Available online at www.sciencedirect.com
SCIENCE DIRECT
Electronic Notes in Theoretical Computer Science 154 (2006) 3–18
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Causality Versus True-Concurrency

Sibylle Fröschle[1,2] and Sławomir Lasota[3,4]

Institute of Informatics, Warsaw University
02–097 Warszawa, Banacha 2, Poland

Abstract

Category theory has been successfully employed to structure the confusing setup of models and equivalences for concurrency: Winskel and Nielsen have related the standard models via adjunctions and (co)reflections while Joyal et al. have defined an abstract notion of equivalence, known as open map bisimilarity. One model has not been integrated into this framework: the causal trees of Darondeau and Degano. Here we fill this gap. In particular, we show that there is an adjunction from causal trees to event structures, which we bring to light via a mediating model, that of event trees. Further, we achieve an open map characterization of history preserving bisimilarity: the latter is captured by the natural instantiation of the abstract bisimilarity for causal trees.

Keywords: Event structures, causal trees, history preserving bisimulation.

In [6] Winskel and Nielsen employ category theory to relate and unify the many models for concurrency. The basic idea is to represent models as categories: each model is equipped with a notion of morphism that shows how one model instance can be simulated by another. Category theoretical notions such as adjunctions and (co)reflections can then be applied to understand the relationships between the models. We give an example. *Synchronization trees* are intuitively those transition systems with no cyclic behaviour. Formally, the two models are related by a coreflection: the inclusion functor embedding synchronization trees into transition systems is accompanied by a right adjoint that unfolds transition systems to synchronization trees.

The categorical approach has also been applied to bring uniformity to the confusing setup of behavioural equivalences. Joyal et al. define an abstract notion of bisimilarity in the following way [4]: given a category of models $\mathbf{M}$ and a choice of path category $\mathbf{P}$ within $\mathbf{M}$, two model instances of $\mathbf{M}$ are $\mathbf{P}$-*bisimilar* iff there is a span of $\mathbf{P}$-*open maps* between them. $\mathbf{P}$-open maps are morphisms that satisfy a

[1] Partially supported by the EU Research Training Network GAMES.

[2] Email: sib@mimuw.edu.pl

[3] Partially supported by the EU project SENSORIA within the IST 6th Framework Programme.

[4] Email: sl@mimuw.edu.pl

doi:10.1016/j.entcs.2006.05.003

special path-lifting property with respect to **P**. As one would expect, on transition systems and synchronization trees the abstract bisimilarity gives rise to classical bisimilarity [4]. Various well-known equivalences are motivated as instantiations of **P**-bisimilarity in a natural way [2].

Winskel and Nielsen's framework has helped to clarify the connections between truly-concurrent models such as event structures, asynchronous transition systems, and Petri nets. These are all independence models: they have additional structure which shows when two transitions are independent of each other. Common to these models is that they come with a notion of *event*: given two runs r_1, r_2 and two transitions t_1 on r_1, t_2 on r_2 it is possible to tell whether t_1 and t_2 represent two occurrences of the same event and can thus be considered equivalent modulo independent behaviour. The notion of event is primary in event structures; they can be considered to be *the* independence model for unfolded behaviour.

On independence models **P**-bisimilarity was shown to yield *hereditary history preserving bisimilarity* (*hhp-b*) [4]. This left open whether it is at all possible to capture *history preserving bisimilarity* (*hp-b*) via open maps, which was then thought to be *the* truly-concurrent bisimilarity. In particular, it was found that the characterization of hhp-b is very robust with respect to the choice of path category.

Along a different strand of research, a new model emerged in the late 80's: the *causal trees* of Darondeau and Degano [3]. They are a variant of synchronization trees with enriched action labels that supply information about which transitions are causally dependent on each other. Thereby, they reflect one aspect of true-concurrency, causality, while being different from the truly-concurrent models of [6] in that they do not come with a notion of event. However, the precise relationship between causal trees and the standard models has never been clarified.

Roughly one could say the strand of research along which causal trees have emerged is that of syntax-enriched process calculi. A unifying framework for a wide range of such calculi, including the π-calculus, has been provided by the *history-dependent automata* of Pistore [5]. In this context a first, albeit indirect, open map account of hp-b has been achieved: in [5] *history-dependent bisimilarity*, which induces hp-b with respect to Petri nets, is captured via open maps. It has remained open, though, whether hp-b has a direct open map characterization: one that is as natural as that of hhp-b and illustrates the difference between the two equivalences, one within a model related to event structures.

Our contribution is twofold. Firstly, we integrate the model of causal trees into Winskel and Nielsen's framework. We equip causal trees with a notion of morphism, and thus define the category of causal trees, **C**. We investigate how **C** relates to the other model categories. In particular, we show that there is an adjunction from causal trees to event structures. This is brought to light via a larger model, called *event trees*: the adjunction arises as the composition of a coreflection from causal trees to event trees and a reflection from event trees to event structures.

Secondly, we identify the natural instantiation of **P**-bisimilarity for causal trees: $\mathbf{CBran}_L$-bisimilarity. It turns out that $\mathbf{CBran}_L$-bisimilarity fills in a prominent gap: it characterizes hp-b in a direct fashion. Finally, we capture the difference between hp-b and hhp-b by characterizing them within the category of event trees.

1 Relating Causal Trees to Other Models for Concurrency

We first define the category of transition systems, **T**, and that of synchronization trees, **S**.

A *transition system* is a tuple $(S, s^{in}, L, \mathit{Tran})$ where S is a set of *states*, $s^{in} \in S$ is the *initial state*, L is a set of *labels*, and $\mathit{Tran} \subseteq S \times L \times S$ is the *transition relation*. We write $s \xrightarrow{a} s'$ to denote that $(s, a, s') \in \mathit{Tran}$. We extend this notation to possibly empty strings of labels $v = a_1 \dots a_n$ writing $s \xrightarrow{v} s'$ to indicate that $s_0 \xrightarrow{a_1} s_1 \cdots \xrightarrow{a_n} s_n$ for some $s_0, \dots, s_n$ with $s = s_0$ and $s_n = s'$. Given $t = (s, a, s') \in \mathit{Tran}$, we use $\mathit{src}(t)$ for s, $\mathit{tgt}(t)$ for s', and $l(t)$ for a.

A *run* of a transition system T is a sequence of transitions $t_1 t_2 \dots t_n$, $n \geq 0$, such that if $n > 0$ then $\mathit{src}(t_1) = s^{in}$ and for all $i \in [1, n-1]$ $\mathit{tgt}(t_i) = \mathit{src}(t_{i+1})$. We denote the set of runs of T by $\mathit{Runs}(T)$.

Let $T_0 = (S_0, s_0^{in}, L_0, \mathit{Tran}_0)$ and $T_1 = (S_1, s_1^{in}, L_1, \mathit{Tran}_1)$ be transition systems. A morphism $f : T_0 \to T_1$ is a pair $f = (\sigma, \lambda)$ where $\sigma : S_0 \to S_1$ is a function and $\lambda : L_0 \rightharpoonup L_1$ is a partial function such that

(i) $\sigma(s_0^{in}) = s_1^{in}$,

(ii) $(s, a, s') \in \mathit{Tran}_0$ & $\lambda(a)$ defined $\Longrightarrow (\sigma(s), \lambda(a), \sigma(s')) \in \mathit{Tran}_1$,
$(s, a, s') \in \mathit{Tran}_0$ & $\lambda(a)$ undefined $\Longrightarrow \sigma(s) = \sigma(s')$.

