will be denotationally deterministic if all causally indistinguishable events are *non-conflicting*, and moreover *isomorphic* in the sense that there is an auto-isomorphism of the entire event structure that maps them to each other. The operation of determinising a given event structure therefore consists of manipulating its causally indistinguishable events: if they are conflicting then they are merged, otherwise a copy of

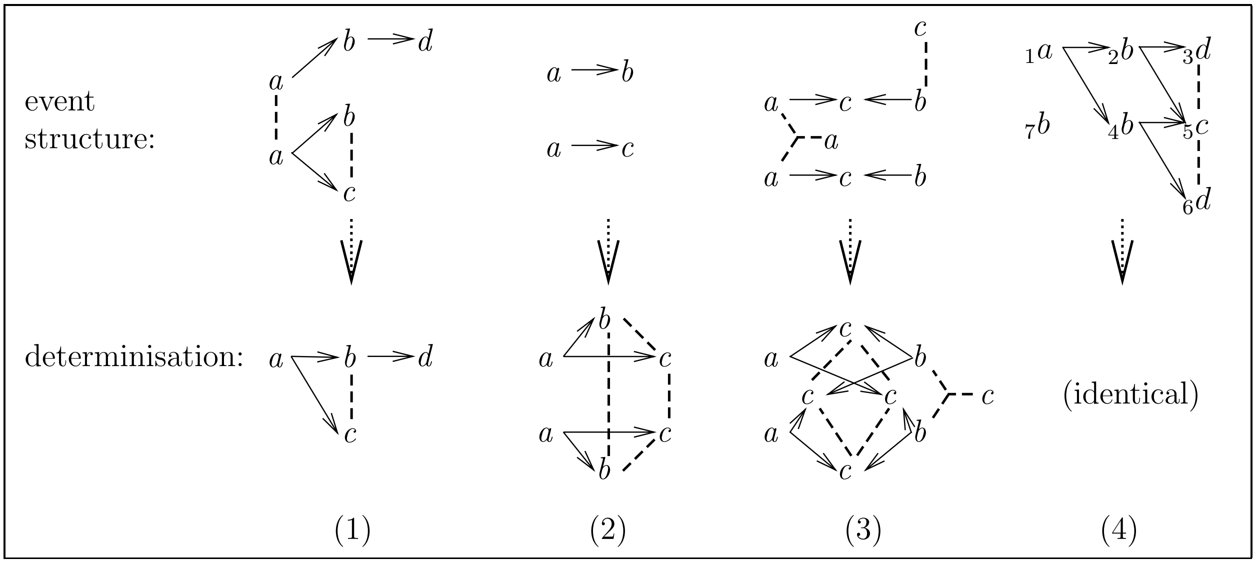


Figure 4: Event structures and their denotational determinisations

the "causal context" of each is added with respect to the other, so that they end up being isomorphic. This is illustrated by structures (1)–(3) of Figure 4.

We will see that in general, the connection (one-to-one correspondence up to isomorphism) between denotationally deterministic event structures and concurrent languages cannot be made very strong; in particular, it does *not* give rise to a categorical equivalence. In fact, it is difficult to construct even a functor from the denotationally deterministic event structures to concurrent languages. By the same token, the subcategory of denotationally deterministic event structures does *not* occupy any special position within **ES**.

3.1 Definition (denotational determinism). An event structure $\mathcal{E}$ is *denotationally deterministic* if the following conditions hold:

- for all $e \in E$, if $F \subseteq_{\mathrm{fin}} [e]_\sim$ then $F \in \mathrm{Coh}$.
- for all pairwise concurrent $F \in \mathrm{Coh}_{\mathcal{E}}$ with $d, e \in F$ such that $d \sim e$, there is an auto-isomorphism $\phi: \mathcal{E} \cong \mathcal{E}$ such that $\phi(e) = d$ and ϕ is the identity on $F - \{d, e\}$.

(The first condition cannot be simplified to $[e]_\sim \in \mathrm{Coh}$, because $[e]_\sim$ may be an infinite set.) The class of denotationally deterministic event structures will be denoted $\mathbf{ES}_{\mathrm{dd}}$. A necessary condition for denotational determinism is that every isomorphism between two configurations of an event structure (which therefore give rise to identical concurrent words) can be extended to an isomorphism of the entire structure.

3.2 Proposition. If $\mathcal{E}$ is denotationally deterministic then for all $F, G \in \mathcal{C}(\mathcal{E})$ such that $\phi: \mathcal{E} \restriction F \cong \mathcal{E} \restriction G$ there is a $\psi: \mathcal{E} \cong \mathcal{E}$ such that $\psi \restriction F = \phi$.

Proof. By induction on $|F|$. The theorem is trivially true for $F = \varnothing$, since then ϕ is the empty function and $\psi = id$ will do. Now assume the lemma to have been proved for all $|F| \leq n$, and assume $|F| = n + 1$. Let $F' \subseteq F$ be the set of $<$-maximal elements (note that F' is then pairwise concurrent) and let $e \in F'$ be arbitrary and $F'' = F - \{e\}$; then clearly $|F''| = n$ and $(\phi \restriction F'') \colon \mathcal{E} \restriction F'' \cong \mathcal{E} \restriction G''$ where $G'' = \phi(F'')$. Hence by the induction hypothesis, there is a $\psi \colon \mathcal{E} \cong \mathcal{E}$ such that $\psi \restriction F'' = \phi \restriction F''$; it follows that $\phi(e) \sim \psi(e)$. But then, according to Definition 3.1, there is a $\psi' \colon \mathcal{E} \cong \mathcal{E}$ such that $\psi'(\psi(e)) = \phi(e)$ and $\psi'(d) = d$ for all $d \in F' - \{e\}$. If we replace the part $\psi' \restriction F''$ by $id_{F''}$, the resulting function is still an isomorphism; hence $\psi'' = (\psi' \restriction (E - F'') \cup id_{F''}) \circ \psi$ is the required auto-isomorphism $\psi'' \colon \mathcal{E} \cong \mathcal{E}$ such that $\psi'' \restriction F = \phi$. □

It follows immediately that every configuration can be extended to every concurrent word of which it yields a prefix. This is expressed by the following lemma.

3.3 Lemma. Let $\mathcal{E} \in \mathbf{ES}_{\mathrm{dd}}$. If $[\mathcal{E} \restriction F] \preceq [p] \in es.cl(\mathcal{E})$ for some $F \in \mathcal{C}(\mathcal{E})$ then there is a $G \in \mathcal{C}(\mathcal{E})$ such that $F \subseteq G$ and $\mathcal{E} \restriction G \cong p$.

Proof. $[\mathcal{E} \restriction F] \preceq [p] \in es.cl(\mathcal{E})$ implies $G', G'' \in \mathcal{C}(\mathcal{F})$ such that $G' \subseteq G''$, $\phi \colon \mathcal{E} \restriction F \cong \mathcal{E} \restriction G'$ and $\mathcal{E} \restriction G'' \cong p$. Hence (by Proposition 3.2) there is a $\psi \colon \mathcal{E} \cong \mathcal{E}$ such that $\psi \restriction F = \phi$; hence $G = \psi(G'')$ satisfies the conditions of the lemma. □

3.1 Denotationally deterministic event structures

One of the crucial consequences of denotational determinism is that there exists a denotationally deterministic event structure for every concurrent language. This is proved in the following theorem.

3.4 Theorem. $\mathcal{L} \in \mathbf{CL}$ iff $\mathcal{L} = es.cl(\mathcal{E})$ for some $\mathcal{E} \in \mathbf{ES}_{\mathrm{dd}}$.

Proof. The "if" part is trivial. For the "only if", we give the construction of $\mathcal{E}$ through a series of approximants $\mathcal{E}_i = \langle E_i, <_i, \mathrm{Coh}_i, \ell_i \rangle$ for $i \in \mathbb{N}$, by induction on the *depth* of events (where the depth of $e \in E_{\mathcal{E}}$ equals the length of the longest chain $e_0 <_{\mathcal{E}} e_1 <_{\mathcal{E}} \cdots <_{\mathcal{E}} e$; hence initial events have depth 1). $\mathcal{E}_0$ is the empty structure; the construction of $\mathcal{E}_{i+1}$ from $\mathcal{E}_i$ and $\mathcal{L}$ is as follows.

Events. For all *topped* $[p] \in \mathcal{L}$ where $\top_p$ has depth $i+1$, let n be the least upper bound of the number of distinct, p-isomorphic prefixes of any element of $\mathcal{L}$, i.e.,

$$n = \bigsqcup_{[q] \in \mathcal{L}} |\{e \in E_q \mid q \restriction \lceil e \rceil \cong p\}| \ .$$

Note that $n \in \{1, \ldots, \infty\}$. Now for all $G \in \mathcal{C}(\mathcal{E}_i)$ such that $\mathcal{E}_i \restriction G \cong p \restriction \llbracket e \rrbracket$ and all $m < n$ let $(G, \ell_p(\top_p), m)$ be a new event of $\mathcal{E}_{i+1}$.

