Revisiting Causality, Coalgebraically

Roberto Bruni · Ugo Montanari · Matteo Sammartino

Abstract In this paper we recast the classical Darondeau-Degano's causal semantics of concurrency in a coalgebraic setting, where we derive a compact model. Our construction is inspired by the one of Montanari and Pistore yielding *causal automata*, but we show that it is instance of an existing categorical framework for modeling the semantics of nominal calculi, whose relevance is further demonstrated. The key idea is to represent events as names, and the occurrence of a new event as name generation. We model causal semantics as a coalgebra over a presheaf, along the lines of the Fiore-Turi approach to the semantics of nominal calculi. More specifically, we take a suitable category of finite posets, representing causal relations over events, and we equip it with an endofunctor that allocates new events and relates them to their causes. Presheaves over this category express the relationship between processes and causal relations among the processes' events. We use the allocation operator to define a category of well-behaved coalgebras: it models the occurrence of a new event along each transition. Then we turn the causal transition relation into a coalgebra in this category, where labels only exhibit maximal events with respect to the source states' poset, and we show that its bisimilarity is essentially Darondeau-Degano's strong causal bisimilarity. This coalgebra is still infinite-state, but we exploit the equivalence between coalgebras over a class of presheves and *History Dependent automata* to derive a compact representation, where states only retain the poset of the most recent events for each atomic subprocess, and are isomorphic up to order-preserving permutations. Remarkably, this reduction of states is automatically performed along the equivalence.

Dipartimento di Informatica, Università di Pisa, Largo Bruno Pontecorvo 3, 56127 Pisa, Italy ·
Roberto Bruni
E-mail: bruni@di.unipi.it

Ugo Montanari
E-mail: ugo@di.unipi.it

Matteo Sammartino (Corresponding author)
E-mail: sammarti@di.unipi.it
Tel.:+39 0502213117

1 Introduction

Causal trees [9] are a variant of Milner's synchronization trees with enriched action labels, specifying the set of causes for each edge. They can be used to provide process calculi with a semantics that makes dependencies among actions explicit. In [9] the authors introduce a technique for deriving a causal semantics from a labelled one. The basic idea is to explicitly decorate each atomic subprocess with a set of *causes*. When one subprocess performs an action, or two subprocesses synchronize, a new event is generated and the causes of the involved processes are shown in the label, together with the original action. These causes, updated with the new event, are then assigned to the continuations of the subprocess(es).

The key issue is that causal semantics is usually infinite state, because states keep track of the whole history of events, which is enlarged at each transition. Moreover, observations keep growing in size, while minimization would require a more succinct form of observation. In this paper we aim at providing a technique for obtaining equivalent, but more compact models for the causal semantics of concurrency. Our approach has the following two steps:

(i) **Reduction of labels.** Each causal process is equipped with a partial order over its events, representing causal relations determined by past transitions. Then events that are not maximal according to the ordering, i.e. all but the most recent ones, are removed from labels.

(ii) **Reduction of states.** Only *immediate causes* of atomic subprocesses are kept, i.e. events that are maximal in the ordering w.r.t. at least one of the subprocesses. Intuitively, we keep causes for the most recent transitions. Then states are identified up to a suitable order-preserving notion of isomorphism, and transitions are enriched with maps that keep track of the original identity of events.

Our main source of inspiration is Montanari and Pistore [14], where the issue of providing a minimization procedure for Petri Nets with a causal semantics is tackled by introducing *causal automata*. However, an ordinary LTS is eventually recovered by computing "active names" and minimization is performed with respect to ordinary LTS bisimulation. This is an ad-hoc technique for special classes of Petri nets: in general, the computation of active names is not decidable.

Following Montanari and Pistore, we first give a set-theoretical construction that performs (i) and (ii) on causal transition systems. It is quite involved, due to its very concrete nature. Then we recast it in a categorical setting, where it becomes much more natural and simple. We will use: (a) *coalgebras* [16,1] over a presheaf category to represent causal transition systems; (b) *History dependent automata* (HD-automata) [15,8] to achieve, in lots of practical cases, a concrete model with a finite number of states, suitable for verification. The choice of (a) and (b) is due to their intimate relationship: when they are defined over particular categories, the latter can be automatically derived from the former through a general categorical construction which has had, and possibly will have, several other similar instances. We now introduce our categorical framework.

1.1 A coalgebra for causality

Colgebras are convenient models of dynamic systems. Their theory is rich and well-developed, and many kinds of systems have been characterized in this setting. Coalgebras are also of practical interest: minimization procedures such as *partition refinement* [13], which are essential for finite-state verification, have been formulated in coalgebraic terms (see e.g. [2]). This further motivates the coalgebraic framework: algorithms implemented at this level of abstraction can be easily instantiated to many classes of systems.

Our coalgebraic model of causality is based on the idea of representing events as *names*, that are atomic entities characterized only by their identity, and the occurrence of a new event as *name generation*. This allows us to construct a coalgebra where states are equipped with nominal structures, namely causal relations between events, and event generation is explicit, along the lines of [11]. The key idea is to define coalgebras over *presheaves*, that are functors from a certain *index category* **C** to **Set**, the category of sets and functions. Presheaves formalize the association between collections of names, seen as objects of **C**, and sets of processes within **Set**. Fresh name generation can be formalized as an endofunctor on **C**, that is lifted to presheaves and used in the definition of coalgebras.

We take as index category for presheaves a suitable category of *partially ordered finite sets*, representing causal relations between events. This category provides us with the needed structure to model operations over causal relations. In fact, we use colimits to implement a *well-behaved functorial* model of event generation, which augments a given poset with fresh events and causal relations to their causes. Our definition ensures that its lifting to preheaves, when used to define coalgebras, yields a category of coalgebras with a final object and a final semantics in agreement with coalgebraic bisimilarity. This is essential for a correct notion of minimal model. Then, we define a presheaf of processes, yielding, for each poset, the set of causal processes whose causes are "compatible" with that poset. We construct a *causal coalgebra* by translating the LTS produced by the reduction step (i). The important result is that coalgebraic bisimulations on this coalgebra are equivalent to a class of (strong Darondeau-Degano) causal bisimulations. In particular, the equivalence holds for ordinary and coalgebraic bisimilarity.

1.2 An efficient operational model: HD-automata

The state space explosion issue still exists in the causal coalgebra, because the poset of a causal process keeps growing along transitions. However, if the presheaf of states is "well-behaved", according to [7], it is always possible to recover the *support* of a causal process, that is the minimal poset including all and only events that appear in the process. This is the key condition for the equivalence between presheaf-based coalgebras and *History Dependent (HD) automata*.

HD-automata are coalgebras with states in *named-sets* [8], that are sets whose elements are equipped with a symmetry group over finite collections of names. They have two main features:

- a single state can represent the whole orbit of its symmetry;
- the names of each state are *local*, related to those of other states via suitable mappings.

Both are important for applying finite state methods, such as minimization and model-checking, to nominal calculi. In particular, the latter point captures *deallocation*: maps between states can discard unused names and "compact" remaining ones, much like *garbage collectors* do for memory locations. A minimization procedure for HD-automata for the (finite-control) π-calculus have been shown and implemented in [10].