Transition systems together with their morphisms form a category **T**. Composition of morphisms is pairwise and identity for an object T is $(1_S, 1_L)$ where 1_S is identity on the set of states S of T and 1_L is identity on the set of labels L of T.

A *synchronization tree* is a transition system $(S, s^{in}, L, \mathit{Tran})$ such that

(i) every state is reachable: $\forall s \in S.\ \exists v.\ s^{in} \xrightarrow{v} s$,

(ii) the transition system is acyclic: $s \xrightarrow{v} s$ for some $v \in L^* \Longrightarrow v = \varepsilon$,

(iii) there is no backwards branching: $s' \xrightarrow{a} s$ & $s'' \xrightarrow{b} s \Longrightarrow a = b$ & $s' = s''$.

Write **S** for the full subcategory of synchronization trees in **T**.

We define *causal trees* explicitly as a generalization of synchronization trees. In particular, this means: we add causality information not via enriched labelling and backwards pointers as in [3] but by a causal dependency relation on transitions.

Definition 1.1 A *causal tree* is a tuple $(S, s^{in}, L, \mathit{Tran}, <)$ where $(S, s^{in}, L, \mathit{Tran})$ is a synchronization tree and $< \subseteq \mathit{Tran} \times \mathit{Tran}$, the *causal dependency relation*, is a strict order, which satisfy:

(i) for all $t, t' \in \mathit{Tran}$, $t < t' \Longrightarrow \mathit{tgt}(t) \xrightarrow{v} \mathit{src}(t')$ for some $v \in L^*$.

Axiom (i) expresses a natural property of causality: if t is a cause of t' then t must have happened before t'. Causal trees inherit their notion of *run* from that of transition systems. We say two transitions t, $t' \in \mathit{Tran}$ are *consistent*, denoted by $t\ Con\ t'$, iff they appear on the same branch: $t\ Con\ t' \iff t = t' \vee \exists v \in L^*.\ tgt(t) \xrightarrow{v} src(t') \vee tgt(t') \xrightarrow{v} src(t)$.

The morphisms of the truly-concurrent models of [6] preserve concurrency. Let t, t' be consistent transitions of a causal tree C; t and t' are *concurrent* iff they are not identical and they are not related by $<$. Note that in contrast to event-based models, here concurrency is only meaningful when interpreted with respect to a branch. Thus, we define causal tree morphisms as follows.

Definition 1.2 Let $C_0 = (S_0, s_0^{in}, L_0, Tran_0, <_0)$, $C_1 = (S_1, s_1^{in}, L_1, Tran_1, <_1)$ be causal trees. A morphism $f : C_0 \to C_1$ is a morphism of transition systems $(\sigma, \lambda) : (S_0, s_0^{in}, L_0, Tran_0) \to (S_1, s_1^{in}, L_1, Tran_1)$ such that σ preserves concurrency:

(i) for all $t = (s, a, s')$, $t' = (u, b, u') \in Tran_0$ such that $t\ Con_0\ t'$, and $\lambda(a)$, $\lambda(b)$ are both defined, $(\sigma(s), \lambda(a), \sigma(s')) <_1 (\sigma(u), \lambda(b), \sigma(u')) \Longrightarrow t <_0 t'$.

Causal trees and their morphisms give rise to the *category of causal trees*, **C**.

There is an obvious coreflection from **S** to **C**: a synchronization tree can be regarded as a causal tree, one in which the causal dependency relation is given by the order of the transitions in the tree; the corresponding functor is accompanied by a right adjoint which forgets about the causality information. It is more difficult to understand the precise relationship between causal trees and event structures. We first give the definition of the category of event structures, **E**.

A *(labelled) event structure* is a structure $(E, <, Con, L, l)$ consisting of a set E of *events*, which are strictly ordered[5] by $<$, the *causal dependency relation*, a *consistency relation* Con consisting of finite subsets of events, a set L of *labels* and a *labelling function* $l : E \to L$, which satisfy

(i) $e{\downarrow} = \{e' \mid e' < e\}$ is finite,

(ii) $\{e\} \in Con$,

(iii) $Y \subseteq X \in Con \Rightarrow Y \in Con$,

(iv) $X \in Con\ \&\ e < e' \in X \Rightarrow X \cup \{e\} \in Con$,

for all events e, e' and their subsets X, Y. Axiom (i) ensures an event occurrence depends only on finitely many previous event occurrences. The consistency relation is thought to specify which finite subsets of events can occur together in a run. Axioms (ii)–(iv) express natural properties of this interpretation.

To define a *run* of an event structure $(E, <, Con, L, l)$, we need the notion of *configuration*, defined as any finite[6] set $X \subseteq E$ which is

(i) downwards-closed: $e' < e \in X \Rightarrow e' \in X$, and

(ii) consistent: $X \in Con$.

In particular, $e{\downarrow}$ is always a configuration. For two configurations X, X' we write $X \xrightarrow{e} X'$ when $e \notin X$ and $X' = X \cup \{e\}$. A *run* is a possibly empty sequence $e_1 \ldots e_n$ of events such that there is a sequence of transitions $\emptyset \xrightarrow{e_1} X_1 \ldots \xrightarrow{e_n} X_n$

[5] Defining causal dependency in terms of a strict rather than a partial order is more convenient here.

[6] We deliberately restrict ourselves to finite configurations only.

starting from the empty configuration, for some configurations $X_1 \ldots X_n$. For runs, $r \xrightarrow{e} r'$ means $r' = re$. The set of all runs of an event structure E is denoted by $Runs(E)$. Let $E_0 = (E_0, <_0, Con_0, L_0, l_0)$ and $E_1 = (E_1, <_1, Con_1, L_1, l_1)$ be labelled event structures. A morphism $E_0 \to E_1$ is a pair (η, λ) where $\eta : E_0 \rightharpoonup E_1$ and $\lambda : L_0 \rightharpoonup L_1$ are partial functions such that

(i) $\eta(e)$ defined $\Rightarrow \eta(e){\downarrow} \subseteq \eta(e{\downarrow})$,

(ii) $X \in Con_0 \Rightarrow \eta(X) \in Con_1$,

(iii) $\forall e, e' {\in} E_0.\ \{e, e'\} {\in} Con_0\ \&\ \eta(e),\ \eta(e')$ both defined $\&\ \eta(e) = \eta(e') \Rightarrow e = e'$,

(iv) $\lambda \circ l_0 = l_1 \circ \eta$.

Event structures and their morphisms form the *category of event structures*, **E**.

The runs of an event structure give rise to a tree. Thus, any event structure can be transformed into a causal tree by abstracting away the notion of event; this operation has been defined in, e.g., [3]. On the other hand, there is no uniform way of reconstructing the notion of event so as to obtain a coreflection between **C** and **E**. Indeed, there is one aspect in which event structures are less expressive than causal trees: their notion of run is induced abstractly by the consistency and causal dependency relation; in particular, this means the set of runs of any event structure is *trace-closed*, that is closed under the shuffling of concurrent transitions. In the following, we expose an adjunction from **C** to **E** via a larger model, which we call *event trees*, that embeds **C** as well as **E**. Event trees are like event structures in that causality and concurrency are event-based, global notions. They are like causal trees in that their possible runs are specified explicitly by a tree.