Orderings. For all new events (G, a, m), let $e <_{i+1} (G, a, m)$ iff $e \in G$.

Labels. For all new $(G, a, m) \in E_{i+1}$, let $\ell_{i+1}(G, a, m) = a$.

Coherence. For all $F \subseteq E_{i+1}$, let $p = \langle \lceil F \rceil, <_{i+1} \restriction (\lceil F \rceil \times \lceil F \rceil), \ell_{i+1} \restriction \lceil F \rceil \rangle$ be the smallest initial segment of $\mathcal{E}_{i+1}$ containing F; let $F \in \mathrm{Coh}_{i+1}$ iff $[p] \in \mathcal{L}$.

It follows that $\mathcal{E}_i = \mathcal{E}_{i+1} \upharpoonright E_i$ for all $i \in \mathbb{N}$; we define $\mathcal{E} = \bigcup_{i\in\mathbb{N}} \mathcal{E}_i$ by component-wise union of the approximants. Clearly, $\mathcal{E}$ is an event structure; moreover, it is not difficult to prove that $es.cl(\mathcal{E}_i)$ contains all elements of $\mathcal{L}$ up to depth i; hence $es.cl(\mathcal{E}) = \mathcal{L}$. Finally, to see that $\mathcal{E}$ is deterministic, we check the conditions of Definition 3.1:

- If $F \subseteq_{\rm fin} [e]_\sim$ for some $e \in E_\mathcal{E}$, then it follows that $e = (G, a, m)$ and $[e]_\sim = \{(G, a, k) \mid k < n\}$ for some $m < n \in \{1, \ldots, \infty\}$. Hence there is a $p \in \mathcal{L}$ such that $|\{d \in E_p \mid p \upharpoonright \lceil d \rceil \cong \mathcal{E}_i \upharpoonright \lceil e \rceil\}| \geq |F|$, where i is the depth of e, which implies $[\mathcal{E}_i \upharpoonright \lceil F \rceil] \preceq p$ and hence $F \in \mathrm{Coh}_i$. Since $\mathcal{E}_i = \mathcal{E} \upharpoonright E_i$, it follows that $F \in \mathrm{Coh}_\mathcal{E}$.
- If $F \in \mathrm{Coh}_\mathcal{E}$ is a pairwise concurrent set of events, then by construction $e = (G_e, a_e, k_e)$ such that $d \notin G_e$ for all $d, e \in F$. If $d \sim e$ for $d, e \in F$ then $G_d = G_e$ and $a_d = a_e$. Now define $\phi: E_\mathcal{E} \to E_\mathcal{E}$ inductively by $\phi(G, a, k) = (\phi(G), a, i)$ where $i = k_d$ if $(G, a, i) = e$, $i = k_e$ if $(G, a, i) = d$ and $i = k$ otherwise. Note that $\phi(G_{e'}) = G_{e'}$ for all $e' \in F$, and hence $\phi(e') = e'$ for $e' \in F - \{d, e\}$ whereas $\phi(e) = d$ and $\phi(d) = e$. It is not difficult to check $\phi: \mathcal{E} \cong \mathcal{E}$. □

The resulting mapping from concurrent languages to (deterministic) event structures will be denoted $cl.es: \mathbf{CL} \to \mathbf{ES}_{\rm dd}$. It follows that, in a sense, $\mathbf{ES}_{\rm dd}$ is large enough (namely to capture all concurrent languages). However, this is perforce also true of any *larger* class of event structure; for instance of the entire **ES**. The following theorem, however, expresses the *dual* fact that $\mathbf{ES}_{\rm dd}$ is also, in a sense, *small* enough: its elements are completely determined (up to isomorphism) by their concurrent language.

3.5 Theorem. For all $\mathcal{E}, \mathcal{F} \in \mathbf{ES}_{\rm dd}$, $es.cl(\mathcal{E}) = es.cl(\mathcal{F})$ iff $\mathcal{E} \cong \mathcal{F}$.

Proof. We prove that $\mathcal{E} \cong es.cl(cl.es(\mathcal{E}))$ for all $\mathcal{E} \in \mathbf{ES}_{\rm dd}$; in combination with Theorem 3.4 this proves the theorem.

Let $\mathcal{F} = es.cl(cl.es(\mathcal{E}))$. For every $F \in \mathcal{E}/{\sim}$, let $n_e: F \to \{i \in \mathbb{N} \mid i < |F|\}$ be bijective such that for all $d \in F$, $i < n_e(d)$ implies $i = n_e(d')$ for some $d' \in F$. Now define $\phi: E_\mathcal{E} \to E_\mathcal{F}$ inductively by

$$\phi: e \mapsto (\phi(\llbracket e \rrbracket_\mathcal{E}), \ell_\mathcal{E}(e), n_{[e]_\sim}(e)) \ .$$

We show $\phi: \mathcal{E} \cong \mathcal{F}$. First, it can be shown inductively on the definition of ϕ that $\phi(e) \in E_\mathcal{F}$ for all $e \in E_\mathcal{E}$. Especially, note that since $F \in \mathrm{Coh}_\mathcal{E}$ for all $F \subseteq_{\rm fin} [e]_\sim$ (Definition 3.1), the n computed in the proof of Theorem 3.4 equals $|F|$. Then, it is easy to show (again inductively on the definition of ϕ) that ϕ is injective. Moreover, for all $e \in E_\mathcal{E}$, $\phi(\llbracket e \rrbracket_\mathcal{E}) = \llbracket \phi(e) \rrbracket_\mathcal{F}$ by the definition of ϕ and $<_\mathcal{F}$; hence $d <_\mathcal{E} e \Leftrightarrow \phi(d) <_\mathcal{F} \phi(e)$. Furthermore, if $F \in \mathrm{Coh}_\mathcal{E}$ then $[\mathcal{F} \upharpoonright \lceil \phi(F) \rceil] = [\mathcal{E} \upharpoonright \lceil F \rceil] \in es.cl(\mathcal{F})$; hence $\phi(F) \in \mathrm{Coh}_\mathcal{F}$. Finally, $\ell_\mathcal{F}(\phi(e)) = \ell_\mathcal{E}(e)$ by definition of ϕ and $\ell_\mathcal{F}$. It remains to be proved that $\phi(F) \in \mathrm{Coh}_\mathcal{F}$ implies $F \in \mathrm{Coh}_\mathcal{E}$ and that ϕ is surjective. We start with the latter, which we prove by induction on the depth of events in $\mathcal{F}$.

- Depth 0: no events have this depth, so ϕ is trivially surjective.
- Assume that the surjectivity of ϕ has been proved for events up to depth i, and let $(G, a, m) \in E_\mathcal{F}$ have depth $i + 1$. It follows that $\llbracket (G, a, m) \rrbracket = G \in \mathcal{C}(\mathcal{F})$ consists

of events of depth at most i; let $F \subseteq E_{\mathcal{E}}$ be the unique set existing by the induction hypothesis such that $\phi(F) = G$; then $F \in \mathcal{C}(\mathcal{E})$ and $\mathcal{E} \upharpoonright F = \mathcal{F} \upharpoonright G$.

Let $G' = \lceil\{(G, a, i) \mid 1 \leq i \leq m\}\rceil_{\mathcal{F}}$. By construction of $\mathcal{F}$, $[p] = [\mathcal{F} \upharpoonright G] \in \mathit{es.cl}(\mathcal{E})$; since also $[\mathcal{E} \upharpoonright F] \preceq [p]$ it follows by Lemma 3.3 that there is a $F' \in \mathcal{C}(\mathcal{F})$ such that $F \subseteq F'$ and $\mathcal{E} \upharpoonright F' \cong p$. In fact, it follows that $|F' - F| = m$, $\llbracket e \rrbracket = F$ and $d \sim e$ for all $d, e \in F' - F$. Let $F'' \in E_{\mathcal{E}}/\!\sim$ be such that $F' \subseteq F''$; then $m = n_{F''}(e)$ for some (unique) $e \in F''$, which then satisfies $\phi(e) = (\phi(\llbracket e \rrbracket_{\mathcal{E}}), \ell_{\mathcal{E}}(e), n_{F''}(e)) = (\phi(F), a, m) = (G, a, m)$.

Finally, $\phi(F) \in \mathrm{Coh}_F \Rightarrow F \in \mathrm{Coh}_{\mathcal{E}}$ is proved by induction on the maximum depth of events in F. Without loss of generality, F can be assumed to be pairwise unordered (otherwise take the $<_{\mathcal{E}}$-maximal elements).