Interestingly, we are able to define the presheaf of processes in a way that the computation of the support discards all but the immediate causes. Therefore, the aforementioned equivalence implements the reduction step (ii) and gives an HD-automaton over a named set of *minimal* causal processes, equipped with symmetry groups over their posets. This is similar to Montanari and Pistore's causal automata, but our category-theoretic version allows for the further identification of states up to symmetries, as a state can be bisimilar to itself via an order-preserving permutation of its poset. Symmetries are not present in causal automata.
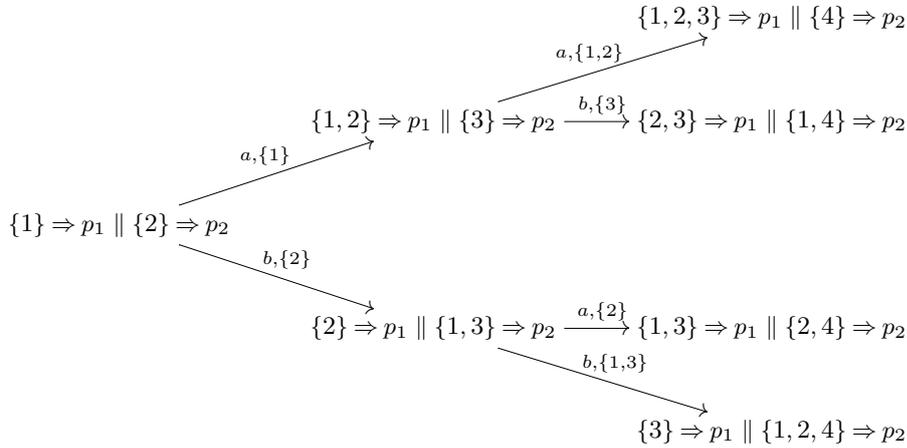
1.3 Illustrative example

We give an example of how the reduction steps (i) and (ii) can be achieved. Consider two atomic processes p_1 and p_2 that have the following transitions

$$p_1 \xrightarrow{a} p_1 \qquad p_2 \xrightarrow{b} p_2 \ .$$

We assign cause 1 to p_1 and 2 to p_2, written $\{1\} \Rightarrow p_1$ and $\{2\} \Rightarrow p_2$. According to the Darondeau-Degano LTS, these two causal processes separately have the following transitions

$$\{1\} \Rightarrow p_1 \xrightarrow{a,\{1\}} \{1,2\} \Rightarrow p_1 \qquad \{2\} \Rightarrow p_2 \xrightarrow{b,\{2\}} p_2 \Rightarrow \{1,3\}$$

where, in the target process, cause 1 denotes the last event and previous cases have been incremented by one. Their state-space is infinite, and so is that of their parallel composition

$$\{1\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$$

with transitions

$$\xrightarrow{a,\{1\}} \{1,2\} \Rightarrow p_1 \parallel \{3\} \Rightarrow p_2 \xrightarrow{a,\{1,2\}} \{1,2,3\} \Rightarrow p_1 \parallel \{4\} \Rightarrow p_2$$

$$\xrightarrow{b,\{3\}} \{2,3\} \Rightarrow p_1 \parallel \{1,4\} \Rightarrow p_2$$

$$\xrightarrow{b,\{2\}} \{2\} \Rightarrow p_1 \parallel \{1,3\} \Rightarrow p_2 \xrightarrow{a,\{2\}} \{1,3\} \Rightarrow p_1 \parallel \{2,4\} \Rightarrow p_2$$

$$\xrightarrow{b,\{1,3\}} \{3\} \Rightarrow p_1 \parallel \{1,2,4\} \Rightarrow p_2$$

We get a more efficient representation by explicitly associating to each process the causal relations determined by its transitions, in the form of a poset over

causes, and then letting labels contain only causes that are maximal elements of this poset. For instance we can associate the discrete poset $\{1,2\}$ to the leftmost process, written

$$\{1,2\} \rhd \{1\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2 \tag{1}$$

Since 1 and 2 are both maximal elements, the labels for the leftmost transitions are kept, and their continuations become

$$O_1 \rhd \{1,2\} \Rightarrow p_1 \parallel \{3\} \Rightarrow p_2 \qquad O_2 \rhd \{2\} \Rightarrow p_1 \parallel \{1,3\} \Rightarrow p_2 \tag{2}$$

where O_1 and O_2 are posets over $\{1,2,3\}$ such that $2 \prec_{O_1} 1$ and $3 \prec_{O_2} 1$. Now, since 1 is maximal in both cases, outgoing labels from these processes can be reduced as follows:

$$a, \{1,2\} \longmapsto a, \{1\}$$
$$b, \{1,3\} \longmapsto b, \{1\}$$

We got more compact labels, but the state space is still infinite. To solve this problem, we also reduce processes by only keeping immediate causes, that are causes that are maximal with respect to at least one of the atomic subprocesses. Under this reduction, (2) become

$$\{1,3\} \rhd \{1\} \Rightarrow p_1 \parallel \{3\} \Rightarrow p_2 \qquad \{1,2\} \rhd \{2\} \Rightarrow p_1 \parallel \{1\} \Rightarrow p_2 \tag{3}$$

This transformation is not enough, as the LTS is still infinite-state. The key observation here is that processes (3) are isomorphic, and so are their causal trees. Indeed, all the processes in the figure above become isomorphic after the reduction. Therefore we can replace all of them with a canonical representative for their isomorphism class. For instance, under the isomorphisms ϕ_1, ϕ_2 defined as follows

$$\phi_1(1) = 1 \quad \phi_1(3) = 2 \quad \phi_2(1) = 2 \quad \phi_2(2) = 1$$

processes (3) become the process (1), and this also affects their transitions. We can apply a similar transformation to all the processes in the figure above, getting

$$a,\{1\}$$
$$\overset{\frown}{h_1} \searrow$$
$$\{1,2\} \rhd \{1\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$$
$$\overset{h_2}{\smile} \nearrow$$
$$b,\{2\}$$

The information about the original transitions is encoded in the *history maps* $h_1 = \phi_1^{-1}$ and $h_2 = \phi_2^{-2}$: they translate events of the unique continuation to those of the original continuations.

We gave a set-theoretic example for simplicity. Even if the result is already minimal in this case, and in fact essentially equivalent to Montanari and Pistore construction, the category-theoretic treatment will yield more compact models in some cases, thanks to the presence of symmetry groups for each state.

2 Background

2.1 Functor categories

Definition 1 (Functor category) Let $\mathbf{C}$ and $\mathbf{D}$ be two categories. The *functor category* $\mathbf{D}^{\mathbf{C}}$ has functors $\mathbf{C} \to \mathbf{D}$ as objects and natural transformations between them as morphisms.

Functors from any category $\mathbf{C}$ to $\mathbf{Set}$ are called (covariant) *presheaves*. A presheaf P can be intuitively seen as a family of sets indexed over the objects of $\mathbf{C}$ plus, for each $\sigma\colon c \to c'$, an action of σ on Pc, which we write

$$p[\sigma]_P := P\sigma(p) \qquad (p \in Pc) \ ,$$

omitting the subscript P in $[\sigma]_P$ when clear from the context. This notation intentionally resembles the application of a renaming σ to a process p, namely $p\sigma$: it will, in fact, have this meaning in later chapters. The set $\int P$ of *elements* of a presheaf P is

$$\int P := \sum_{c \in |\mathbf{C}|} Pc$$

and we denote by $c \rhd p$ a pair belonging to $\int P$. Presheaf categories have the following nice property.

Property 1 For any $\mathbf{C}$, $\mathbf{Set}^{\mathbf{C}}$ has all limits and colimits, both computed pointwise.

2.2 Coalgebras

The *behavior* of systems can be modeled in a categorical setting through *coalgebras* [16,1]. Given a *behavioral endofunctor* $B\colon \mathbf{C} \to \mathbf{C}$, describing the "shape" of a class of systems, we have a corresponding category of coalgebras.

Definition 2 (B-Coalg) The category B-**Coalg** is defined as follows: objects are B-*coalgebras*, i.e. pairs (X, h) of an object $X \in |\mathbf{C}|$, called *carrier*, and a morphism $h\colon X \to BX$, called *structure map*; B-*coalgebra homomorphisms* $f\colon (X, h) \to (Y, g)$ are morphisms $f\colon X \to Y$ in $\mathbf{C}$ making the following diagram commute

$$
\begin{array}{ccc}
X & \xrightarrow{\ h\ } & BX \\
{\scriptstyle f}\big\downarrow & & \big\downarrow{\scriptstyle Bf} \\
Y & \xrightarrow{\ g\ } & BY
\end{array}
$$

For instance, consider the functor

$$B_{clts} := \mathcal{P}_f(L \times -)$$

where $\mathcal{P}_f\colon \mathbf{Set} \to \mathbf{Set}$ is the *countable powerset functor*, defined on a set A and on a function $f\colon A \to A'$ as follows

$$\mathcal{P}_f A := \{B \subseteq A \mid B \text{ countable}\} \qquad \mathcal{P}_f f(B) := \{f(b) \mid b \in B\}$$

B_{clts}-coalgebras are *finitely-branching labelled transition systems*, with labels in L, and their homomorphisms are functions that preserve and reflect transitions.

In this category many notions of behavioral equivalence can be defined [18]. We adopt the following one.