Definition 1.3 A *(labelled) event tree* is a tuple $(S, s^{in}, E, Tran, <, L, l)$ where $(S, s^{in}, E, Tran)$ is a synchronization tree, $< \subseteq E \times E$ is a strict order on the set E of *events*, L is a set of *labels*, and $l : E \to L$ is a *labelling function* such that

(i) $e \in E \Rightarrow \exists s, s' \in S.\ s \xrightarrow{e} s'$,

(ii) $s \xrightarrow{e} s'\ \&\ s \xrightarrow{e} s'' \Rightarrow s' = s''$,

(iii) $s \xrightarrow{e} s'\ \&\ u \xrightarrow{e} u' \Rightarrow \nexists v \in E^*.\ s' \xrightarrow{v} u$,

(iv) $e < e'\ \&\ s \xrightarrow{e'} s' \Rightarrow \exists u \xrightarrow{e} u',\ v \in E^*.\ u' \xrightarrow{v} s$.

Axiom (i) says every event appears as a transition, and axiom (ii) that the occurrence of an event at a state leads to a unique state. (This is as for asynchronous transition systems.) Axiom (iii) expresses a natural property of acyclic models: every event appears at most once on a branch. Axiom (iv) ensures that if e is a cause of e' then e must have happened before e'. We say two events e, e' are *consistent* iff they appear on the same branch: $e\ Con\ e' \iff e = e' \vee \exists s, s_1, s_2, s_3 \in S, v \in E^*.\ s \xrightarrow{e} s_1 \xrightarrow{v} s_2 \xrightarrow{e'} s_3 \vee s \xrightarrow{e'} s_1 \xrightarrow{v} s_2 \xrightarrow{e} s_3$. Event trees inherit a notion of *run* from synchronization trees, where a run is a sequence of consecutive transitions. By axiom (ii) the sequence of events appearing along a run determines this run uniquely. Hence, we consider a run of an event tree to be a sequence of events rather than one of transitions.

A partial function $\eta : E_0 \rightharpoonup E_1$ induces a total function $\bar{\eta} : E_0^* \to E_1^*$ defined inductively by: $\bar{\eta}(\varepsilon) = \varepsilon$, and $\bar{\eta}(re) = \bar{\eta}(r)\eta(e)$ if $\eta(e)$ defined, and $\bar{\eta}(r)$ otherwise.

Definition 1.4 Assume two event trees $T_0 = (S_0, s_0^{in}, E_0, \mathit{Tran}_0, <_0, L_0, l_0)$, $T_1 = (S_1, s_1^{in}, E_1, \mathit{Tran}_1, <_1, L_1, l_1)$. A morphism from T_0 to T_1 is a pair (η, λ) where $\eta : E_0 \rightharpoonup E_1$ and $\lambda : L_0 \rightharpoonup L_1$ are partial functions such that

(i) $\eta(e)$ defined $\Rightarrow \eta(e)\!\downarrow\, \subseteq \eta(e\!\downarrow)$,

(ii) $r \in \mathit{Runs}(T_0) \Rightarrow \bar{\eta}(r) \in \mathit{Runs}(T_1)$,

(iii) $\lambda \circ l_0 = l_1 \circ \eta$.

Clause (ii) implies that we also have: $\forall e, e' \in E_0.\ e\ \mathit{Con}_0\ e'\ \&\ \eta(e),\ \eta(e')$ both defined & $\eta(e) = \eta(e') \Rightarrow e = e'$. This is analogous to clause (iii) of event structure morphisms.

If $(\eta, \lambda) : T_0 \to T_1$ is a morphism of event trees then $\bar{\eta}$ maps $\mathit{Runs}(T_0)$ to $\mathit{Runs}(T_1)$. Since each state of an event tree is reachable by a unique run, $\bar{\eta}$ induces a total function, say σ_η, from S_0 to S_1. It is routine to check:

Proposition 1.5 *If $(\eta, \lambda) : T_0 \to T_1$ is a morphism of event trees then (σ_η, η) is a morphism of transition systems $(S_0, s_0^{in}, E_0, \mathit{Tran}_0) \to (S_1, s_1^{in}, E_1, \mathit{Tran}_1)$ such that η preserves concurrency: $\forall e,\ e' \in E_0.\ e\ \mathit{Con}_0\ e'\ \&\ \eta(e),\ \eta(e')$ both defined & $\eta(e) <_1 \eta(e') \Rightarrow e <_0 e'$.*

Event trees and their morphisms give rise to the *category of event trees*, **ET**.

Any event tree gives rise to a causal tree by forgetting about events. Considering axiom (i) of causal trees, we carry over the causal dependency relation from events to *consistent* transitions only. Extending this operation to a functor $\mathit{et2c} : \mathbf{ET} \to \mathbf{C}$ we make use of Prop. 1.5 in our translation of morphisms.

Definition 1.6 Let $T = (S_T, s_T^{in}, E_T, \mathit{Tran}_T, <_T, L_T, l_T)$ be an event tree. Define $\mathit{et2c}(T) = (S_T, s_T^{in}, L_T, \mathit{Tran}, <)$ where

- $\mathit{Tran} = \{(s, l_T(e), s') \mid s \xrightarrow{e}_T s'\}$, and
- $< \;= \{((s, l_T(e), s'), (u, l_T(e'), u')) \mid s \xrightarrow{e}_T s',\ u \xrightarrow{e'}_T u',\ e <_T e'\ \&$
$\exists v \in E_T^*.\ s' \xrightarrow{v}_T u\}$.

Let $f = (\eta, \lambda)$ be a morphism of event trees. Define $\mathit{et2c}(f) = (\sigma_\eta, \lambda)$.

On the other hand, every causal tree C determines an event tree: that induced by C when we assume each transition of C represents a separate event. We take as events the transitions of C, and label each arc of C by the corresponding transition. This operation extends to a functor $\mathit{c2et} : \mathbf{C} \to \mathbf{ET}$.

Definition 1.7 Let $C = (S_C, s_C^{in}, L_C, \mathit{Tran}_C, <_C)$ be a causal tree. Let $\mathit{c2et}(C) = (S_C, s_C^{in}, \mathit{Tran}_C, \mathit{Tran}, <_C, L_C, l)$ where

- $\mathit{Tran} = \{(s, (s, a, s'), s') \mid s \xrightarrow{a}_C s'\}$, and
- l is given by $l(s, a, s') = a$.

For $f = (\sigma, \lambda) : C_0 \to C_1$, define $\mathit{c2et}(f) = (\eta, \lambda)$ where $\eta : \mathit{Tran}_0 \to \mathit{Tran}_1$ is given by: $\eta(s, a, s') = \begin{cases} (\sigma(s), \lambda(a), \sigma(s')) & \text{if } \lambda(a) \text{ is defined,} \\ \text{undefined} & \text{otherwise.} \end{cases}$

Theorem 1.8 *The functor c2et is left adjoint to et2c. The adjunction is a coreflection, i.e., the unit is a (natural) isomorphism.*

Proof. [Sketch] Let C be a causal tree. Then $et2c(c2et(C)) = C$, and the unit of the adjunction at C, η_C, is the pair of identities $(1_S, 1_L)$.