- Maximum depth 0: $F = \varnothing$, trivial.
- Assume that the implication has been proved for every F with maximum depth i, and assume F has maximum depth $i+1$. Let $F' \subseteq F$ contain all elements with depth $i+1$. Then $G = \lceil F \rceil_{\mathcal{E}} - F'$ has maximum depth i, and $\phi(G) = \lceil \phi(F) \rceil_{\mathcal{F}} - \phi(F') \in \mathrm{Coh}_{\mathcal{F}}$, hence (by the induction hypothesis) $G \in \mathrm{Coh}_{\mathcal{E}}$, implying also $G \in \mathcal{C}(\mathcal{E})$. Moreover, $[\mathcal{E} \upharpoonright G] \preceq [\mathcal{F} \upharpoonright \lceil \phi(F) \rceil] \in \mathit{es.cl}(\mathcal{E})$; let $G' \in \mathcal{C}(\mathcal{E})$ be such that $G \subseteq G'$ and $\psi \mathcal{E} \upharpoonright G' \cong \mathcal{F} \upharpoonright \lceil \phi(F) \rceil$ (which exists by Lemma 3.3). It follows that $\psi(G' - G) = F'$ and $\psi(e) \sim e$ for all $e \in G' - G$. By induction on $|F|$ and application of the second property of Definition 3.1 it follows that there is an isomorphism $\psi' \colon \mathcal{E} \cong \mathcal{E}$ mapping F into G', which implies $F \in \mathrm{Coh}_{\mathcal{E}}$. □

It follows that every event structure can be *determinised* uniquely, in the sense that there exists an event structure, unique up to isomorphism, with the same concurrent language. The determinisation mapping will be denoted $\mathit{es.des} = \mathit{cl.es} \circ \mathit{es.cl}$.

3.6 Corollary. For every $\mathcal{E} \in \mathbf{ES}$, $\mathit{es.des}(\mathcal{E}) \in \mathbf{ES}_{\mathrm{dd}}$ is unique up to isomorphism such that $\mathit{es.cl}(\mathcal{E}) = \mathit{es.cl}(\mathit{es.des}(\mathcal{E}))$.

3.2 Denotational determinism categorically

So far for the positive results about denotational determinism. We now show that the role of the objects of $\mathbf{ES}_{\mathrm{dd}}$ as representatives of the concurrent languages of arbitrary event structures is rather superficial, in the sense that it cannot be generalised to a category theoretical setting. In particular, the one-to-one correspondence between concurrent languages and deterministic event structures modulo isomorphism does *not* give rise to an equivalence of categories.

In fact, the first surprise is that the mapping $\mathit{cl.es} \colon \mathbf{CL} \to \mathbf{ES}_{\mathrm{dd}}$ cannot even be extended naturally to a functor: there are morphisms $\lambda \colon \mathcal{L} \to \mathcal{M}$ for which no $(\lambda, \eta) \colon \mathit{cl.es}(\mathcal{L}) \to \mathit{cl.es}(\mathcal{M})$ exists. This is due to the fact that the relabelling part of a morphism may map different topped pomsets onto the same one, which on the level of event structures gives rise to confusion about which indistinguishable events of the source are to be mapped onto which events of the target. Consider for instance $\lambda = (a \mapsto d, b \mapsto d, c \mapsto d)$, which is a morphism

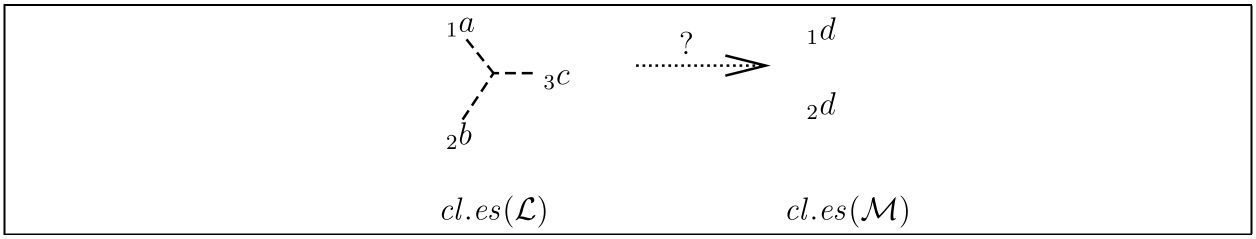


Figure 5: There is no morphism $cl.es(\mathcal{L}) \rightarrow cl.es(\mathcal{M})$

from $\mathcal{L} = \left\{\epsilon, \boxed{a}, \boxed{b}, \boxed{c}, \boxed{\begin{matrix}a\\b\end{matrix}}, \boxed{\begin{matrix}a\\c\end{matrix}}, \boxed{\begin{matrix}b\\c\end{matrix}}\right\}$ to $\mathcal{M} = \left\{\epsilon, \boxed{d}, \boxed{\begin{matrix}d\\d\end{matrix}}\right\}$. The corresponding deterministic event structures are given in Figure 5, where $cl.es(\mathcal{L})$ is such that every pair of events is coherent, but the three together are conflicting. The coherence of every pair of events implies that no pair of events may be mapped to the same event of $cl.es(\mathcal{M})$, and hence no morphism exists.

If we disallow relabelling in morphisms (that is, $\lambda = id_{\mathbf{A}}$ always, and omitted in the remainder of this section; hence the event part is totally defined), a functor can be defined on the basis of $cl.es$, but even so $cl.es$ and $es.cl$ do not form an equivalence of categories. One way of explaining this is that deterministic event structures contain nonessential information (due to the copying of events in the context of causally indistinguishable events) that is accessible by morphisms; there are consequently too many morphisms, which on the level of concurrent languages collapse or disappear. Consider: even a single pair of causally indistinguishable events in a denotationally deterministic event structure gives rise to nontrivial auto-isomorphisms; see for instance structure (1) of Figure 6. On the other hand, as remarked in Section 2.3, there are no nontrivial auto-isomorphisms over families of pomsets. Hence, if $\mathcal{L} = \left\{\epsilon, \boxed{a}, \boxed{a \rightarrow b}, \boxed{\begin{matrix}a\\a\end{matrix}}, \boxed{\begin{matrix}a\\a \rightarrow b\end{matrix}}\right\}$ and $\mathcal{E} = cl.es(\mathcal{L})$ is as in Figure 6 then there are more morphisms from $\mathcal{E}$ to $cl.es(\mathcal{L})$ (namely two) than from $es.cl(\mathcal{E})$ to $\mathcal{L}$ (just one), and more morphisms from $cl.es(\mathcal{L})$ to $\mathcal{E}$ than from $\mathcal{L}$ to $es.cl(\mathcal{L})$. Hence $cl.es$ is neither left nor right adjoint to $es.cl$; nor can this be repaired by varying the arrow part of $cl.es$.

For much the same reason, there does not exist a *reflection* from $\mathbf{ES}$ to $\mathbf{ES}_{\mathrm{dd}}$ —the existence of which is taken in [11, 6] as the *sine qua non* of a proper notion of determinisation. Consider: a reflection would require the existence, for every event structure $\mathcal{E}$, of a morphism