Definition 3 (B-bisimulation) Given a B-coalgebra (X, h), a *B-bisimulation* on it is an object R of $\mathbf{C}$ such that $R \hookrightarrow X \times X$ and there is $r \colon R \to BR$ making the following diagram commute

$$
\begin{array}{ccc}
X & \xleftarrow{\ \pi_1\ } R \xrightarrow{\ \pi_2\ } & X \\
{\scriptstyle h}\downarrow & \ \ {\scriptstyle r}\downarrow & \downarrow{\scriptstyle h} \\
BX & \xleftarrow[B\pi_1]{} BR \xrightarrow[B\pi_2]{} & BX
\end{array}
$$

A B_{clts}-bisimulation R on a B_{clts}-coalgebra is an ordinary bisimulation on the corresponding transition system: the diagram means that $x, x' \in X$ such that $(x, x') \in R$ must be able to perform transitions with the same label l, and these transitions are represented as a single one $(x, x') \xrightarrow{l} (y, y')$ in (R, r); by definition of B_{clts}, y and y' must again be related by R.

An important property of categories of coalgebras is the existence of the terminal object; the unique morphism from each coalgebra to it assigns to each state its abstract semantics. If B preserves weak pullbacks, i.e. pullbacks such that the mediating morphism need not be unique, then B-bisimilarity and the terminal coalgebra agree.

The requirement for the existence of the final coalgebra is that B is an *accessible* functor on a *locally presentable* category (see [3, 19, 1] for details). We just recall the main results. Given a regular cardinal λ, a category $\mathbf{C}$ is λ-*filtered* if each diagram of cardinality less than λ is the base of a cocone in $\mathbf{C}$. λ-filtered categories generalize the notion of directed preorders, that are sets such that every finite subset has an upper bound. For any category $\mathbf{D}$, a λ-*filtered colimit* in $\mathbf{D}$ is the colimit of a diagram of shape $\mathbf{C}$, i.e. a functor $\mathbf{C} \to \mathbf{D}$, such that $\mathbf{C}$ is a λ-filtered category.

Definition 4 (Locally λ-presentable category) An object c of a category $\mathbf{C}$ is λ-presentable if the functor $\mathrm{Hom}_{\mathbf{C}}(c, -) \colon \mathbf{C} \to \mathbf{Set}$ preserves λ-filtered colimits. A category $\mathbf{C}$ is locally λ-presentable if it has all colimits and there is a set of λ-presentable objects $X \subseteq |\mathbf{C}|$ such that every object is a λ-filtered colimit of objects from X. We say that a category is locally presentable if it is locally λ-presentable for some λ.

For instance, locally λ-presentable objects in $\mathbf{Set}$ are precisely the finite sets with cardinality less than λ. $\mathbf{Set}$ is locally ω-presentable: every set is the ω-filtered colimit of its finite subsets and the whole $\mathbf{Set}$ is generated by the set containing one finite set of cardinality n for all $n \in \mathbb{N}$.

For functor categories we have the following.

Proposition 1 *For each locally λ-presentable category $\mathbf{C}$ and small category $\mathbf{D}$, the functor category $\mathbf{C}^{\mathbf{D}}$ is λ-presentable.*

In particular, since $\mathbf{Set}$ is ω-presentable, we have that the presheaf category $\mathbf{Set}^{\mathbf{D}}$ is ω-presentable as well.

Definition 5 (Accessible functor) Let $\mathbf{C}$ and $\mathbf{D}$ be locally λ-presentable categories. A functor $F\colon \mathbf{C} \to \mathbf{D}$ is λ-*accessible* if it preserves λ-filtered colimits. We just say F is accessible if it is λ-accessible for some λ.

Products, coproducts and composition of accessible functors are accessible as well.

2.3 Coalgebras over presheaves

Coalgebras for functors $B\colon \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}^{\mathbf{C}}$ have some additional structure: they are pairs (P,ρ) of a presheaf $P\colon \mathbf{C} \to \mathbf{Set}$ and a natural transformation $\rho\colon P \to BP$. The naturality of ρ imposes a constraint on behavior

$$
\begin{array}{ccc}
c & p \in Pc \xmapsto{\ \ \rho_c\ \ } beh(p) \\[2mm]
f \downarrow & [f]_P \uparrow \qquad\qquad\qquad \uparrow [f]_{BP} \\[2mm]
c' & p[f]_P \in P(c') \xmapsto[\ \rho_{c'}\ ]{} beh(p)[\sigma]_{BP}
\end{array}
$$

Intuitively, this diagram means that, if we take a state, apply a function to it and then compute its behavior, we should get the same thing as first computing the behavior and then applying the function to it. In other words, behavior must be *preserved* and *reflected* by the index category morphisms.

Also bisimulations have more structure. A B-bisimulation R is a presheaf in $\mathbf{Set}^{\mathbf{C}}$ and all the legs of the bisimulation diagram in Definition 3 are natural transformations. In particular, the naturality of projections implies that, given $(p,q) \in Rc$ and $f\colon c \to c'$ in $\mathbf{C}$, $(p[f], q[f]) \in R(c')$, i.e. B-bisimulations are *closed under the index category morphisms*.

3 Causal processes

We recall the Darondeau-Degano causal semantics of concurrency. We denote atomic processes by $p, q, \ldots$ and their parallel composition by $t, t', \ldots$. Let Act be a set of actions such that, for each $a \in Act$, there is also $\bar{a} \in Act$. We assume a set of basic transitions for atomic processes

$$
\Delta = \{p \xrightarrow{a} t \mid a \in Act\}
$$

such that the subset $\Delta_p = \{p \xrightarrow{a} t \in \Delta\}$ is finite, for all p.

Causal processes are process terms whose constants are decorated with sets of positive natural numbers representing their causes, for instance

$$
K_1 \Rightarrow p_1 \parallel \cdots \parallel K_n \Rightarrow p_n
$$

where $K_1, \ldots, K_n \subseteq \mathbb{N}^+$. We will use $k, k', \ldots$ to denote these processes. We assume that atomic processes have an initial cause, i.e. for all p there is a unique e such that $\{e\} \Rightarrow p$. We write $K \Rightarrow t$ for the causal process obtained by giving causes K to every atomic subprocess in t and $\mathscr{K}(k)$ for the set of all causes appearing in k. For example, given $k = \{1,3\} \Rightarrow p_1 \parallel \{1,2\} \Rightarrow p_2$, we have $\mathscr{K}(k) = \{1,2,3\}$. The following operators are needed for the LTS:

$$\frac{p \xrightarrow{a} t \in \Delta}{K \Rightarrow p \xrightarrow{a,K}_{\text{DD}} \{1\} \cup \delta(K) \Rightarrow t} \qquad\qquad \frac{t_1 \xrightarrow{l} t_1'}{t_1 \parallel t_2 \xrightarrow{l}_{\text{DD}} t_1' \parallel \delta(t_2)}$$

$$\frac{t_1 \xrightarrow{a,K_1}_{\text{DD}} t_1' \qquad t_2 \xrightarrow{\overline{a},K_2}_{\text{DD}} t_2'}{t_1 \parallel t_2 \xrightarrow{\tau,K_1 \cup K_2}_{\text{DD}} \eta(\delta(K_2),t_1') \parallel \eta(\delta(K_1),t_2')} \qquad \frac{t_2 \xrightarrow{l} t_2'}{t_1 \parallel t_2 \xrightarrow{l}_{\text{DD}} \delta(t_1) \parallel t_2'}$$

Fig. 1 Inference rules for the Darondeau-Degano LTS.

- $\delta(K)$ shifts all the causes in K by one, in order to "make room" for the new event 1; we let $\delta(K \Rightarrow p) = \delta(K) \Rightarrow p$
- $\eta(K_1, K_2)$ joins K_1 and K_2 only if $1 \in K_2$, otherwise returns K_2; we let $\eta(K_1, K_2 \Rightarrow p) = \eta(K_1, K_2) \Rightarrow p$.

These operators are assumed to distribute over parallel composition, i.e. $\delta(k_1 \parallel k_2) = \delta(k_1) \parallel \delta(k_2)$ and $\eta(K, k_1 \parallel k_2) = \eta(K, k_1) \parallel \eta(K, k_2)$.

Definition 6 (Darondeau-Degano LTS) The *Darondeau-Degano LTS* (LTS$_{\text{DD}}$) is the smallest one generated by the rules in Figure 1.

Definition 7 (Causal bisimulation) *Causal bisimulations* are ordinary bisimulations on LTS$_{\text{DD}}$. The greatest one, namely *causal bisimilarity*, is denoted by $\sim_{\text{DD}}$.