The pair $(c2et(C), \eta_C)$ is free over C wrt. *et2c*, i.e. for any arrow $(\sigma, \lambda) : C \to et2c(T)$ in $\mathbf{C}$, with T an event tree, there is a unique arrow $f : c2et(C) \to T$ in $\mathbf{ET}$ such that $et2c(f) \circ (1_S, 1_L) = (\sigma, \lambda)$: the label component of f is necessarily λ, and the event component of f is determined uniquely since events of $c2et(C)$ are transitions of C. □

As a consequence, $\mathbf{C}$ embeds fully and faithfully into $\mathbf{ET}$ and is equivalent to the full subcategory of $\mathbf{ET}$ consisting of those event trees T that are isomorphic to $c2et(et2c(T))$. These event trees T are exactly those in which each event occurs only once.

The runs of an event structure can be arranged into a tree. Hence, any event structure forms an event tree whose states are the runs of the event structure. This gives rise to a functor $e2et : \mathbf{E} \to \mathbf{ET}$.

Definition 1.9 Let $E = (E_E, <_E, Con_E, L_E, l_E)$ be an event structure. Define $e2et(E) = (Runs(E), \varepsilon, E_E, \to_E, <_E, L_E, l_E)$. On morphisms, $e2et(f) = f$.

On the other hand, any event tree determines an event structure: we define a set of events to be consistent iff they appear together on some branch, and, having extracted this information, we forget about the tree structure. Thereby we obtain a functor $et2e : \mathbf{ET} \to \mathbf{E}$.

Definition 1.10 Let $T = (S_T, s_T^{in}, E_T, Tran_T, <_T, L_T, l_T)$ be an event tree. Define $et2e(T) = (E_T, <_T, Con, L_T, l_T)$ where Con exactly contains all sets $\{e_1, \ldots, e_n\}$ such that $s_1 \overset{e_1}{\to} s_1' \overset{v_1}{\to} s_2 \overset{e_2}{\to} \ldots \overset{v_{n-1}}{\to} s_n \overset{e_n}{\to} s_n'$ in T, for some states $s_1 \ldots s_n$, $s_1' \ldots s_n'$ and sequences of events $v_1 \ldots v_{n-1}$. On morphisms, again $et2e(f) = f$.

Theorem 1.11 *The functor e2et is right adjoint to et2e. The adjunction is a reflection, i.e., the counit is a (natural) isomorphism.*

Proof. [Sketch] Let E be an event structure. Then $et2e(e2et(E)) = E$, essentially because the consistency relation derived from $e2et(E)$ recovers that of E. Hence, the counit ε_E is the pair of identities $(1_E, 1_L)$.

The pair $(e2et(E), \varepsilon_E)$ is cofree over E wrt. *et2e*, i.e. for any arrow $(\eta, \lambda) : et2e(T) \to E$ in $\mathbf{E}$, with T an event tree, there is a unique arrow $f : T \to e2et(E)$ in $\mathbf{ET}$ such that $(1_E, 1_L) \circ et2e(f) = (\eta, \lambda)$: it is $f = (\eta, \lambda)$, considering that (η, λ) is a morphism from T to $e2et(E)$ as well; f is uniquely determined since *et2e* is identity on morphisms. □

As a consequence, **E** embeds fully and faithfully into **ET** and is equivalent to the full subcategory of **ET** consisting of those event trees T that are isomorphic to $e2et(et2e(T))$. Event trees that correspond to event structures are characterized as follows. We say that two distinct events e_1, e_2 of an event tree T are *concurrent*, denoted by $e_1\ co_T\ e_2$, if they are consistent and neither $e_1 <_T e_2$ nor $e_2 <_T e_1$, similarly as it is done for event structures.

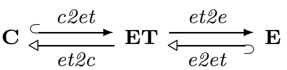
Proposition 1.12 *An event tree T is isomorphic to $e2et(et2e(T))$ iff $Runs(T)$ is* trace-closed*, i.e., satisfies the following condition: if $re_1e_2r' \in Runs(T)$ and $e_1\ co_T\ e_2$ then $re_2e_1r' \in Runs(T)$ as well.*

Proof. Assume T is isomorphic to $e2et(et2e(T))$. The latter is obtained by $e2et$ from some event structure E. In particular, the co relation and runs of $e2et(et2e(T))$ are precisely the same as in E. Hence $Runs(e2et(et2e(T)))$ is trace-closed since $Runs(E)$ is. As a consequence of the isomorphism, $Runs(T)$ is trace-closed as well.

For the opposite direction, assume that $Runs(T)$ is trace-closed. Note that events, causality relation, and labelling in $e2et(et2e(T))$ are the same as in T. Moreover, each run of T is a run of $e2et(et2e(T))$. We only need to show the opposite: each run $r = e_1 \ldots e_n$ of $e2et(et2e(T))$ is a run of T.

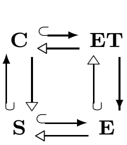
A run of $e2et(et2e(T))$ is also a run of the event structure $et2e(T)$, hence $\{e_1 \ldots e_n\}$ is a consistent set. Hence, events $e_1 \ldots e_n$ appear together in some run of T, i.e., there is $e'_1 \ldots e'_m \in Runs(T)$ such that $\{e_1 \ldots e_n\} = \{e'_{i_1} \ldots e'_{i_n}\}$, for some $1 \leq i_1 < \ldots < i_n \leq m$. Moreover, since r is a run of $et2e(T)$, it is downwards-closed. I.e., if $e'_i < e_j$ then e'_i appears among $e_1 \ldots e_n$, say $e'_i = e_k$; and necessarily $k < j$. Due to this observation, by trace-closure of $Runs(T)$, we can regroup the run $e'_1 \ldots e'_m$ of T so that the events $e_1 \ldots e_n$ form a prefix. Having done this, we can furthermore reorder them as in r. Since the runs of T are prefix-closed, we obtain that $r \in Runs(T)$, which completes the proof. □

The following diagram summarizes the four functors, which relate causal trees and event structures via event trees.



The hooks represent embeddings and the black arrows indicate the direction of left adjoints. Altogether, we have derived a composed adjunction between causal trees and event structures. It is not a coreflection, but induced by a coreflection and a reflection via a larger category. The object component of the right adjoint of this adjunction amounts to the transformation suggested in, e.g., [3]: it 'linearizes' an event structure into a causal tree by forgetting about events.

Integrating the coreflection from synchronization trees **S** to **C**, and the well-known coreflection from **S** to **E** of [6] we obtain:



The diagram can be seen as a decomposition of the coreflection from **S** to **E** into three consecutive adjunctions. It is routine to check that the embeddings and left adjoints commute. The latter implies that right adjoints commute as well, and hence we obtain three different commuting squares:

$$\begin{array}{ccc} \mathbf{C} & \hookrightarrow & \mathbf{ET} \\ \uparrow & \circlearrowleft & \uparrow \\ \mathbf{S} & \hookrightarrow & \mathbf{E} \end{array} \qquad \begin{array}{ccc} \mathbf{C} & \hookrightarrow & \mathbf{ET} \\ \uparrow & \circlearrowleft & \downarrow \\ \mathbf{S} & \hookrightarrow & \mathbf{E} \end{array} \qquad \begin{array}{ccc} \mathbf{C} & \leftarrow & \mathbf{ET} \\ \downarrow & \circlearrowleft & \uparrow \\ \mathbf{S} & \leftarrow & \mathbf{E} \end{array}$$

2 Bisimulation from Open Maps

2.1 **P**-*bisimilarity*

Assume a category of models **M** and a choice of path category $\mathbf{P} \hookrightarrow \mathbf{M}$, a subcategory of **M**. The choice for **P** determines the notion of computation path that will be reflected by **P**-*bisimilarity.*

A morphism $f : X \to Y$ in **M** is **P**-*open* iff it satisfies the following *path-lifting condition.* Whenever, for $m : P \to Q$ a morphism in **P**, a square (1) (c.f. diagrams below) in **M** commutes, *i.e.* $q \circ m = f \circ p$, meaning the path $f \circ p$ in Y can be extended via m to a path q in Y, then there is a morphism p' such that in diagram (2) the two triangles commute, *i.e.* $p' \circ m = p$ and $f \circ p' = q$, meaning the path p can be extended via m to a path p' in X which matches q.