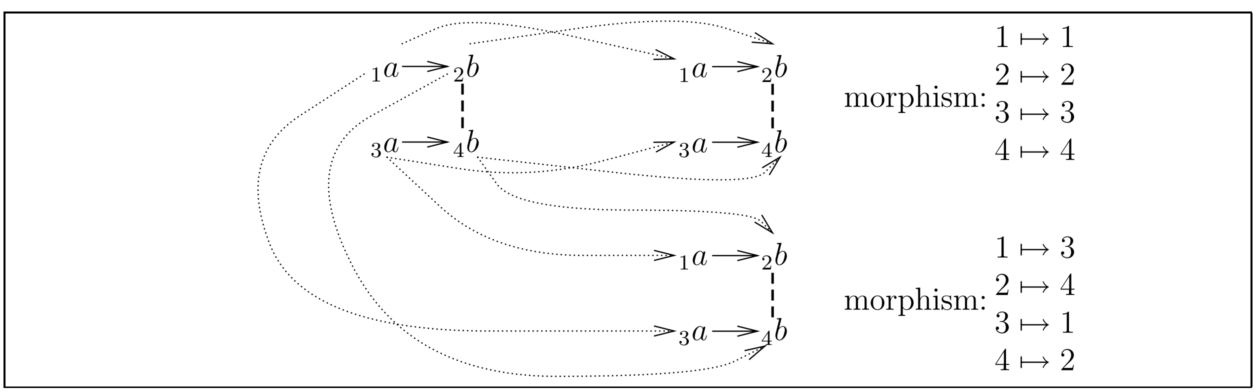


Figure 6: Distinct auto-isomorphisms due to causal indistinguishability

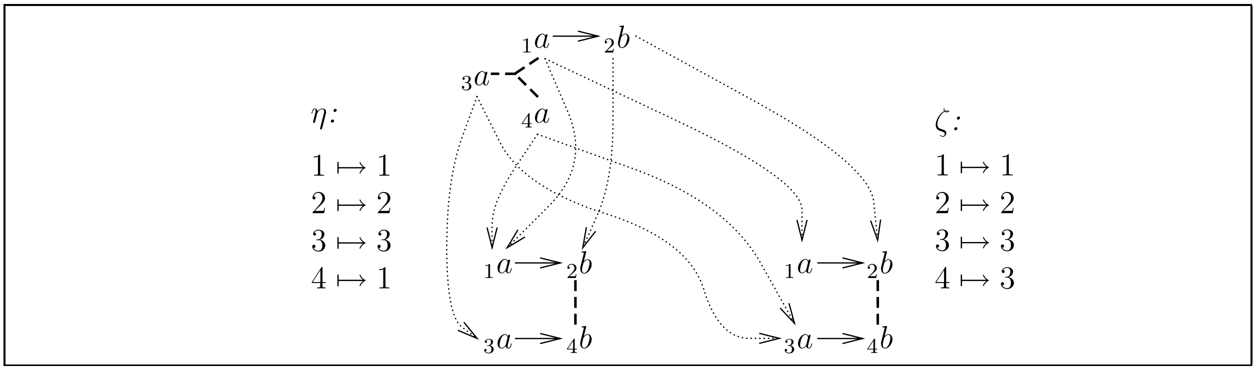


Figure 7: ζ does not factor through the (candidate) determinising morphism η

η mapping it to its determinisation, in such a way that any other morphism ζ from $\mathcal{E}$ to a deterministic event structure uniquely factors through η. Although the determinisation exists as an object (Corollary 3.6) and a determinising morphism η can generally be found, the necessary copying during determinisation destroys the factorisation property. See for instance Figure 7. The figure shows one of the six (symmetric) candidates for η for the given $\mathcal{E}$; the morphism ζ, also shown, does not factor through η, since η already collapses events 1 and 4, whereas ζ is injective on 1 and 4. Note that any choice of η is vulnerable to the construction of such a counterexample.

4 Causal determinism

We move to the second notion of determinism over event structures, called *causal determinism*. It is stricter than denotational determinism, i.e., rules out certain denotationally deterministic models; the property that every event structure can be determinised is therefore automatically lost. However, causal determinism is much better behaved categorically, albeit only with respect to a subcategory of **ES**. Causal determinism was studied under the name of *determinism* in [7, 8].

4.1 Definition (causal determinism). An event structure $\mathcal{E}$ is called *causally deterministic* if for all $d, e \in E_{\mathcal{E}}$, $d \sim e$ implies $d = e$.

The class of causally deterministic event structures will be denoted $\mathbf{ES}_{\rm cd}$. For instance, of the event structures in Figure 1, (1) and (4) are causally deterministic. The following is immediate.

4.2 Proposition. $\mathbf{ES}_{\rm cd} \subset \mathbf{ES}_{\rm dd}$.

Note that the inclusion is proper; structure (2) of Figure 1 is an element of $\mathbf{ES}_{\rm dd} - \mathbf{ES}_{\rm cd}$. Below, we reconsider the results we established for denotational determinism in the current, more restrictive setting. First, however, we present a completely different characterisation of causal determinism.

4.1 Causal determinism is determinism of causal trees

The *causal tree* model defined by Darondeau and Degano in [1, 2] enriches the labels of standard trees with additional information about the causal dependency of a transition. It turns out that tree determinism in this enriched model precisely corresponds to causal determinism of event structures. First we recall the model.

4.3 Definition (causal trees). A *causal tree* is a tree $\langle L, S, \rightarrow, \iota\rangle$ where $L = A \times \mathbf{2}^{\mathbb{N}^+}$, i.e., transitions are labelled by pairs of actions and sets of positive natural numbers, the so-called *causes*.

The intuition is that if $s \xrightarrow{a,K} s'$ then the elements of K point back along the (unique) path up to s, determining which transitions in the past were causal predecessors of the current one. Causal trees can be derived from event structures in the following fashion (cf. [2]):

4.4 Definition. Let $\mathcal{E} \in \mathbf{ES}$ be an event structure. The corresponding causal tree is given by $es.ct(\mathcal{E}) = \langle L, S, \rightarrow, \iota\rangle$ where L is as in Definition 4.3 and

- $S \subseteq E_{\mathcal{E}}^*$ is a set of distinct event sequences, such that for all $e_1 \cdots e_n \in S$ and all $i \leq n$, $\{e_1, \ldots, e_i\} \in \mathcal{C}(\mathcal{E})$.
- $s \xrightarrow{a,K} s'$ iff $s = e_1 \cdots e_n$ and $s' = se$ for some $e \in E_{\mathcal{E}}$ such that $\ell_{\mathcal{E}}(e) = a$ and $[\![e]\!] = \{e_{n+1-k} \mid k \in K\}$.
- $\iota = \epsilon$ is the empty event sequence.

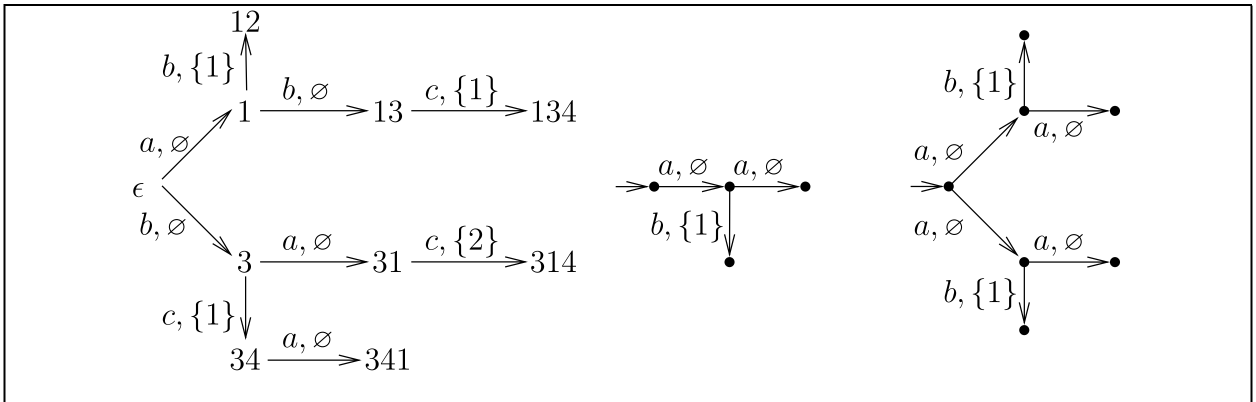
The announced correspondence of determinism in causal trees and causal determinism in event structures is stated formally in the following theorem:

4.5 Theorem. $\mathcal{E}$ is causally deterministic iff $es.ct(\mathcal{E})$ is a deterministic tree.

Proof: if. Assume that $\mathcal{E}$ is not causally deterministic and let $es.ct(\mathcal{E}) = \langle L, S, \rightarrow, \iota\rangle$. Now let $d, e \in E_{\mathcal{E}}$ be two distinct events such that $d \sim e$. Now let $s = e_1 \cdots e_n$ be a linearisation of $[\![e]\!]$ such that $e_i <_{\mathcal{E}} e_j$ implies $i < j$. Then $s \in S$ and also $sd, se \in S$, where $s \xrightarrow{\ell(d),K_d} sd$ and $s \xrightarrow{\ell(e),K_e} se$ with $K_d = \{i \mid e_{n+1-i} < d\}$ and $K_e = \{i \mid e_{n+1-i} < e\}$. Due to $d \sim e$ it follows that $\ell(d) = \ell(e)$ and $K_d = K_e$; however, $sd \neq se$, hence $es.ct(\mathcal{E})$ is not a deterministic tree.

only if. Assume that $es.ct(\mathcal{E}) = \langle L, S, \rightarrow, \iota\rangle$ is not a deterministic tree and let $s \xrightarrow{a,K} s'$ and $s \xrightarrow{a,K} s''$ such that $s' \neq s''$. It follows that $s = e_1 \cdots e_n$, $s' = sd$ and $s'' = se$ for some distinct $d, e \in E_{\mathcal{E}}$. Moreover, $\ell(d) = \ell(e) = a$ and $[\![d]\!] = \{e_{n+1-i} \mid i \in K\} = [\![e]\!]$; hence $d \sim e$, implying that $\mathcal{E}$ is not causally deterministic. □

For instance, causal tree (1) in Figure 8 is derived from structure (1) of Figure 1; see also Figure 2, where it is shown that the standard transition system of this event structure is *not* deterministic. It should be noted that Theorem 4.5 cannot be modified easily to go from deterministic causal trees to event structures, because not all causal trees are modulo isomorphism derived from an event structure. Consider structure (2) in Figure 8: it is



(1) (2) (3)

Figure 8: Some causal trees

deterministic (as a causal tree), but the transitions $\bullet \xrightarrow{a,\varnothing} \bullet \xrightarrow{a,\varnothing} \bullet$ in fact correspond to *concurrent* (because not causally dependent) events, which can only represented by an event structure having two distinct causally indistinguishable events; however, the causal tree derived from that structure is not (2) but (3) of Figure 8.[3]

4.2 Causally deterministic event structures

Causal determinism has the characterising property that there are no non-trivially isomorphic configurations in the model; in other words, the mapping from configurations to concurrent language is injective. This can be seen to rule out precisely the existence of distinct causally indistinguishable events; since these were the prime source of complications in the previous section, this is one indication why the categorical situation improves.