4 Two partial order LTSs

In this section we present two refinements of LTS$_{\text{DD}}$. The goal is obtaining a compact LTS, where labels are more succinct and states only keep track of the most recent events. The crucial idea is equipping causal processes with a poset that keeps track of causal relations determined by transitions. Given a poset O, in the following we write $|O|$ for the underlying set of O, and $\prec_O$ to denote the relation of O in infix notation.

Definition 8 (Poset-indexed causal processes) A *poset-indexed causal process*, P-process in short, is a pair

$$O \triangleright k$$

of a causal process k and a poset O such that $\mathscr{K}(k) \subseteq O$ and, for all $K \Rightarrow p$ in k

$$\forall e, e' \in |O| : e \in K \wedge e' \prec_O e \implies e' \in K \ . \tag{4}$$

Condition (4) requires the set of causes of each atomic subprocess to contain the whole "history" of each event, as described by O. Nevertheless, O may contain events that are unrelated to or caused by those of $\mathscr{K}(k)$, but are not among them.

The poset of a P-process can be enlarged by adding causes for existing events, but this requires "closing" the process under the new causes in order to satisfy (4). Given P-process $O \triangleright k$ and $O' \supseteq O$, we define a *closure operator* $k{\downarrow}_{O'}$ as follows

$$(K \Rightarrow p){\downarrow}_{O'} = \bigcup_{e \in K} \{ e' \in |O'| \mid e' \prec_{O'} e \} \Rightarrow p$$

distributing over parallel composition. Then it can be easily checked that $O' \triangleright k{\downarrow}_{O'}$ is a proper P-process.

4.1 Poset-indexed LTS

We introduce the first LTS, namely the *poset-indexed LTS*, whose states are P-processes, suitably decorated by posets, but transitions only show *maximal* events, according to the poset of the source process.

Definition 9 (Poset-indexed LTS) The *poset-indexed LTS* ($\mathrm{LTS_{PO}}$) is generated from the $\mathrm{LTS_{DD}}$ by the following rule

$$\frac{k \xrightarrow{a,K}_{\mathsf{DD}} k' \qquad M = max_O(K)}{O \rhd k \xrightarrow{a,M}_{\mathsf{PO}} \delta_M(O) \rhd k'}$$

where

- $max_O(K)$ is the subset of K containing only maximal causes according to O;
- $\delta_K(O)$, for any set of causes K, is the transitive and reflexive closure of

$$\{(n+1, m+1) \mid (n,m) \in |O|\} \cup K \times \{1\} \quad .$$

The key operation here is $\delta_M(O)$: it acts similarly to $\{1\} \cup \delta(O)$ defined in section 3, but, besides adding a new event, it also establishes connections with its causes. One can easily see that $\delta_M(O) \rhd k'$ is a proper P-process: all the causes of the only new event in k' are already in k', by construction (see Figure 1).

The behavioral equivalence for $\mathrm{LTS_{PO}}$ is the following.

Definition 10 (Poset-indexed causal bisimulation) Poset-indexed causal bisimulations are families of binary relations $\{R_O\}$, where O is a poset on a finite subset of $\mathbb{N}$, such that, for each $(O \rhd k, O \rhd k') \in R_O$, if $O \rhd k \xrightarrow{a,K}_{\mathsf{PO}} O' \rhd k''$ then there is $O \rhd k' \xrightarrow{a,K}_{\mathsf{PO}} O' \rhd k'''$ with $(O' \rhd k'', O' \rhd k''') \in R_{O'}$. The greatest poset-indexed causal bisimulation is denoted by $\sim_{\mathsf{PO}}$.

Proposition 2 *Given $O \rhd k$ and $O \rhd k'$, $O \rhd k \sim_{\mathsf{PO}} O \rhd k'$ if and only if $k \sim_{\mathsf{DD}} k'$.*

We list some closure properties, which will be important in the following. We say that a monotone function $\sigma \colon O \to O'$ is *poset-reflecting* whenever

$$\forall x, y \in |O| : \sigma(x) \prec_{O'} \sigma(y) \implies x \prec_O y \quad . \tag{5}$$

That is, it does not introduce spurious causal relations between (images of) existing events. We introduce the following notation: given a process k and a set of events K, $k\sigma$ and $K\sigma$ denotes the application of σ to each event in k and K, respectively.

Proposition 3 *Transitions of $\mathrm{LTS_{PO}}$ are* preserved *and* reflected *by injective poset-reflecting functions $\sigma \colon O \to O'$, that is:*

(i) *If $O \rhd k \xrightarrow{a,K}_{\mathsf{PO}} \delta_K(O) \rhd k'$ then $O' \rhd (k\sigma)\!\downarrow_{O'} \xrightarrow{a,K\sigma}_{\mathsf{PO}} \delta_{K\sigma}(O') \rhd (k'\sigma^+)\!\downarrow_{\delta_{K\sigma}(O')}$ (preservation);*

(ii) *If $O' \rhd (k\sigma)\!\downarrow_{O'} \xrightarrow{a,K'}_{\mathsf{PO}} \delta_{K'}(O') \rhd k'$ then there are K and k'' such that $K\sigma = K'$, $(k''\sigma^+)\!\downarrow_{\delta_{K'}(O')} = k'$ and $O \rhd k \xrightarrow{a,K}_{\mathsf{PO}} \delta_K(O) \rhd k''$ (reflection);*

where σ^+ is an injective poset-reflecting function $\delta_K(O) \to \delta_{K\sigma}(O')$ given by

$$\sigma^+(n) = \begin{cases} 1 & n = 1 \\ \sigma(n-1) + 1 & otherwise \end{cases}$$

The definition of preservation and reflection are quite involved, due to the presence of event allocation and the necessity of applying the closure operator to compute proper continuations. In particular, we need to introduce σ^+, a version of σ that takes into account the shift of events along transitions. We will see that the categorical counterparts of these properties will be remarkably simpler.

Example 1 We motivate the requirement of poset-reflection by showing that transitions of LTS_{PO} are not reflected by functions without such property. Take the process $\{1, 2\} \rhd \{1\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$ and suppose it has the following transition

$$\{1, 2\} \rhd \{1\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2 \xrightarrow{\tau, \{1,2\}}_{\text{PO}} O' \rhd \{1, 2, 3\} \Rightarrow p_1 \parallel \{1, 2, 3\} \Rightarrow p_2$$

where O' is a poset over $\{1, 2, 3\}$ with 1 greater than 2 and 3. Consider the function $\sigma \colon \{1, 2\} \to O$, where O has two elements such that $2 \prec_O 1$. Clearly σ is not poset-reflecting. If we apply σ and then closure $\downarrow_O$ to the source process, we get

$$O \rhd \{1, 2\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$$

but its τ transition is

$$O \rhd \{1, 2\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2 \xrightarrow{\tau, \{2\}}_{\text{PO}} O'' \rhd \{1, 2, 3\} \Rightarrow p_1 \parallel \{1, 2, 3\} \Rightarrow p_2$$

because only 2 is maximal for p_1, according to O. However, this transition cannot be obtained from the one of $\{1, 2\} \rhd \{1\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$ via an application of σ.

The following theorem is a consequence of Proposition 3.

Theorem 1 $\sim_{\text{PO}}$ *is closed under injective poset-reflecting functions. Explicitly: given $O \rhd k \sim_{\text{PO}} O \rhd k'$ and $\sigma \colon O \to O'$ injective and poset-reflecting, we have $O' \rhd (k\sigma) \downarrow_{O'} \sim_{\text{PO}} O' \rhd (k'\sigma) \downarrow_{O'}$.*

4.2 Immediate causes LTS

We now introduce a further refinement of the LTS_{PO}, called *immediate causes LTS* (LTS_{IC}): we keep only *immediate causes*, i.e. causes that are maximal w.r.t at least one of the atomic subprocesses, and we identify isomorphic states. Given a causal process k, its immediate causes w.r.t. a poset O are given by

$$ic_O(K \Rightarrow p) = max_O(K)$$
$$ic_O(k_1 \parallel k_2) = ic_O(k_1) \cup ic_O(k_2)$$

The notion of isomorphism we adopt is the following one

$$O \rhd k \cong O' \rhd k' \iff \sigma \colon O \cong O' \land k\sigma = k'$$

where $k\sigma$ applies σ to all the events occurring in k. We denote by $[O \rhd k]_{\cong}$ a canonical representative of the $\cong$-class of $O \rhd k$ and by $[O]_{\cong}$ its poset.