Two objects X_1, X_2 of **M** are **P**-*bisimilar* iff there is a *span* of **P**-open morphisms f_1, f_2 as depicted in diagram (3). For the categories considered in this paper, **P**-bisimilarity is indeed an equivalence relation.

$$(1)\quad \begin{array}{ccc} P & \xrightarrow{p} & X \\ {\scriptstyle m}\downarrow & & \downarrow{\scriptstyle f} \\ Q & \xrightarrow{q} & Y \end{array} \qquad (2)\quad \begin{array}{ccc} P & \xrightarrow{p} & X \\ {\scriptstyle m}\downarrow & \nearrow^{p'} & \downarrow{\scriptstyle f} \\ Q & \xrightarrow{q} & Y \end{array} \qquad (3)\quad \begin{array}{ccccc} & & X & & \\ & {\scriptstyle f_1}\swarrow & & \searrow{\scriptstyle f_2} & \\ X_1 & & & & X_2 \end{array}$$

In the following, we work with respect to a fixed label set L. Given a model category **M**, whose objects have a label set, we restrict our attention to the fibre over L in **M** with respect to the obvious functor projecting the model objects to their label sets. This is exactly the subcategory of **M** with objects those models with label sets L, and morphisms those having the identity on L, 1_L, as label component. We denote the fibre over L in **M** by $\mathbf{M}_L$. Observe that all the adjunctions of Section 1 cut down to the fibres; in particular we have:

$$\mathbf{C}_L \rightleftarrows \mathbf{ET}_L \rightleftarrows \mathbf{E}_L$$

2.2 Hp-b via Open Maps

To obtain a natural instantiation of **P**-bisimilarity for causal trees we single out a path category within $\mathbf{C}_L$. Path objects are naturally taken to be *causal branches*, that is those causal trees which correspond to finite sequences of transitions.

Definition 2.1 *With respect to L, define the* category of causal branches $\mathbf{CBran}_L$ *to be the full subcategory of* $\mathbf{C}_L$ *with objects those* finite *causal trees C satisfying:*

(i) *no forwards branching:* $s \xrightarrow{a} s'$ & $s \xrightarrow{b} s'' \Longrightarrow a = b$ & $s' = s''$.

A morphism $m : P \to Q$ in $\mathbf{CBran}_L$ shows how the causal branch Q can extend the causal branch P: by additional transitions, and/or by increased concurrency. The $\mathbf{CBran}_L$-open morphisms are exactly those which are *zig-zag* (c.f. [4]) and additionally preserve causality; short we say they are *causal zig-zag*.

Definition 2.2 *Let* $f = (\sigma, 1_L) : C \to C'$ *be a morphism in* $\mathbf{C}_L$*. We say* f *is* causal zig-zag *iff it satisfies the following two conditions:*

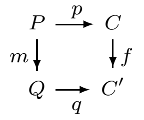
(i) zig-zag: *for all* $s \in S_C$*, if* $\sigma(s) \xrightarrow{a} s'$ *in* C' *then* $s \xrightarrow{a} u$ *in* C *and* $\sigma(u) = s'$*, for some* $u \in S_C$*.*

(ii) causality-preserving: *for all* $t, t' \in Tran_C$, $t <_C t' \Longrightarrow f(t) <_{C'} f(t')$*.*

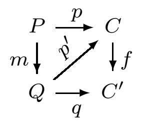
Lemma 2.3 *The* $\mathbf{CBran}_L$*-open morphisms of* $\mathbf{C}_L$ *are exactly those which are causal zig-zag.*

Proof. Let $f = (\sigma, 1_L) : C \to C'$ be a morphism in $\mathbf{C}_L$.

'$\Rightarrow$': Suppose f is $\mathbf{CBran}_L$-open. To prove that f is zig-zag assume $s \in S_C$ and a transition $\sigma(s) \xrightarrow{a} s'$ in C'. Every state in a causal tree is reachable. This implies there must be a run $w = s^{in}_C \xrightarrow{a_1} s_1 \cdots \xrightarrow{a_n} s$ in C, and consequently a run $w' = s^{in}_{C'} \xrightarrow{a_1} \sigma(s_1) \cdots \xrightarrow{a_n} \sigma(s) \xrightarrow{a} s'$ in C'. Let P be the causal branch induced by w, and Q that induced by w'. In $\mathbf{C}_L$ there is a morphism $p : P \to C$ mapping P to w, and a morphism $q : Q \to C'$ mapping Q to w' respectively. Furthermore, there is a unique morphism $m : P \to Q$, which extends P by the a-transition (and possibly by increased concurrency). But altogether this amounts to the following commuting diagram:



Since f is $\mathbf{CBran}_L$-open we obtain a morphism $p' : Q \to C$ such that $p' \circ m = p$ and $f \circ p' = q$:



But this implies there must be $s \xrightarrow{a} u$ in C with $\sigma(u) = s'$ for some $u \in S_C$, as required by the zig-zag condition.

To show that f preserves causality, let $t, t' \in Tran_C$ such that $t' <_C t$. To the contrary assume $f(t') \not<_{C'} f(t)$. There must be a run $w = t_1 \ldots t_i \ldots t_n$ in C with $t_i = t'$ and $t_n = t$, and consequently a run $w' = f(t_1) \ldots f(t_i) \ldots f(t_n)$ in C'. Let P be the causal branch induced by w and Q be that induced by w'. In $\mathbf{C}_L$ there is a morphism $p : P \to C$ mapping P to w, and a morphism $q : Q \to C'$ mapping Q to w' respectively. Further, there is a unique morphism $m : P \to Q$, which at least extends P by requiring the ith transition to be concurrent with the nth. As before, f, p, q, and m amount to a commuting square, and since f is $\mathbf{CBran}_L$-open there must be $p' : Q \to C$ such that $p' \circ m = p$ and $f \circ p' = q$. But since morphisms preserve concurrency this contradicts our assumption $t' <_C t$.