4.6 Theorem. $\mathcal{E} \in \mathbf{ES}_{\mathrm{cd}}$ iff for all $F, G \in \mathcal{C}(\mathcal{E})$, $\mathcal{E} \restriction F \cong \mathcal{E} \restriction G$ implies $F = G$.

Proof: if. Assume that $\mathcal{E}$ is not deterministic; let $d, e \in E_{\mathcal{F}}$ be distinct, causally indistinguishable events. Then $\mathcal{E} \restriction \lceil d \rceil \cong \mathcal{E} \restriction \lceil e \rceil$ whereas $\lceil d \rceil \neq \lceil e \rceil$.

only if. Assume that $\mathcal{E}$ is deterministic, and let $F, G \in \mathcal{C}(\mathcal{E})$ be such that $\phi\colon \mathcal{E} \restriction F \cong \mathcal{E} \restriction G$ and $F \neq G$, and let $e \in F$ be $<$-minimal such that $\phi(e) \neq e$; then apparently $[\![e]\!] = [\![\phi(e)]\!]$. It follows that $e \sim \phi(e)$, and hence $e = \phi(e)$ which contradicts the assumptions. □

The concurrent words of causally deterministic event structures are themselves causally deterministic, meaning that although they may contain concurrent events with the same label, those may not have precisely the same predecessors. The class of causally deterministic concurrent languages will be denoted $\mathbf{CL}_{\mathrm{cd}}$. We recall some facts about causally deterministic pomsets from [7] in order to facilitate proofs later on. For arbitrary finite sets P of causally deterministic pomsets, there is a least upper bound $\bigvee P$ with respect to pomset prefix. (Note

[3] Note that structures (2) and (3), although not isomorphic, are strongly bisimilar. Possibly denotational determinism of event structures could be captured by determinism of causal trees *up to strong bisimulation.* We do not pursue this further here; see however Section 6.3.

that this does not hold for all pomsets; for instance, $\boxed{\begin{smallmatrix} a\to b \\ a \end{smallmatrix}}$ and $\boxed{a\to c}$ have the incomparable upper bounds $\boxed{\begin{smallmatrix} a\to b \\ \searrow \\ a \quad c \end{smallmatrix}}$ and $\boxed{\begin{smallmatrix} a\to b \\ a\to c \end{smallmatrix}}$.) In fact, $\bigvee P$ is easily constructed, given an appropriate choice of representatives:

$$\bigvee P = \left[\bigcup_{[p]\in P} E_p, \bigcup_{[p]\in\mathcal{L}} <_p, \bigcup_{[p]\in P} \ell_p\right]$$

where the representatives $[p],[q] \in P$ are chosen such that if $\llbracket d \rrbracket = \llbracket e \rrbracket$ and $\ell_p(d) = \ell_q(e)$ for some $d \in E_p, e \in E_q$ then $d = e$. (Such representatives exist by virtue of causal determinism.) Moreover, $[p] = \bigvee\{[q] \preceq [p] \mid [q] \text{ is topped}\}$ for all causally deterministic $[p]$.[4] We now get the following counterpart to Theorem 3.4.

4.7 Theorem. $\mathcal{L} \in \mathbf{CL}_{\mathrm{cd}}$ iff $\mathcal{L} = es.cl(\mathcal{E})$ for some $\mathcal{E} \in \mathbf{ES}_{\mathrm{cd}}$.

Proof. The "if" part is immediate. The "only if" in fact follows from Theorem 3.4 and Theorem 4.6; however, we give the construction of $\mathcal{E}$ explicitly for the present, much simpler case. Namely, there is a bijective correspondence between the topped pomsets of $\mathcal{L}$ and the events of $\mathcal{E}$. We define

$$\begin{array}{rcl} E_{\mathcal{E}} & = & \{[p] \in \mathcal{L} \mid [p] \text{ is topped}\} \\ <_{\mathcal{E}} & = & \preceq \cap\, (E_{\mathcal{E}} \times E_{\mathcal{E}}) \\ \mathrm{Coh}_{\mathcal{E}} & = & \{P \subseteq E_{\mathcal{E}} \mid \bigvee P \in \mathcal{L}\} \\ \ell_{\mathcal{E}} & = & \{([p], a) \in E_{\mathcal{E}} \times \mathbf{A} \mid a = \ell_p(\top_p)\} \end{array}$$

$es.cl(\mathcal{E}) = \mathcal{L}$ due to the properties of causally deterministic pomsets: $\mathcal{E} \upharpoonright F \in \bigvee F$ for all $F \in \mathcal{C}(\mathcal{E})$ and $[p] = \bigvee\{[q] \in E_{\mathcal{E}} \mid [q] \preceq [p]\}$ for all $[p] \in \mathcal{L}$. □

Naturally, $\mathbf{ES}_{\mathrm{cd}}$ being properly smaller than $\mathbf{ES}_{\mathrm{dd}}$, not every event structure can be causally determinised while retaining its concurrent language. The class of event structures for which this is still possible will be called *causally distinct*.

4.8 Definition (causal distinctness). An event structure $\mathcal{E}$ is called *causally distinct* if for all $d, e \in E_{\mathcal{E}}$, $d \sim e$ implies $d \#^{=} e$.

The class of causally distinct event structures will be denoted $\mathbf{ES}_{\mathrm{cdst}}$. Causal distinctness is easily seen to be equivalent to having only causally deterministic concurrent words. The following is the counterpart to Corollary 3.6.

4.9 Theorem. $\mathcal{E} \in \mathbf{ES}_{\mathrm{cdst}}$ iff there exists a $\mathcal{F} \in \mathbf{ES}_{\mathrm{cd}}$ such that $es.cl(\mathcal{E}) = es.cl(\mathcal{F})$.

Proof. This comes down to showing that $\mathcal{E}$ is causally distinct iff all its concurrent words are causally deterministic, which is straightforward, and then applying Theorem 4.7. □

[4] This is a consequence of the fact that the class of causally deterministic pomsets is a prime algebraic domain with topped pomsets as primes; see [7].

4.3 Causal determinism categorically

With respect to the category theoretical situation, the properties that failed to hold in the general case turn out to be valid when regarded only for causally deterministic and causally distinct event structures.

4.10 Proposition. The mapping $cl.es\colon \mathbf{CL}_{\mathrm{cd}} \to \mathbf{ES}_{\mathrm{cd}}$ gives rise to a functor, with arrow part given by $\lambda \mapsto (\lambda, \eta)$ where

$$\eta\colon [p] \mapsto \begin{cases} \hat{\lambda}([p]) & \text{if } \lambda \text{ is defined on } \ell_p(\top_p) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

the following then is an event structure counterpart to the deterministic tree equivalence in Proposition 2.1.

4.11 Theorem. $(es.cl, cl.es)$ is an equivalence between $\mathbf{ES}_{\mathrm{cd}}$ and $\mathbf{CL}_{\mathrm{cd}}$.

It suffices to show that given a partial mapping from actions to actions, there is at most one morphism between any pair of causally deterministic event structures. In fact, we show the following slightly stronger statement (which we also need for a further result below).

4.12 Lemma. Let $\lambda\colon \mathbf{A} \to \mathbf{A}$ be a partial function. For any $\mathcal{E} \in \mathbf{ES}$ and $\mathcal{F} \in \mathbf{ES}_{\mathrm{cd}}$, there is at most one morphism $(\lambda, \eta)\colon \mathcal{E} \to \mathcal{F}$.