Definition 11 (Minimal P-process) A *minimal P-process* $O \blacktriangleright k$ is a P-process such that:

- O contains all and only the events in $\mathcal{K}(k)$;
- for each $K \Rightarrow p$ in k, $K \subseteq ic_O(k)$;
- it is the canonical representative of a $\cong$-equivalence class.

Given $O \triangleright k$, let O_i be O restricted to $ic_O(k)$; the corresponding minimal P-process is

$$\llbracket O \triangleright k \rrbracket = [O_i \triangleright norm_{O_i}(k)]_\cong$$

where $norm_O(K \Rightarrow p) = K \cap |O| \Rightarrow p$ and distributes over parallel. We denote by $\mu_{O \triangleright k}$ the map $[O_i]_\cong \to O$ obtained by composing the isomorphism $[O_i] \to O_i$ and the embedding $O_i \hookrightarrow O$.

Definition 12 (Immediate causes LTS) The *immediate causes LTS* (LTS$_{\mathtt{IC}}$) is the smallest LTS on minimal P-processes generated by the following rule

$$\frac{O \triangleright k \xrightarrow{a,K}_{\mathtt{PO}} O' \triangleright k'}{O \blacktriangleright k \xrightarrow[\mu_{O' \triangleright k}]{a,K}_{\mathtt{IC}} \llbracket O' \triangleright k \rrbracket}$$

This rule replaces the continuation with its minimal version and, in order to keep track of the original identity of events, equips the transition with a "history map", mapping canonical events to the original ones. In particular, the one with image 1 is the fresh event generated by the original transition. Notice that the continuation poset may contain non-maximal events, for instance

$$O \triangleright \{1, 2\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$$

with $2 \prec_O 1$ cannot be further reduced.

The notion of bisimilarity for LTS$_{\mathtt{IC}}$ is more involved: while, given two P-processes, we may find a common poset for them (if any), which enables them to be compared w.r.t. $\sim_{\mathtt{PO}}$, this is not possible in LTS$_{\mathtt{IC}}$, because its states must have minimal posets. In other words: posets have a meaning *local* to states. Therefore, we have to introduce an explicit correspondence between posets.

Definition 13 (Immediate causes bisimilarity) An *immediate causes bisimulation R* is a set of triples $(O \blacktriangleright k, \sigma, O' \blacktriangleright k')$ such that σ is a partial monotone bijection from O and O' and:

(i) if $O \blacktriangleright k \xrightarrow[h]{a,K}_{\mathtt{IC}} O'' \blacktriangleright k''$ then σ is defined on K, and there are $O' \blacktriangleright k' \xrightarrow[h']{a,K\sigma}_{\mathtt{IC}}$ $O''' \blacktriangleright k'''$ and σ' such that $(O'' \blacktriangleright k'', \sigma', O''' \blacktriangleright k''') \in R$ and $\sigma'(n) = m$ implies $h(n) = h'(m) = 1$ or $\sigma(h(n) - 1) = h'(m) - 1$;

(ii) if $O' \blacktriangleright k' \xrightarrow[h']{a,K}_{\mathtt{IC}} O''' \blacktriangleright k'''$ then σ is defined on K, and there are $O \blacktriangleright k \xrightarrow[h]{a,K\sigma^{-1}}_{\mathtt{IC}}$ $O'' \blacktriangleright k''$ and σ' as in the previous item.

The greatest such bisimulation is denoted $\sim_{\mathtt{IC}}$. We write $O \blacktriangleright k \sim_{\mathtt{IC}}^\sigma O' \blacktriangleright k'$ to mean $(O \blacktriangleright k, \sigma, O' \blacktriangleright k') \in \sim_{\mathtt{IC}}$.

Notice that states should be able to simulate each other only up to σ. The continuations are again related by a partial bijection σ' between O'' and O''', which should act consistently on names by "commuting" with history maps h, h' and σ. In the case $h(n) \neq 1 \neq h'(m)$, since h and h' have codomain $\delta_K(O')$ and $\delta_{\sigma(K)}(O'')$ respectively, where names in O and O' have been shifted by one, we should subtract one in order to recover the counterparts of $h(n)$ and $h(m)$ in O and O'.

We have the following correspondence between $\sim_{\mathtt{IC}}$ and $\sim_{\mathtt{PO}}$.

Theorem 2 $\sim_{\mathtt{IC}}$ *is fully abstract w.r.t.* $\sim_{\mathtt{PO}}$ *in the following sense:*

(i) *If* $O \triangleright k \sim_{\mathtt{PO}} O \triangleright k'$ *then* $[\![O \triangleright k]\!] \sim_{\mathtt{IC}} [\![O \triangleright k']\!]$;

(ii) *If* $O \blacktriangleright k \sim_{\mathtt{IC}}^{\sigma} O' \blacktriangleright k'$ *then for all* $\hat{O} \triangleright \hat{k}$ *and* $\hat{O} \triangleright \hat{k}'$ *such that:*

   (a) $[\![\hat{O} \triangleright \hat{k}]\!] = O \blacktriangleright k$ *and* $[\![\hat{O} \triangleright \hat{k}']\!] = O' \blacktriangleright k'$;

   (b) $\mu_{\hat{O} \triangleright \hat{k}}|_{dom(\sigma)} = \mu_{\hat{O} \triangleright \hat{k}'} \circ \sigma$;

   we have $\hat{O} \triangleright \hat{k} \sim_{\mathtt{PO}} \hat{O} \triangleright \hat{k}'$.

Remark 1 The transition system $\mathrm{LTS}_{\mathtt{IC}}$ is derived in a similar way as Montanari and Pistore causal automata. However, their derivation removes causal relations from states, keeping only the underlying set of events. This also affects the notion of bisimulation, where partial bijections are between sets of names. We have chosen to keep causal relations, and to give a compatible notion of bisimulation. This seems a natural choice, and it reflects what will produced, in a completely automatic and standard way, by our categorical construction.

5 Coalgebraic semantics

In this section we construct a coalgebra for causal semantics, equivalent to $\mathrm{LTS}_{\mathtt{PO}}$. Since we work in a more abstract setting, we do not need to concretely represent events as natural numbers to implement event generation. The notions introduced in the previous section are instances of our categorical machinery.

Definition 14 (Categories FinPos, P and $\mathbf{P}_m$) Let **FinPos** be the category of finite posets and monotone functions, and let **P** its skeletal category. The category $\mathbf{P}_m$ is the subcategory of **P** with only monic and poset-reflecting morphisms.

This is the category that we will take as index for presheaves. We now describe it structure.

Proposition 4 *The category* $\mathbf{P}_m$ *is small and has pullbacks.*

It lacks colimits, but the one we are interested in can be computed in **P** and then embedded into $\mathbf{P}_m$. We omit to specify this when using the expression "colimits in $\mathbf{P}_m$".

We introduce some notation for particular objects and morphisms of $\mathbf{P}_m$: we denote by $[k]$ the discrete poset with k elements and by $[k]^\top$ the same poset plus a top element; $b_k \colon [k] \to [k]^\top$ is the embedding of $[k]$ into $[k]^\top$; and $\top_k \colon [1] \to [k]^\top$ picks the top element in $[k]^\top$.

In $\mathbf{P}_m$ we can model the operator δ_K of Definition 9 as a pushout. Given $O \in |\mathbf{P}|$, let $K \colon [k] \hookrightarrow O$ be the subobject picking K within O. Then we have

$$
\begin{array}{ccc}
[k] & \xrightarrow{\ \ K\ \ } & O \\
{\scriptstyle b_k} \downarrow & & \downarrow {\scriptstyle old_O^K} \\
[k]^\top & \xrightarrow[\ new_O^K\ ]{} & \delta_K(O)
\end{array}
\qquad (6)
$$

Intuitively, the pushout merges the poset O with $[k]^\top$, identifying the causes in O, as given by K, and the k minimal elements of $[k]^\top$, so that the causes will be smaller than the same fresh event in $\delta_K(O)$.

Now we want to turn δ_K into an endofunctor on $\mathbf{P}_m$. However, δ_K does not define a proper functor, because K depends on the specific poset fed to δ_K. Therefore, we make δ independent from K by adding a new event for *each* possible set of independent causes, i.e. each discrete subposet of O.