'$\Leftarrow$': Assuming f is zig-zag and causality-preserving, we must show f is $\mathbf{CBran}_L$-open. Suppose two causal branches P and Q in a commuting square:

$$\begin{array}{ccc} P & \xrightarrow{p} & C \\ m\big\downarrow & & \big\downarrow f \\ Q & \xrightarrow[q]{} & C' \end{array}$$

Clearly, p maps P to a run $w = s^{in}_C \stackrel{a_1}{\to} s_1 \ldots \stackrel{a_n}{\to} s_n$ in C, and q maps Q to a possibly extended run $w' = s^{in}_{C'} \stackrel{a_1}{\to} \sigma(s_1) \ldots \stackrel{a_n}{\to} \sigma(s_n) \stackrel{b_1}{\to} s'_1 \ldots \stackrel{b_m}{\to} s'_m$ in C'. Since f is zig-zag there exists a suitable extension of w in C: there is $w_e = s_n \stackrel{b_1}{\to} u_1 \ldots \stackrel{b_m}{\to} u_m$ such that $\sigma(u_i) = s'_i$ for all $i \in [1, m]$. Let $\sigma_{p'} : S_Q \to S_C$ be the unique function that maps Q to ww_e. It is clear that $p' = (\sigma_{p'}, 1_L)$ is a map from Q to C which satisfies $p' \circ m = p$ and $f \circ p' = q$. Further, p' is clearly a morphism from the underlying transition system of Q to that of C. If we additionally achieve that p' preserves concurrency then we can conclude: p' is a morphism as required to establish that f is $\mathbf{CBran}_L$-open.

Let t, $t' \in \mathit{Tran}_Q$ such that $t\ \mathit{Con}_Q\ t'$ (this is indeed always given). Assuming $p'(t') <_C p'(t)$ we want to show $t' <_Q t$. Since f is causality-preserving we obtain $f(p'(t')) <_{C'} f(p'(t))$, which immediately implies $q(t') <_{C'} q(t)$. By q being a morphism and $t\ \mathit{Con}_Q\ t'$ the latter gives us $t' <_Q t$ as required. $\square$

It turns out that $\mathbf{CBran}_L$-bisimilarity coincides with the well-known hp-b. Two systems are hp-bisimilar iff their behaviour can be bisimulated while preserving the causal dependencies between their transitions. Technically, this can be realized by basing hp-b on pairs of *synchronous runs*.

Let C_1, C_2 be causal trees with label sets L, $r_1 = t_1 \ldots t_n \in \mathit{Runs}(C_1)$, and $r_2 = t'_1 \ldots t'_m \in \mathit{Runs}(C_2)$. r_1 and r_2 are *synchronous* iff $n = m$, $\forall i \in [1, n]$, $l_1(t_i) = l_2(t'_i)$, and $\forall i, j \in [1, n]$, $t_i < t_j$ iff $t'_i < t'_j$. We denote the set of synchronous runs of C_1 and C_2 by $\mathit{SRuns}(C_1, C_2)$.

$\mathcal{H} \subseteq \mathit{SRuns}(C_1, C_2)$ is *prefix-closed* iff $(r_1t_1, r_2t_2) \in \mathcal{H}$ implies $(r_1, r_2) \in \mathcal{H}$. We assume hp-bisimulations to be prefix-closed; this restriction has no effect on the induced equivalence.

Definition 2.4 *Let C_1 and C_2 be causal trees with label sets L.*

A history preserving *(*hp-*)* bisimulation *relating C_1 and C_2 is a prefix-closed relation $\mathcal{H} \subseteq \mathit{SRuns}(C_1, C_2)$ that satisfies:*

(i) $(\varepsilon, \varepsilon) \in \mathcal{H}$.

(ii) *If $(r_1, r_2) \in \mathcal{H}$ and $r_1t_1 \in \mathit{Runs}(C_1)$ for some $t_1 \in \mathit{Tran}_1$, then there is $t_2 \in \mathit{Tran}_2$ such that $(r_1t_1, r_2t_2) \in \mathcal{H}$.*

(iii) *Vice versa.*

C_1 and C_2 are hp-bisimilar *iff there exists a hp-bisimulation relating C_1 and C_2.*

Given a morphism $f = (\sigma, 1_L) : C \to C'$ in $\mathbf{C}_L$ we define the image of runs of C in C' inductively by: $f(\varepsilon) = \varepsilon$; $f(r\,(s, a, s')) = f(r)\,(\sigma(s), a, \sigma(s'))$. If f is

$\mathbf{CBran}_L$-open and thus causality-preserving, it is easy to show that a run r of C and its image in C' form a pair of synchronous runs.

Proposition 2.5 *Let $f : C \to C'$ be a $\mathbf{CBran}_L$-open morphism in $\mathbf{C}_L$. For any $r \in \mathit{Runs}(C)$ we have: $(r, f(r)) \in \mathit{SRuns}(C, C')$.*

Proof. Let $f : C \to C'$ be a $\mathbf{CBran}_L$-open morphism in $\mathbf{C}_L$, and suppose $r = t_1 \ldots t_n \in \mathit{Runs}(C)$. Clearly, $f(r) \in \mathit{Runs}(C')$. It is also clear that r and $f(r)$ are of equal length, and that $\forall i \in [1, n]$, $l(t_i) = l'(f(t_i))$ (since the label component of f is 1_L). It remains to show that $\forall i, j \in [1, n]$, $t_i < t_j$ iff $f(t_i) <' f(t_j)$. One direction follows since morphisms preserve concurrency; the other direction is a consequence of Lemma 2.3, which implies that f preserves causality. □

Theorem 2.6 *Two causal trees, with label sets L, are $\mathbf{CBran}_L$-bisimilar iff they are hp-bisimilar.*

Proof. '⇒'. Let $f : C \to C'$ be a $\mathbf{CBran}_L$-open morphism in $\mathbf{C}_L$. We show how from f we obtain a hp-bisimulation relating C and C'. By transitivity of hp-b this will clearly establish the '⇒'-direction. Define $\mathcal{H} = \{(r, f(r)) \mid r \in \mathit{Runs}(C)\}$. By Prop. 2.5 and prefix-closure of $\mathit{Runs}(C)$ it is clear that $\mathcal{H}$ is a prefix-closed subset of $\mathit{SRuns}(C, C')$. To prove that $\mathcal{H}$ is a hp-bisimulation for C and C' we further need to verify that conditions (i)-(iii) of Def. 2.4 are satisfied. (i) is obvious by $\varepsilon \in \mathit{Runs}(C)$. (ii) follows easily from f being a morphism. (iii) can be obtained with the zig-zag condition, which f satisfies by Lemma 2.3.

'⇐'. Let $\mathcal{H}$ be a hp-bisimulation relating two causal trees C_1 and C_2, with label sets L. We observe that $\mathcal{H}$ can be regarded as a causal tree, $C_{\mathcal{H}}$, and that there are two morphisms $f_1 : C_{\mathcal{H}} \to C_1$ and $f_2 : C_{\mathcal{H}} \to C_2$ in $\mathbf{C}_L$.

For $i \in \{1, 2\}$ we define a function $\pi_i : \mathit{SRuns}(C_1, C_2) \to S_i$ by: $\pi_i(\varepsilon, \varepsilon) = s_i^{in}$, and $\pi_i(r_1 t_1, r_2 t_2) = \mathit{tgt}(t_i)$. Further, for $i \in \{1, 2\}$ we define the pair of maps $f_i = (\pi_i, 1_L)$. Given $(r, a, r') \in \mathit{SRuns}(C_1, C_2) \times L \times \mathit{SRuns}(C_1, C_2)$ we write $f_i(r, a, r')$ short for $(\pi_i(r), a, \pi_i(r'))$.