Proof. Assume two different morphisms $(\lambda, \eta), (\lambda, \zeta)\colon \mathcal{E} \to \mathcal{F}$, and let $e \in E_{\mathcal{E}}$ be $<_{\mathcal{E}}$-minimal such that λ is defined on $\ell_{\mathcal{E}}(e)$ and $\eta(e) \neq \zeta(e)$. It follows that $\mathcal{F} \restriction \lceil \eta(e) \rceil \cong \mathcal{E} \restriction \lceil e \rceil \cong \mathcal{F} \restriction \lceil \zeta(e) \rceil$, which implies $\lceil \eta(e) \rceil = \lceil \zeta(e) \rceil$ as a consequence of Theorem 4.6. Since $[\![\eta(e)]\!] = [\![\zeta(e)]\!]$ by the choice of e, it follows that $\eta(e) = \zeta(e)$, which contradicts the assumptions. □

Proof of Theorem 4.11. To prove equivalence, we have to establish bijections, (i) between the morphisms $\mathcal{E} \to cl.es(\mathcal{L})$ and $es.cl(\mathcal{E}) \to \mathcal{L}$ and (ii) between the morphisms $cl.es(\mathcal{L}) \to \mathcal{E}$ and $\mathcal{L} \to es.cl(\mathcal{E})$. Indeed, both bijections are given by $(\lambda, \eta) \leftrightarrow \lambda$; since all event structures around are causally deterministic, η is uniquely determined by λ (Lemma 4.12).

(i) If $(\eta, \lambda)\colon \mathcal{E} \to cl.es(\mathcal{L})$ then $\lambda\colon es.cl(\mathcal{E}) \to \mathcal{L}$ due to Proposition 2.3. On the other hand, if $\lambda\colon es.cl(\mathcal{E}) \to \mathcal{L}$ then $(\lambda, \eta)\colon es.des(\mathcal{E}) \to cl.es(\mathcal{L})$ where $(\lambda, \eta) = cl.es(\lambda)$; since $\phi\colon \mathcal{E} \cong es.des(\mathcal{E})$ for some isomorphism ϕ by Theorem 4.9, it follows that $(\lambda, \phi \circ \eta)\colon \mathcal{E} \to cl.es(\mathcal{L})$.

(ii) Similar to the above. □

The following is an event structure counterpart to the tree determinisation property in Proposition 2.2.

4.13 Theorem. $es.des$ is a concurrent-language-preserving reflection from $\mathbf{ES}_{\mathrm{cdst}}$ to $\mathbf{ES}_{\mathrm{cd}}$.

Proof. The proof comes down to showing that for every causally distinct $\mathcal{E} \in \mathbf{ES}_{\rm cdst}$ there is a *determinising morphism* $(\lambda_\mathcal{E}, \eta_\mathcal{E}) \colon \mathcal{E} \to es.des(\mathcal{E})$ such that for every causally deterministic $\mathcal{F} \in \mathbf{ES}_{\rm cd}$ and morphism $(\mu, \zeta) \colon \mathcal{E} \to \mathcal{F}$, there is a unique $(\pi, \theta) \colon es.des(\mathcal{E}) \to \mathcal{F}$ with $(\mu, \zeta) = (\pi, \theta) \circ (\lambda_\mathcal{E}, \eta_\mathcal{E})$.

We define $\lambda_\mathcal{E} = id_\mathbf{A}$ and $\eta_\mathcal{E} \colon e \mapsto [\mathcal{E} \restriction \lceil e \rceil]$ for all $e \in E_\mathcal{E}$. $(\lambda_\mathcal{E}, \eta_\mathcal{E})$ is easily shown to be a morphism from $\mathcal{E}$ to $es.des(\mathcal{E})$. Since $(\mu, \zeta) \colon \mathcal{E} \to \mathcal{F}$ translates through $es.cl$ to $\mu \colon es.cl(\mathcal{E}) \to es.cl(\mathcal{F})$, and from there by Proposition 4.10 and Theorem 4.11 to $(\mu, \theta) \colon es.des(\mathcal{E}) \to \mathcal{F}$ for some unique θ. But then $(\mu, \theta) \circ (id_\mathbf{A}, \eta_\mathcal{E})$ and (μ, ζ) are both morphisms from $\mathcal{E}$ to $\mathcal{F}$ with action part μ, which by Lemma 4.12 implies $(\mu, \zeta) = (\mu, \theta) \circ (id_\mathbf{A}, \eta_\mathcal{E})$. □

5 Operational determinism

The last of the notions of determinism studied in this paper is the one obtained by observing the *transition structure* of the event structures in question, without taking causality into account. This makes for a stronger property than the previous two. Among other things, event structures can only be operationally deterministic if they contain *no auto-concurrency*, i.e., equilabelled events cannot be concurrent. Operationally deterministic event structures were studied under the name *deterministic event structures* in the aforementioned papers [11, 6]. A number of the results of this section are reconstructed from those papers.

Just how strong the property of operational determinism is has been made clear by Vaandrager in [13], where he shows that operationally deterministic event structures are isomorphic if and only if their set of *step sequences* (i.e., words over sets of actions rather than single actions) are equal. We briefly recall his arguments below.

5.1 Definition (operational determinism). An event structure $\mathcal{E}$ is called *operationally deterministic* if the underlying transition system $es.t(\mathcal{E})$ is deterministic.

The class of operationally deterministic event structures is denoted $\mathbf{ES}_{\rm od}$. The following is immediate.

5.2 Proposition. $\mathbf{ES}_{\rm od} \subset \mathbf{ES}_{\rm cd}$.

Note that the inclusion is proper; structure (1) of Figure 1 is an element of $\mathbf{ES}_{\rm cd} - \mathbf{ES}_{\rm od}$. Below, we reconsider the results we established for denotational determinism in the current, more restrictive setting.

5.1 Operationally deterministic event structures

Operational determinism has an easy characterisation in terms of the relations between events (as also shown by Vaandrager in [13]). Say that in some event structure $\mathcal{E}$, $d, e \in E$ are in *direct conflict*, denoted $d \;\#^! \; e$, if $d \# e$ and for all $d' \leq d$ and $e' \leq e$, $d' \# e'$ implies $d = e$ (in other words, no proper predecessors of d $[e]$ are in conflict with e $[d]$).

5.3 Theorem (see [13, Propositon 3.8]). $\mathcal{E}$ is operationally deterministic if for all $d, e \in E$ such that $\ell(d) = \ell(e)$, if $d \not\leq e \not\leq d$ then $d \# e$ and $\neg(d \;\#^! \; e)$.

In [6] it is shown that the concurrent languages of operationally deterministic event structures are characterised by two properties: (i) no concurrent word may be *auto-concurrent*, and (ii) no pair of distinct concurrent words may share an *augmentation*. A pomset $[p]$ is said to be auto-concurrent if there are $d, e \in E_p$ such that $\ell(d) = \ell(p)$ and $d \not\leq e \not\leq e$.

After [12], we call a concurrent language $\mathcal{L}$ a *semilanguage* if no $[p] \in \mathcal{L}$ is auto-concurrent and a *deterministic semilanguage* if in addition, for all $[p], [q] \in \mathcal{L}$, the existence of a pomset $[p']$ that augments both $[p]$ and $[q]$ implies $p \cong q$. The class of deterministic semilanguages will be denoted $\mathbf{SL}_{\mathrm{d}}$.[5] The following is the counterpart of Theorem 4.7; see also [6, Theorems 4.8 and 4.9].

5.4 Theorem. $\mathcal{L} \in \mathbf{SL}_{\mathrm{d}}$ iff $\mathcal{L} = es.cl(\mathcal{E})$ for some $\mathcal{E} \in \mathbf{ES}_{\mathrm{od}}$.

This in turn gives rise to Vaandrager's result, since the absence of auto-concurrency and common linearisations among a set of pomsets precisely implies that those pomsets can be reconstructed entirely from their *step linearisations* (where a step linearisation of $[p]$ is a sequence $\ell(E_1) \cdots \ell(E_n)$ where the E_i partition E_p such that $E_i \ni d < e \in E_j$ implies $i < j$). Let

$$cl.ss(\mathcal{L}) = \{A_1 \cdots A_n \mid \exists [p] \in \mathcal{L}.\, A_1 \cdots A_n \text{ is a step linearisation of } [p]\} \ ,$$

then in combination with Theorem 4.7, the following is equivalent to [13, Theorem 5.1]:

5.5 Theorem. If $\mathcal{L}, \mathcal{M} \in \mathbf{SL}_{\mathrm{d}}$ then $\mathcal{L} = \mathcal{M}$ iff $cl.ss(\mathcal{L}) = cl.ss(\mathcal{M})$.

Finally, we also characterise the class of event structures that can be determinised operationally (under preservation of the concurrent language).