This idea is formalized as follows. Let $K_1^k, \ldots, K_{n_k}^k \colon [k] \hookrightarrow O$ be all subposets of O with k elements. Notice that, by (5), the image of each K_i^k must be a discrete poset, in other words: K_i^k only picks events that are not already related. Suppose O has cardinality o. Then all the spans $[k]^\top \xleftarrow{\ \ b_k\ \ } [k] \xrightarrow{\ \ K_i^k\ \ } O$, namely those involved in (6), can be combined in the following colimit

$$
\begin{array}{ccccc}
 & [1] & \cdots & [o] & \\
{\scriptstyle b_1}\swarrow & & {\scriptstyle K_1^1}\ \ {\scriptstyle K_{n_o}^o} & & \searrow{\scriptstyle b_o} \\
[1]^\top & & O & & [o]^\top \\
 & \searrow & \downarrow{\scriptstyle old_O} & \swarrow & \\
{\scriptstyle new_O^{K_1^1}} & & O^\star & & {\scriptstyle new_O^{K_{n_o}^o}}
\end{array}
$$

We can define the following *allocation endofunctor* $\delta \colon \mathbf{P}_m \to \mathbf{P}_m$

$$
\delta(O) = O^\star \qquad \delta(\sigma \colon O \to O') = \sigma^*
$$

where σ^* is induced by the universal property of colimits.

Remark 2 Our allocation operator δ may seem inefficient: it generates a new event for each possible set of causes, but only one of them will appear in the continuation after a transition. However, having a functor on $\mathbf{P}_m$ allows us to lift it to presheaves in a way that ensures the existence of both left and right adjoint (giving Kan extensions along δ) for the lifted functor, and then preservation of both limits and colimits, which is essential for coalgebras employing such functor. Generation of unused events is not really an issue: as we will see later, it is always possible to recover the support of a process, i.e. the poset formed by events actually appearing in it.

Now we look at the category $\mathbf{Set}^{\mathbf{P}_m}$ of presheaves on posets. As mentioned, it is locally presentable and has all limits and colimits, in particular products and coproducts. The following constructs are relevant for us.

Presheaf of event names. $\mathcal{E}\colon \mathbf{P}_m \to \mathbf{Set}$ gives the set of event names occurring in $O \in |\mathbf{P}_m|$; formally:

$$\mathcal{E} = \mathrm{Hom}_{\mathbf{P}}([1], -)$$

Explicitly, $\mathcal{E}$ sends $O \in |\mathbf{P}_m|$ to $\mathbf{P}_m[[1], O]$, which is isomorphic to the underlying set of O, and $\sigma\colon O \to O'$ in $\mathbf{P}_m$ to the function $\lambda e \in \mathbf{P}_m[[1], O].\sigma \circ e$, which renames the event e according to σ.

Finite powerset. $\mathscr{P}_f\colon \mathbf{Set}^{\mathbf{P}_m} \to \mathbf{Set}^{\mathbf{P}_m}$, defined as $\mathcal{P}_f \circ (-)$, where $\mathcal{P}_f$ is the finite powerset on $\mathbf{Set}$; we also consider the *non-empty bounded powerset* $\mathscr{P}^+_{\leq k}$, that produces only non-emtpy subsets of cardinality at most k.

Event allocation operator. $\Delta\colon \mathbf{Set}^{\mathbf{P}_m} \to \mathbf{Set}^{\mathbf{P}_m}$, given by $(-) \circ \delta$. Explicitly, for $P\colon \mathbf{P}_m \to \mathbf{Set}$ and $O \in |\mathbf{P}_m|$, $\Delta P(O) = P(\delta(O))$. Intuitively, it generates processes with additional fresh events.

Presheaf of labels. $\mathcal{L}\colon \mathbf{P}_m \to \mathbf{Set}$ given by

$$\mathcal{L}(O) = (Act \cup \{\tau\}) \times \mathscr{P}^+_{\leq 2}\mathcal{E}(O)$$

For each $O \in |\mathbf{P}_m|$, this functor gives pairs (a, K) of an action a and a set of causes K, selected among events in O. Notice that it is sufficient to consider cause sets with cardinality at most two, because in $\mathrm{LTS}_{\mathsf{PO}}$ non-τ actions have one maximal cause, while synchronizations have at most two maximal causes. However our construction can be immediately extended to process models, like e.g. Petri n-safe nets, where more than two processes can synchronize at a time.

We use these operators to define our behavioral endofunctor.

Definition 15 (Behavioral functor) The behavioral functor $B\colon \mathbf{Set}^{\mathbf{P}_m} \to \mathbf{Set}^{\mathbf{P}_m}$ is

$$BP = \mathscr{P}_f(\mathcal{L} \times \Delta P) \ .$$

To understand this definition, consider a B-coalgebra (P, ρ). Given $O \in |\mathbf{P}_m|$ and $p \in P(O)$, $\rho_O(p)$ is a finite set of triples (a, K, p'), telling that p' is the continuation of p after observing a, K. The continuation always belongs to $\Delta P(O)$, i.e. to $P(\delta(O))$, because every transition allocates a new event.

The category B-**Coalg** is well-behaved: it has a final B-coalgebra and B-bisimulation matches with the equivalence induced by the final morphism. This is thanks to the following properties.

Theorem 3 *B is accessible and preserves weak pullbacks.*

B-coalgebras can be regarded as particular LTSs whose states are elements of presheaves, i.e. pairs $O \triangleright p$.

Definition 16 ($\mathbf{P}_m$-ILTS) A a $\mathbf{P}_m$-*indexed labelled transition system* ($\mathbf{P}_m$-ILTS) is a pair $(P, \Longrightarrow)$ of a presheaf $P\colon \mathbf{P}_m \to \mathbf{Set}$ and a finite-branching transition relation $\Longrightarrow \subseteq \int P \times \int \mathcal{L} \times \int P$ of the form:

$$O \triangleright p \overset{a,K}{\Longrightarrow} \delta(O) \triangleright p' \qquad (a, K) \in \mathcal{L}(O)$$

such that, for each morphism $\sigma\colon O \to O'$ in $\mathbf{P}_m$:

(i) if $O \triangleright p \stackrel{l}{\Rightarrow} \delta(O) \triangleright p'$ then $O' \triangleright t[\sigma] \stackrel{l[\sigma]}{\Longrightarrow} \delta(O') \triangleright p'[\delta\sigma]$ (transitions are *preserved* by σ);

(ii) if $O' \triangleright t[\sigma] \stackrel{l}{\Rightarrow} \delta(O') \triangleright p'$ then there are l' and $\delta(O) \triangleright p''$ such that $l'[\sigma] = l$, $p''[\delta\sigma] = p'$ and $O \triangleright p \stackrel{l'}{\Rightarrow} \delta(O) \triangleright p''$ (transitions are *reflected* by σ);

Proposition 5 $\mathbf{P}_m$*-ILTSs are in bijection with B-coalgebras.*

The natural notion of bisimulation for these transition systems is $\mathbf{P}_m$*-indexed bisimulation.*

Definition 17 ($\mathbf{P}_m$-indexed bisimulation) A $\mathbf{P}_m$*-indexed bisimulation* on a $\mathbf{P}_m$-ILTS $(P, \Longrightarrow)$ is an indexed family of relations $\{R_O \subseteq P(O) \times P(O)\}_{O \in |\mathbf{P}_m|}$ such that, for all $(p, q) \in R_O$

(i) if $O \triangleright p \stackrel{a,K}{\Longrightarrow} O' \triangleright p'$ then there is $O' \triangleright q'$ such that $O \triangleright q \stackrel{a,K}{\Longrightarrow} O' \triangleright q'$ and $(p', q') \in R_{O'}$;

(ii) for all $\sigma \colon O \to O'$, $(p[\sigma], q[\sigma]) \in R_{O'}$.

This definition closely resembles that of poset-indexed causal bisimulation (Definition 10). We have an additional condition (ii), requiring closure under morphisms of $\mathbf{P}_m$. This is not satisfied by all poset-indexed causal bisimulation, but it holds for the greatest one (Theorem 1).

We have the following correspondence.

Proposition 6 *Let (P, ρ) be a B-coalgebra. Then every B-bisimulation is equivalent to a $\mathbf{P}_m$-indexed bisimulation on the induced $\mathbf{P}_m$-ILTS.*

Unfortunately, the converse is not true: there are bisimulations on some $\mathbf{P}_m$-ILTS s that cannot be turned into B-bisimulations. This has been pointed out in [17, 3.3.3, Anomaly] for the case of the π-calculus. The solution given there is to narrow the class of presheaves under consideration to pullback-preserving presheaves.