Let $C_{\mathcal{H}} = (\mathcal{H}, (\varepsilon, \varepsilon), L, \mathit{Tran}_{\mathcal{H}}, <_{\mathcal{H}})$ where

$$\mathit{Tran}_{\mathcal{H}} = \{((r_1, r_2), a, (r_1', r_2')) \mid (r_1, r_2), (r_1', r_2') \in \mathcal{H}, r_1 \xrightarrow{a}_1 r_1' \ \&\ r_2 \xrightarrow{a}_2 r_2'\},$$

$$\forall u, u' \in \mathit{Tran}_{\mathcal{H}}.\ u <_{\mathcal{H}} u' \iff f_1(u) <_1 f_1(u') \ \&\ f_2(u) <_2 f_2(u').$$

Below we show that $C_{\mathcal{H}}$ is indeed a causal tree, and that, with π_1, π_2 restricted to $\mathcal{H}$, $f_1 : C_{\mathcal{H}} \to C_1$ and $f_2 : C_{\mathcal{H}} \to C_2$ are indeed morphisms in $\mathbf{C}_L$. Furthermore, we show that f_1 and f_2 are causal zig-zag. But then by Lemma 2.3 there is a span of $\mathbf{CBran}_L$-open morphisms as required.

It is easily seen that $T_{\mathcal{H}} = (\mathcal{H}, (\varepsilon, \varepsilon), L, \mathit{Tran}_{\mathcal{H}})$ is a transition system: $(\varepsilon, \varepsilon) \in \mathcal{H}$ by clause (i) of hp-bisimulation (*c.f.* Def. 2.4). Furthermore, $T_{\mathcal{H}}$ satisfies the axioms of synchronization trees: axioms (ii) and (iii) follow from the definition of $C_{\mathcal{H}}$; to see that (i) holds consider that $\mathcal{H}$ is prefix-closed. $<_{\mathcal{H}}$ is a strict order since $<_1$ and $<_2$ are strict orders.

Then, it only remains to verify that $C_{\mathcal{H}}$ satisfies axiom (i) of Def. 1.1. Let $u = (r_s = (r_s^1, r_s^2), a, r_t)$ and $u' = (r_s', b, r_t')$ be transitions of $C_{\mathcal{H}}$ such that $u' <_{\mathcal{H}} u$.

By definition of $C_{\mathcal{H}}$ we have $f_i(u') <_i f_i(u)$ for $i = 1$, and 2. This means $f_1(u')$ occurs on r_s^1 and $f_2(u')$ on r_s^2. Indeed, they must occur at the same position since u' shows they are matched against each other somewhere. Thus, $r_t' \xrightarrow{v} r_s$ for some $v \in L^*$ as required.

We show that f_1 is a causal zig-zag morphism; the same will follow for f_2 by the symmetric argument. First we check that f_1 satisfies the axioms of transition system morphisms. Axiom (i) is obvious by definition of π_1 and $C_{\mathcal{H}}$'s initial state. Axiom (ii) is straightforward by definition of π_1 and $Tran_{\mathcal{H}}$ when considering that for all $(r_1, r_2) \in \mathcal{H}$, r_1 is a run of C_1. To verify axiom (i) of causal tree morphisms let u, $u' \in Tran_{\mathcal{H}}$ such that $u\ Con_{\mathcal{H}}\ u'$ and $f_1(u') <_1 f_1(u)$. Since the elements of $\mathcal{H}$ are pairs of synchronous runs we also obtain $f_2(u') <_2 f_2(u)$. But this implies $u' <_{\mathcal{H}} u$ by definition of $<_{\mathcal{H}}$.

f_1 is zig-zag follows from $\mathcal{H}$ being a bisimulation. Let $r = (r_1, r_2) \in \mathcal{H}$, and $t_1 = (\pi_1(r), a, s') \in Tran_1$. Clearly, $r_1 t_1 \in Runs(C_1)$. Then by clause (ii) of Def. 2.4 we obtain $t_2 \in Tran_2$ such that $r' = (r_1 t_1, r_2 t_2) \in \mathcal{H}$. But r' is as required to prove the zig-zag condition: clearly, $\pi_1(r') = s'$, and $r \xrightarrow{a} r'$ in $C_{\mathcal{H}}$.

To verify that f_1 is causality-preserving assume $u' <_{\mathcal{H}} u$. But then $f_1(u') <_1 f_1(u)$ by definition of $<_{\mathcal{H}}$. □

2.3 Relating Hp-b and Hhp-b in **ET**

We capture the difference between hp-b and hhp-b by characterizing them within the category **ET**. We carry over hp-b to event structures and event trees. Two event structures E_1 and E_2 are hp-bisimilar iff $et2c(e2et(E_1))$ and $et2c(e2et(E_2))$ are hp-bisimilar; this is consistent with the standard definition. Analogously, it is natural to define: two event trees T_1 and T_2 are hp-bisimilar iff $et2c(T_1)$ and $et2c(T_2)$ are hp-bisimilar.

Consider the following instantiation of **P**-bisimilarity for event trees: as the path category within $\mathbf{ET}_L$ choose the image of $\mathbf{CBran}_L$ under the embedding functor $c2et$; for simplicity, call it $\mathbf{CBran}_L$ as well. $\mathbf{CBran}_L$-bisimilarity in $\mathbf{ET}_L$ characterizes hp-b:

Proposition 2.7 *Two event trees T_1 and T_2 are $\mathbf{CBran}_L$-bisimilar iff they are hp-bisimilar.*

Proof. Let T_1 and T_2 be event trees. It follows from Theorem 2.6 that $et2c(T_1)$ and $et2c(T_2)$ are hp-bisimilar iff they are related by a $\mathbf{CBran}_L$-open span in $\mathbf{C}_L$.

By a general result of [4] for coreflections, f is $\mathbf{CBran}_L$-open in $\mathbf{ET}_L$ iff $et2c(f)$ is $\mathbf{CBran}_L$-open in $\mathbf{C}_L$. Hence, if T_1 and T_2 are related by an open span in $\mathbf{ET}_L$, then $et2c(T_1)$ and $et2c(T_2)$ are related by an open span in $\mathbf{C}_L$, as well.

For the opposite direction, we will use the fact that f is $\mathbf{CBran}_L$-open in $\mathbf{C}_L$ iff $c2et(f)$ is $\mathbf{CBran}_L$-open in $\mathbf{ET}_L$, which was also shown in [4]. Hence, an open span relating $et2c(T_1)$ and $et2c(T_2)$ in $\mathbf{C}_L$ can be transformed by $c2et$ to an open span relating $c2et(et2c(T_1))$ and $c2et(et2c(T_2))$ in $\mathbf{ET}_L$. Now, composing it with the counit components for T_1 and T_2 we get an open span for T_1 and T_2, since counit components are necessarily open (also proved in [4]) and open maps are closed under composition. □

Given a span of morphisms (as depicted in Section 2.1) in $\mathbf{ET}_L$, we say that the span is rooted in $\mathbf{C}_L$ if the root object X is $c2et(C)$ for some causal tree C, and that it is rooted in $\mathbf{E}_L$ if X is $e2et(E)$ for some event structure E. We have:

Proposition 2.8 *Two event trees T_1 and T_2 are $\mathbf{CBran}_L$-bisimilar iff they are related by a $\mathbf{CBran}_L$-open span rooted in $\mathbf{C}_L$.*

Proof. Given an open span relating T_1 and T_2, it is sufficient to compose it with the counit component $\varepsilon_X : c2et(et2c(X)) \to X$, where X is the root object of the span; ε_X is open by a result of [4], hence we get an open span rooted in $\mathbf{C}_L$. □