5.6 Definition (operational distinctness). An event structure $\mathcal{E}$ is called *operationally distinct* if for all $d, e \in E_{\mathcal{E}}$, if $\ell(d) = \ell(e)$ then $d \not\leq e \not\leq d \Rightarrow d \# e$ and $d \#^! e \Rightarrow d \sim e$.

The class of operationally distinct event structures will be denoted $\mathbf{ES}_{\mathrm{odst}}$. It is not difficult to check that for $\mathcal{E} \in \mathbf{ES}_{\mathrm{odst}}$, $es.t(\mathcal{E})$ can only be nondeterministic if $F \xrightarrow{a} F \cup \{d\}$ and $F \xrightarrow{a} F \cup \{e\}$ where $d \sim e$. The following property therefore holds, which is interesting when contrasted with Theorem 4.5:

5.7 Proposition. If $\mathcal{E} \in \mathbf{ES}_{\mathrm{odst}}$, then $es.t(\mathcal{E})$ is deterministic iff $es.ct(\mathcal{E})$ is deterministic.

The following is the counterpart to Theorem 4.9.

5.8 Theorem. $\mathcal{E} \in \mathbf{ES}_{\mathrm{odst}}$ iff there exists an $\mathcal{F} \in \mathbf{ES}_{\mathrm{od}}$ such that $es.cl(\mathcal{E}) = es.cl(\mathcal{F})$.

[5] Note that the absence of common linearisations must be checked on *pairs* of concurrent words; on the other hand, the concurrent languages of causally deterministic event structures are causal, which can be checked "pointwise" on each concurrent word.

5.2 Operational determinism categorically

There are no category theoretic results about operationally deterministic event structures that we had not already established for the larger class of causally deterministic ones; see Section 4.3. The adjunctions we had proved there (Theorems 4.11 and 4.13) simply specialise to the subcategories considered here. (However, see also Section 6 for a discussion of the effect that our choice of morphisms has had on these results.) For the sake of completeness we list the results below. They are special cases of [6, Theorem 4.10] and [6, Theorems 7.3 and 7.16], respectively, except for the phrase "concurrent-language-preserving" in Corollary 5.10.

5.9 Corollary. $\mathbf{ES}_{\mathrm{od}}$ and $\mathbf{SL}_{\mathrm{d}}$ are categorically equivalent.

5.10 Corollary. $\mathbf{ES}_{\mathrm{od}}$ is a concurrent-language-preserving reflective subcategory of $\mathbf{ES}_{\mathrm{odst}}$.

6 Conclusion

6.1 Summary and discussion of the results

We have developed three notions of determinism for event structures, corresponding to three different characterisations of determinism of transition trees. For each of these we have investigated whether the category theoretical properties of deterministic trees, expressed in Propositions 2.1 and 2.2, can be extended to event structures. The results are summarised below.

Denotational determinism corresponds to the view that every event structure should give rise to a deterministic one, unique up to isomorphism, with the same concurrent language. In other words, the correspondence between concurrent languages and denotationally deterministic event structures is one-to-one.

Unfortunately, the *determinisation* of a given event structure is nontrivial, involving the *duplication* of events in the case of *causal indistinguishability*. Mainly because of this duplication, denotationally deterministic event structures do not seem to exhibit many interesting categorical properties.

Causal determinism corresponds to the view that there should be a one-to-one correspondence between the runs of a (causally) deterministic behaviour and its (concurrent) words. For event structures, this is shown to come down to the complete absence of causally indistinguishable events (see above). Causal determinism is strictly stronger than denotational determinism; consequently, the ability to determinise any event structure is necessarily lost.

Causally deterministic event structures share the categorical properties of deterministic trees that they are equivalent (as a category) to the corresponding concurrent languages and that they form a reflective subcategory of the causally distinct event structures. Furthermore, we have shown that an event structure is causally deterministic iff the *causal tree* derived from it is deterministic (in the standard sense for trees).

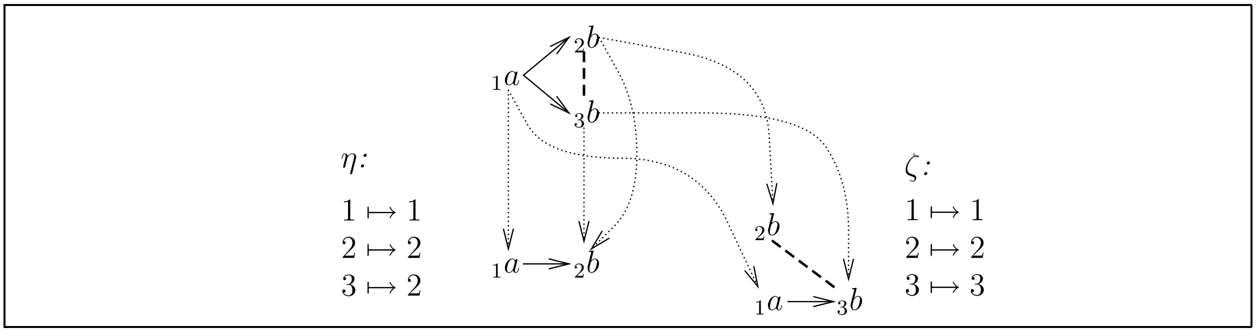


Figure 9: A counterexample to reflection: ζ does not factor through η

Operational determinism corresponds to the view that from every state of the behaviour there should be at most one transition with any given label. For event structures, this was already known (see Vaandrager [13]) to correspond to the absence of auto-concurrency and auto-conflict (where two events are in auto-conflict if they have the same label and none of their predecessors are in conflict). Operational determinism is strictly stronger than causal determinism.

The categorical properties of operationally deterministic event structures are those of causally deterministic ones, restricted to the appropriate subcategories. Hence, in this respect, operational determinism does not yield any additional insights.

The categorical results mentioned above are formulated with respect to our chosen notion of morphism, which, as mentioned before, is more restrictive than the usual one. We briefly discuss how this has affected the outcome of our investigation.

The standard notion of event structure morphism (see [6, 14]) allows to *forget causality*, i.e., only requires $\eta(\llbracket e \rrbracket_{\mathcal{E}}) \supseteq \llbracket \eta(e) \rrbracket_{\mathcal{F}}$ rather than equality, as we have done. Then configurations $F, G \in \mathcal{C}(\mathcal{E})$ yielding identical concurrent words (i.e., such that $\mathcal{E} \restriction F \cong \mathcal{E} \restriction G$) can be mapped to non-isomorphic configurations of $\mathcal{F}$ (i.e., such that $\mathcal{F} \restriction \eta(F) \ncong \mathcal{F} \restriction \eta(G)$), which situation cannot in general be reflected in their deterministic counterparts, since there F and G have just been collapsed in the process of determinisation. Figure 9 shows an example. Summarised, this more general notion of morphism has the following effect on the categorical results of this paper.

- The reflection of $\mathbf{ES}_{\mathrm{cdst}}$ in $\mathbf{ES}_{\mathrm{cd}}$ (Theorem 4.13) is lost. Its restriction to the operational case (Corollary 5.10), however, still holds, as a corollary of a result proved in [6] which we recall below. Explained in terms of the discussion above, $[\mathcal{F} \restriction \eta(F)]$ and $[\mathcal{F} \restriction \eta(G)]$ have $[\mathcal{E} \restriction \{e \in F \mid \eta(e) \text{ is defined}\}]$ as a common augmentation; hence if $\mathcal{F}$ is operationally deterministic then Theorem 5.4 implies that $\mathcal{F} \restriction \eta(F) \cong \mathcal{F} \restriction \eta(G)$ after all.

- The equivalence of $\mathbf{ES}_{\mathrm{cd}}$ to the causal concurrent languages $\mathbf{CL}_{\mathrm{cd}}$ (Theorem 4.11) can be generalised; we have worked this out in a separate paper [9]. Consequently, this is also true of its restriction to the operational case (Corollary 5.9) —which special case was in fact already proved in [12, 6].

- On the other hand, precisely because morphisms may forget causality, a reflection does not necessarily have to retain the concurrent language. In fact, one of the main results of [6] is that a reflection from the *entire* **ES** to $\mathbf{ES}_{\mathrm{od}}$ exists under these circumstances. This reflection first *forgets* all events with labels that occur auto-concurrently, and then effectively *merges* classes of concurrent words that share an augmentation, constructing for each such class P the *least augmented* word of which all elements of P are augmentations, i.e., the least upper bound w.r.t. $\sqsubseteq$.