Theorem 4 (cf. Theorem 4.2.5 of [17]) *Let (P, ρ) a B-coalgebra. If P preserves pullbacks then every $\mathbf{P}_m$-indexed bisimulation on the induced $\mathbf{P}_m$-ILTS is also a B-bisimulation on (P, ρ).*

We now show that $\mathsf{LTS}_{\mathsf{P0}}$ can be represented as a $\mathbf{P}_m$-ILTS. We form a presheaf from states of $\mathsf{LTS}_{\mathsf{P0}}$ as follows.

Definition 18 (Presheaf of P-processes) The *presheaf of P-processes* $\mathscr{C} \colon \mathbf{P}_m \to \mathbf{Set}$ is given by

$$\mathscr{C}(O) = \{k \mid O \triangleright k \text{ is a state of } \mathsf{LTS}_{\mathsf{P0}}\}$$
$$\mathscr{C}(\sigma \colon O \to O') = \lambda O \triangleright k.O' \triangleright k\sigma{\downarrow}_{O'}$$

The action of $\mathscr{C}$ on morphisms needs to apply the closure operator, after renaming the process: this guarantees that the result is a proper P-process. Notice that elements of $\mathscr{C}$ are defined over events with an abstract identity, which may not be natural numbers. More precisely, we implicitly assume the following translation form states of $\mathsf{LTS}_{\mathsf{P0}}$. For each finite poset O over natural numbers, take the isomorphism $\varphi_O \colon O \to [O]$ within **FinPos**, where $[O]$ is the object of $\mathbf{P}_m$ canonically representing the isomorphism class of O. Then $[O] \triangleright k\varphi$ gives the proper element of $\mathscr{C}$ corresponding to $O \triangleright k$.

We have the following property.

Lemma 1 $\mathscr{C}$ *preserves pullbacks.*

We are ready to translate LTS_{P0} to a $\mathbf{P}_m$-ILTS.

Definition 19 (Poset $\mathbf{P}_m$-ILTS$_{\text{P0}}$) The *Poset $\mathbf{P}_m$-ILTS* ($\mathbf{P}_m$-ILTS$_{\text{P0}}$) is the smallest one generated by the rule

$$\frac{O \rhd k \xrightarrow{a,K}_{\text{P0}} O' \rhd k'}{O \rhd k \overset{a,K}{\Longrightarrow}_{\text{P0}} \delta(O) \rhd k'[new_O^K(\top_{\#K})/1, old_O]}$$

The rule in this definition computes a casual process with poset $\delta(O)$ from $O' \rhd k'$. This is done by replacing 1, the concrete fresh event in k', with the abstract fresh event associated to K in $\delta(O)$, via the corresponding colimit map. All the other events are renamed accordingly. This definition gives a proper $\mathbf{P}_m$-ILTS: transitions are clearly of the required form, and preservation and reflection of transition follows from analogous properties of LTS_{P0} (Proposition 3).

We call *causal B-coalgebra* the one equivalent to $(\mathscr{C}, \Longrightarrow_{\text{P0}})$. We have the following theorem, which collects the results of this section, instantiated to the causal B-coalgebra.

Theorem 5 $\mathbf{P}_m$-*indexed bisimulations on* $(\mathscr{C}, \Longrightarrow_{\text{P0}})$ *are equivalent to:*

- *B-bisimulations on the causal B-coalgebra;*
- *poset-indexed causal bisimulations closed under injective and poset-reflecting renamings.*

In particular, we have that the greatest $\mathbf{P}_m$-indexed bisimulation, B-bisimilarity on the causal B-coalgebra and $\sim_{\text{P0}}$ are all equivalent, thanks to Theorem 1.

Remark 3 The carrier of the final B-coalgebra can be intuitively described as follows. It is a presheaf whose elements are pairs $O \rhd T$ of a poset and a tree T. When $O \rhd T$ is the image of a P-process $O \rhd k$ via the final morphism, then T is similar to a (strong) causal tree for k, but its edges only exhibit the most recent events. The "missing information", i.e. the full history of events, is provided by O.

6 From coalgebras to HD-automata

In order to give a characterization of our coalgebra in terms of named sets, we employ the results of [7]. Here authors define a *symmetry group over a category* $\mathbf{C}$ to be a collection of morphisms in $\mathbf{C}[c,c]$, for any $c \in |\mathbf{C}|$, which is a group w.r.t. composition of morphisms. Then they take families of such groups as their notion of generalized named sets. A first result establishes the equivalence between these families and *coproducts of symmetrized representables*, that are functors of the form

$$\sum_{i \in I} \text{Hom}_{\mathbf{C}}(c_i, \text{-})_{/\Phi_i}$$

where Φ_i is a symmetry group over $\mathbf{C}$ with domain c_i, and the quotient identifies morphisms that are obtained one from the other by precomposing elements of Φ_i. These functors, in turn, are shown to be isomorphic to *wide-pullback-preserving* presheaves on $\mathbf{C}$, a wide pullback being the limit of a diagram with an arbitrary

number of morphisms pointing to the same object (pullbacks are a special case, with two such morphisms). The following theorem summarizes the described results.

Theorem 6 *Let $\mathbf{C}$ be a category that is small, has wide pullbacks, and such that all its morphisms are monic and those in $\mathbf{C}[c,c]$ are isomorphisms, for every $c \in |\mathbf{C}|$. Then every wide-pullback-preserving $P \in |\mathbf{Set}^{\mathbf{C}}|$ is equivalent to a coproduct of symmetrised representables.*

Our category $\mathbf{P}_m$ satisfies the hypothesis of this theorem: it is small and has wide pullbacks due to the existence of pullbacks. In fact, the diagram of a wide pullback in $\mathbf{P}_m$ is formed by a finite number of morphisms, because a finite poset always has a finite number of ingoing poset-reflecting monomorphisms, so its limit can be computed via binary pullbacks. Moreover, $\mathbf{P}_m$ has only monos, by definition, and $\mathbf{P}_m[O,O]$ clearly has only isomorphisms, for each $O \in |\mathbf{P}_m|$. Finally, our presheaf of processes $\mathscr{C}$ preserves (wide) pullbacks, so there exists an equivalent coproduct of symmetrized representables.

Now that we have proved that our categorical setting is suitable for HD-automata, we can translate the coalgebra of the previous section to a HD-automaton.∎ We work in a more concrete setting: we introduce a notion of named set closer to a more traditional one, but indeed equivalent to the families mentioned above. Given a set S of morphism and a morphism σ in $\mathbf{P}_m$, we write $S \circ \sigma$ for the set $\{\tau \circ \sigma \mid \tau \in S\}$ (analogously for $\sigma \circ S$).

Definition 20 Let $\mathbf{Sym}(\mathbf{P}_m)$ be the category defined as follows:

– objects are sets $\Phi \subseteq \mathbf{P}_m[O,O]$ that are groups w.r.t. composition in $\mathbf{P}_m$;
– morphisms $\Phi_1 \to \Phi_2$ are sets of morphisms $\sigma \circ \Phi_1$ such that $\sigma \colon dom(\Phi_1) \to dom(\Phi_2)$ and $\Phi_2 \circ \sigma \subseteq \sigma \circ \Phi_1$.

Definition 21 The category $\mathbf{P}_m\text{-}\mathbf{Set}$ is defined as follows:

– objects are $\mathbf{P}_m$-*named sets*, that are pairs $N = (Q_N, \mathsf{G}_N)$ of a set Q_N and a function $\mathsf{G}_N \colon Q \to |Sym(\mathbf{P}_m)|$. The *local poset* of $q \in Q_N$, denoted $\|q\|$, is $dom(\sigma)$, for any $\sigma \in \mathsf{G}_N(q)$.
– morphisms $f \colon N \to M$ are $\mathbf{P}_m$-*named functions*, that are pairs (h, Σ) of a function $h \colon Q_N \to Q_M$ and a function Σ mapping each $q \in Q_N$ to a morphism $\mathsf{G}_M(h(q)) \to \mathsf{G}_N(q)$ in $Sym(\mathbf{P}_m)$.

Since all the functors in Definition 15 can be restricted to wide-pullback-preserving presheaves in $\mathbf{Set}^{\mathbf{P}_m}$, we can define an equivalent functor on $\mathbf{P}_m\text{-}\mathbf{Set}$.