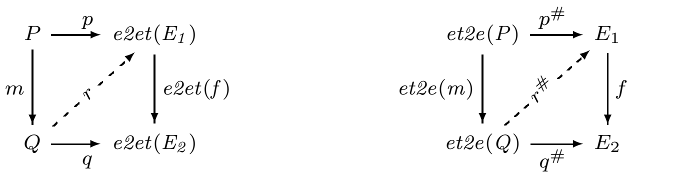
By Prop. 2.7 and 2.8 it follows:

Theorem 2.9 *Two event structures E_1 and E_2 are hp-bisimilar iff $e2et(E_1)$ and $e2et(E_2)$ are related by a $\mathbf{CBran}_L$-open span in $\mathbf{ET}_L$ rooted in $\mathbf{C}_L$.*

Hhp-b is characterized in $\mathbf{E}_L$ as $\mathbf{Pom}_L$-bisimilarity [4], where $\mathbf{Pom}_L$ is the full subcategory of finite *pomsets*, i.e., of finite event structures without conflict (which means all finite subsets of events are consistent). We obtain:

Lemma 2.10 *Let $f : E_1 \to E_2$ be a morphism of event structures. Then f is $\mathbf{Pom}_L$-open in $\mathbf{E}_L$ iff $e2et(f)$ is $\mathbf{CBran}_L$-open in $\mathbf{ET}_L$.*

Proof. A crucial observation is that $\mathbf{Pom}_L$ is an image of $\mathbf{CBran}_L$ via $et2e$, in the following sense: for each T in $\mathbf{CBran}_L$, $et2e(T)$ is a pomset, and further, if f is a morphism in $\mathbf{CBran}_L$ then $et2e(f)$ is in $\mathbf{Pom}_L$; moreover, for any pomsets E_1, E_2 and a morphism $g : E_1 \to E_2$ in $\mathbf{Pom}_L$, there exist objects T_1, T_2 and a morphism $f : T_1 \to T_2$ in $\mathbf{CBran}_L$ such that $E_1 = et2e(T_1)$, $E_2 = et2e(T_2)$, and $g = et2e(f)$.
Hence, two considered openness conditions involve commuting squares of the following related forms, in $\mathbf{ET}_L$ and $\mathbf{E}_L$, respectively:

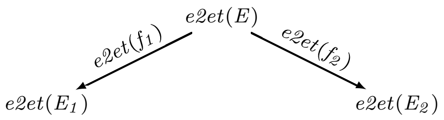


Morphism $m : P \to Q$ is in $\mathbf{CBran}_L$, and $\_^{\#}$ denotes a bijective correspondence of hom-sets, $\mathbf{ET}_L(T, e2et(E)) \longleftrightarrow \mathbf{E}_L(et2e(T), E)$, given by the adjunction between fibres $\mathbf{ET}_L$ and $\mathbf{E}_L$. By the general adjunction law, $(e2et(f) \circ r)^{\#} = f \circ r^{\#}$, hence the bottom-right triangle commutes in the left-hand side diagram iff the corresponding triangle commutes in the right-hand one. Furthermore, by the same law it follows that $(r \circ m)^{\#} = r^{\#} \circ et2e(m)$, hence also the upper-left triangle commutes in the left-hand diagram iff the corresponding triangle commutes in the right-hand one. Finally, combining the two mentioned equations, namely $(e2et(f) \circ p)^{\#} = f \circ p^{\#}$ and $(q \circ m)^{\#} = q^{\#} \circ et2e(m)$, we verify that the left-hand square commutes iff the other square does. As a conclusion, f is $\mathbf{Pom}_L$-open iff $e2et(f)$ is $\mathbf{CBran}_L$-open, which concludes the proof. □

Theorem 2.11 *Two event structures E_1 and E_2 are hhp-bisimilar iff $e2et(E_1)$ and $e2et(E_2)$ are related by a $\mathbf{CBran}_L$-open span in $\mathbf{ET}_L$ rooted in $\mathbf{E}_L$.*

Proof. As shown in [4], two event structures E_1 and E_2 are hhp-bisimilar iff they are $\mathbf{Pom}_L$-bisimilar in $\mathbf{E}_L$. An open span relating E_1 and E_2 can be transformed via *e2et* to a span in $\mathbf{ET}_L$, which is $\mathbf{CBran}_L$-open by Lemma 2.10. Apparently, this span is rooted in $\mathbf{E}_L$.

For the opposite direction, assume that $e2et(E_1)$ and $e2et(E_2)$ are related by a $\mathbf{CBran}_L$-open span, with the root object $et2e(E)$ for some event structure E. Functor *e2et* is full and faithful; hence the two arrows of the span are necessarily obtained from some morphisms of event structures via *e2et*:



such that $f_1 : E \to E_1$ and $f_2 : E \to E_2$. Hence, we get a span relating E_1 and E_2 in $\mathbf{E}_L$, which is $\mathbf{Pom}_L$-open by Lemma 2.10. □

Theorems 2.9 and 2.11 indicate that $\mathbf{C}$ is the proper choice of model for hp-b while $\mathbf{E}$ is the natural choice for hhp-b.

3 Conclusions

Altogether we have advocated causality as a non-embedding but adjoining concept to true-concurrency. (We prefer the admittedly biased term 'true-concurrency' to 'independence' here since (in)dependence can be captured without a notion of event in the style of causal trees, just as well.) We summarize:

(i) Causality models are more basic than truly-concurrent models in that they capture causality without a notion of event. On the other hand, they are more expressive than the latter in that their possible runs can be freely specified in terms of a tree; in contrast, truly-concurrent models and their sets of runs adhere to certain axioms that express characteristics of independent events.

(ii) Hp-b turns out to have a straightforward open map characterization when we take causal trees to be the model category. Our results motivate that hp-b is *the* bisimilarity for causality while hhp-b remains *the* bisimilarity for true-concurrency.

Our work should be compared to [1], which relates causal trees to prioritized event structures. It would also be interesting to confirm our results with respect to models that keep the cyclic structure. A type of history-dependent automata, called *causal automata*, should be examined in this context.

Our investigation has led us to the new model of event trees. We are not keen on advertising yet another model for concurrency but event trees do arise in practice: given a truly-concurrent system, assume we restrict our attention to a subset of its runs that is not necessarily trace-closed. This is exactly what we do during a partial order reduction; indeed it is the intention here to lose trace-closure.

We are working on a characterization of those event structures E which correspond to causal trees in that $E = et2e(c2et(C))$ for some causal tree C. We expect that such event structures are optimal for partial order reduction.

References

[1] C. Bodei. Some concurrency models in a categorical framework. In *ICTCS'98*, pages 180–191. World Scientific, 1998.

[2] Allan Cheng and Mogens Nielsen. Observing behaviour categorically. In *FST&TCS'95*, volume 1026 of *LNCS*, pages 263–278, 1995.

[3] Philippe Darondeau and Pierpaolo Degano. Causal trees: interleaving + causality. In *Semantics of systems of concurrent processes*, volume 469 of *LNCS*, pages 239–255, 1990.

[4] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, 1996.

[5] Marco Pistore. *History Dependent Automata*. PhD thesis, University of Pisa, 1999.

[6] Glynn Winskel and Mogens Nielsen. Models for concurrency. In *Handbook of logic in computer science, Vol. 4*, pages 1–148. Oxford Univ. Press, New York, 1995.