It should be noted that, as a matter of course, the more restricted notion of morphism chosen in this paper affects some other categorical constructions as well. In particular, the *product* in our categories is no longer guaranteed to exist and hence can no longer be used to model synchronisation (although the coproduct still models choice). Indeed, in contrast to operationally deterministic event structures, causally and denotationally deterministic ones are not closed with respect to synchronisation: for instance, the synchronisation of the causally deterministic $\boxed{\begin{smallmatrix} a \to b \\ a \end{smallmatrix}}$ and $\boxed{a \to b \to c}$ over a, b yields (among others) the concurrent word $\boxed{\begin{smallmatrix} a \to b & \\ \searrow & \\ & b \to c \end{smallmatrix}}$, which is not causally deterministic; the corresponding event structure is not even denotationally deterministic.

6.2 Related work: Petri net unfoldings

Apart from the work of Sassone, Nielsen and Winskel on the one hand and Vaandrager on the other, which have been discussed extensively above, there is one field of research from which there exists a somewhat tenuous connection to this paper: namely, that of *Petri net unfoldings* according to the so-called *individual token philosophy*, as investigated by Engelfriet in [3] and by Sassone with Meseguer and Montanari in [4, 10]. The subclass of non-safe P/T-nets for which unfoldings can be defined smoothly (namely those where the initial marking and the post-places of any transition are sets rather than proper multisets) can be seen to give rise to *causally deterministic event structures* if one takes the event structure corresponding to the occurrence net derived in [3, 4] and labels its events with the transitions of the original non-safe net. This notion of unfolding is known to be quite hard to extend to all Petri nets, however; see [4, 10]. Now, it is interesting to note that under the same notion of labelling, a naive unfolding of general Petri nets would yield *denotationally deterministic events structures*; see Figure 10 for an example. Our strong feeling is that the problems encountered in unfolding general Petri nets are precisely the same as the ones involved in the categorical treatment of denotationally determinism (see Section 3.2). In particular, the absence of a notion of event structure determinism that gives rise to a category equivalent to (general) concurrent languages could very well be directly related to the difficulty in unfolding general Petri nets. If this feeling is justified, then the investigation of denotational determinism in *causal trees* proposed below might also shed light on Petri net unfoldings.

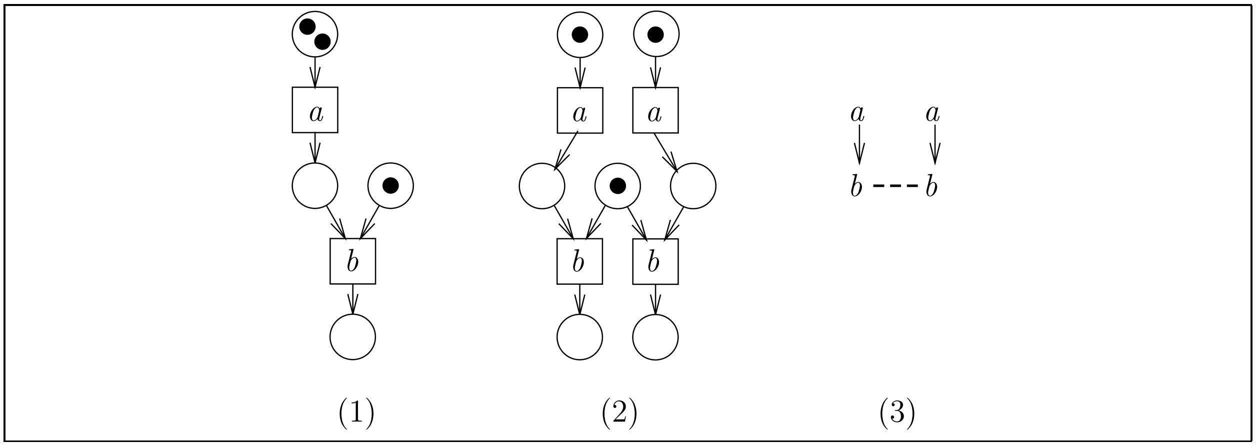


Figure 10: A non-safe Petri net (1), its naive unfolding (2), and the derived denotationally deterministic event structure (3)

6.3 Future work

The results of this paper point out directions of further research, which in part has been carried out already. Especially the connection to *causal trees*, touched upon briefly in Section 4.1, deserves further investigation. In particular, the notion of *denotational determinism* might be captured more effectively by causal trees than by event structures. Consider: causally indistinguishable events may appear in a deterministic causal tree as successive transitions $s_1 \xrightarrow{a,K_1} s_2 \xrightarrow{a,K_2} s_3$ where $K_1 = \{i+1 \mid i \in K_1\}$. The corresponding event structure, however, when transformed into a causal tree, would generate a *second* sub-path $s_1 \xrightarrow{a,K_1} s'_2 \xrightarrow{a,K_2} s'_3$ where $s_2 \neq s'_2$, and hence the result would not be deterministic as a tree. Conversely, this means that the duplication of events during denotational determinisation can be avoided at least partly, and maybe completely, if one goes to causal trees instead. See also Figure 8.

Part of the above investigation has been carried out in [9], which properly generalises the results of [12], namely the categorical equivalence between deterministic semilanguages, generalised Mazurkiewicz trace languages, and operationally deterministic event structures. We show a similar equivalence between causal concurrent languages, restricted causal trace languages, and causally deterministic event structures, which partly generalises even further to (general) concurrent languages and (general) causal trace languages (however, in the latter part the event structure angle is absent).

As a possible further consequence of this line of research, we intend to investigate if the framework of models proposed by Sassone, Nielsen and Winskel might not be improved if one replaces event structures with causal trees. In particular, it might be possible to get rid of the need to forget auto-concurrent events when moving from nondeterministic to deterministic behavioural models of concurrency.

Acknowledgement. Thanks to Roberto Gorrieri and Frits Vaandrager for clarifying some of the issues of operationally deterministic event structures.

References

[1] P. Darondeau and P. Degano. Causal trees. In G. Ausiello, M. Dezani-Ciancaglini, and S. Ronchi Della Rocca, editors, *Automata, Languages and Programming*, volume 372 of *Lecture Notes in Computer Science*, pages 234–248. Springer-Verlag, 1989.

[2] P. Darondeau and P. Degano. Causal trees = interleaving + causality. In I. Guessarian, editor, *Semantics of Systems of Concurrent Processes*, volume 469 of *Lecture Notes in Computer Science*, pages 239–255. Springer-Verlag, 1990.

[3] J. Engelfriet. Branching processes of Petri nets. *Acta Inf.*, 28:575–591, 1991.

[4] J. Meseguer, U. Montanari, and V. Sassone. On the semantics of Petri nets. In W. R. Cleaveland, editor, *Concur '92*, volume 630 of *Lecture Notes in Computer Science*, pages 286–301. Springer-Verlag, 1992.

[5] M. Nielsen, G. D. Plotkin, and G. Winskel. Petri nets, event structures and domains, part 1. *Theoretical Comput. Sci.*, 13:85–108, 1981.

[6] 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*, volume 803 of *Lecture Notes in Computer Science*, pages 425–476. Springer-Verlag, 1994.

[7] A. Rensink. Deterministic pomsets. Hildesheimer Informatik-Berichte 30/94, Institut für Informatik, Universität Hildesheim, Nov. 1994.

[8] A. Rensink. A complete theory of deterministic event structures. In I. Lee and S. A. Smolka, editors, *Concur '95: Concurrency Theory*, volume 962 of *Lecture Notes in Computer Science*, pages 160–174. Springer-Verlag, 1995.

[9] A. Rensink. Further deterministic behavioural models for concurrency. Hildesheimer Informatik-Berichte 30/95, Institut für Informatik, Universität Hildesheim, Oct. 1995.

[10] V. Sassone. *On the Semantics of Petri Nets: Processes, Unfoldings and Infinite Computations*. PhD thesis, Universitá de Pisa, Mar. 1994. Available as TD–6/94.

[11] V. Sassone, M. Nielsen, and G. Winskel. A classification of models for concurrency. In E. Best, editor, *Concur '93*, volume 715 of *Lecture Notes in Computer Science*, pages 82–96. Springer-Verlag, 1993. Extended abstract.

[12] V. Sassone, M. Nielsen, and G. Winskel. Deterministic behavioural models for concurrency. In A. M. Borzyszkowski and S. Sokołowski, editors, *Mathematical Foundations of Computer Science 1993*, volume 711 of *Lecture Notes in Computer Science*, pages 682–692. Springer-Verlag, 1993. Extended abstract.

[13] F. W. Vaandrager. Determinism $\longrightarrow$ (event structure isomorphism = step sequence equivalence). *Theoretical Comput. Sci.*, 79:275–294, 1991.

[14] G. Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Applications and Relationships to Other Models of Concurrency*, volume 255 of *Lecture Notes in Computer Science*, pages 325–392. Springer-Verlag, 1987.