Definition 22 The behavioral functor $\widetilde{B} \colon \mathbf{P}_m\text{-}\mathbf{Set} \to \mathbf{P}_m\text{-}\mathbf{Set}$ is defined by lifting all functors in the definition of B along the equivalence.

The following result is an immediate consequence.

Theorem 7 *The category B-**Coalg** is equivalent to $\widetilde{B}$-**Coalg**.*

In particular, the equivalence relates the final B-coalgebra and the final $\widetilde{B}$-coalgebra,∎ and their final morphisms. Moreover, since kernels are preserved by equivalence, identifications made by the final morphisms are preserved, hence behavioral equivalence is preserved too.

In the rest of this section we give an explicit description of the $\mathbf{P}_m$-named set produced from $\mathscr{C}$ by the equivalence. Its elements will be minimal P-processes: we will show that the translation from P-processes to minimal ones is achieved via categorical constructions. We need the notions of support, seed and orbit.

Definition 23 (Support and seed) Given $O \triangleright k$, its *support*, denoted $supp(k)$, is the wide-pullback-object of the following morphisms

$$\{\sigma \colon O' \hookrightarrow O \mid \exists O' \triangleright k' : k'[\sigma] = k\}$$

Let Σ_k be the embedding $supp(k) \hookrightarrow O$ given by the pullback. Then the *seed* of k, denoted $seed(k)$, is the unique element of $\mathscr{C}(supp(k))$ such that $seed(k)[\Sigma_k] = k$.

As shown in [7, 12], an essential condition for existence and uniqueness of the seed is preservation of pullbacks, which holds for $\mathscr{C}$ (Lemma 1). The seed operation achieves the first two properties of minimal P-processes (see Definition 11): $seed(k)$ just contains immediate causes for each of its components and $supp(O)$ contains all and only those causes. This is illustrated by the following example.

Example 2 Consider the following P-process

$$O \triangleright \{1, 2, 3\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$$

where O is a poset over $\{1, 2, 3, 4\}$ with

$$2 \prec_O 1 \qquad 3 \prec_O 1 \qquad 2 \prec_O 4 \qquad 3 \prec_O 4 \ .$$

Then the set of morphisms of Definition 23 has two elements $f_1 \colon O_1 \to O$ and $f_2 \colon O_2 \to O$ where O_1 is a poset over $\{1, 2, 3\}$ with $2 \prec_{O_1} 1$ and $3 \prec_{O_1} 1$, and O_2 is a poset over $\{1, 2\}$ such that $2 \prec_{O_2} 1$. These morphisms just map events preserving their names. We have

$$O_1 \triangleright \{1, 2, 3\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2 \xmapsto{\;[f_1]\;} O \triangleright \{1, 2, 3\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$$

$$O_2 \triangleright \{1, 2\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2 \xmapsto{\;[f_2]\;} O \triangleright \{1, 2, 3\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$$

It is easy to check that the pullback object of f_1 and f_2 is O_2, so the corresponding seed is $O_2 \triangleright \{1, 2\} \Rightarrow p_1 \parallel \{2\} \Rightarrow p_2$. Notice that the event 4 has been discarded, because it does not syntactically appear in the process, but also 3, because it is not an immediate cause for either p_1 or p_2.

Definition 24 (Orbit) The *orbit* of $O \triangleright k$ is

$$orb(k) = \{\ k[\sigma] \mid \sigma \in \mathbf{P}_m[O, O]\}$$

We denote by $[k]^o$ a canonical choice of an element of $orb(k)$.

The orbit of k the set of elements obtained by applying to it all functions induced by poset automorphisms. The representative $[k]^o$ plays the same role as $[O \triangleright k]_{\cong}$ defined in the previous section. However, each $\cong$-equivalence class may contain P-processes with different, but isomorphic, posets. These posets all become the same one in $\mathbf{P}_m$, due to skeletality, so it is enough to consider automorphisms, which are always iso in $\mathbf{P}_m$.

Definition 25 The $\mathbf{P}_m$-named set of minimal P-processes is $(C, \mathtt{G}_C)$, where

$$C = \{supp(k) \blacktriangleright [seed(k)]^o \mid O \rhd k \in \int \mathscr{C}\}$$

$$\mathtt{G}_C = \lambda O \blacktriangleright k.\{\Phi \in |\mathbf{Sym}(\mathbf{P}_m)| \mid dom(\Phi) = O \land \forall \sigma \in \Phi : k[\sigma] = k\}$$

Let us explain this definition. The set C is produced from elements of $\mathscr{C}$: for each of these, we compute the seed, and then we only take the canonical representative for the seed's orbit. The former operation achieves the third requirement for minimal P-processes. The symmetry group for a process is the set of poset automorphisms fixing the process.

The HD-automaton on $(C, \mathtt{G}_C)$ in $\widetilde{B}\text{-}\mathbf{Coalg}$, equivalent to the causal B-coalgebra, is the category-theoretic counterpart of $\mathrm{LTS}_{\mathtt{IC}}$: states are minimal P-processes, and transitions have history maps. In $\widetilde{B}\text{-}\mathbf{Coalg}$, history maps come from the fact that coalgebra structure maps are $\mathbf{P}_m$-named functions, so they are equipped with backward morphisms towards the poset of the source state. However, there is a crucial difference: states of the HD-automaton have symmetries, which allow for further identifications of states. For instance, the process

$$[2] \blacktriangleright \{1\} \Rightarrow p \parallel \{2\} \Rightarrow p$$

can be associated the symmetry $\{id_{[2]}, (1\ 2), (1\ 2)^{-1}\}$, because swapping 1 and 2 yields bisimilar processes.

7 Conclusions

In this paper we have given a construction for obtaining compact models of causal semantics. In order to do this, we have equipped causal processes with nominal structures, namely posets over event names, representing causal relations. We have presented a first, set-theoretic version of our construction, along the lines of [14], and then a category-theoretic one that employs standard constructs and results for nominal calculi, namely presheaf-based coalgebras and their equivalence with HD-automata. The categorical version is much more concise and natural. In particular, reducing the state-space and showing that this operation preserves the semantics require some technical effort in the set-theoretic version, whereas the categorical version employs a general construction that automatically performs this reduction in a semantics-preserving way.

This paper is mainly related to [14]. While the definition of causal automata should be attributed to the ingenuity of their authors, the derivation of HD-automata we show in this paper is due to a general categorical construction. The main difference between the two notions of automata is in the information each state keeps: causal automata keep events, but discard their causal relations; our HD-automata retain causal relations, in the form of posets, and, in addition, there are symmetry groups over them. This allows for a further reduction of states. A representation of events in terms of names, with the aim of capturing Darondeau-Degano causal semantics, can also be found in [5], even if in the different context of tile systems. We can cite [6] for the introduction of transitions systems for causality whose states are elements of presheaves, intended to model the causal semantics of the π-calculus as defined in [4]. However, the index of a state is a set of

names, without any information about events and causal relations. The advantage of our index category is that it allows reducing the state-space in an automatic way, exploiting a standard categorical construction. This cannot be done in the framework of [6]. Finally, an HD-automaton for causality has been described in [8], but it is derived as a direct translation of causal automata and its states do not take into account causal relations.

As a future work, we plan to further investigate the correspondence with [14]. In particular, the relationship between the notions of bisimulations is not clear.

A Proofs

Proof (of Proposition 3) We show (i), the other point is analogous. We rely on the following properties of $\mathrm{LTS_{DD}}$, which can be easily checked by induction on the inference via rules Figure 1: for all $O \rhd k$, $\sigma \colon O \to O'$ injective and poset-reflecting, and $O'' \subseteq O$, we have

$$k \xrightarrow{a,K}_{\mathrm{DD}} k' \implies k\sigma \xrightarrow{a,K\sigma}_{\mathrm{DD}} k'\sigma^+ \tag{7}$$

$$k \xrightarrow{a,K}_{\mathrm{DD}} k' \implies k{\downarrow}_{O''} \xrightarrow{a,K{\downarrow}_{O''}}_{\mathrm{DD}} k'{\downarrow}_{\delta_{K{\downarrow}_{O''}}(O'')} \tag{8}$$

where $\downarrow$ is extended to sets of causes by performing their closure. Property (8) is the less obvious: the idea is that, since labels of $\mathrm{LTS_{DD}}$ shows the whole history of causes for an action, the additional ones in O'', preceding K, should be shown as well; the continuation is closed accordingly.

Now, take $O \rhd k \xrightarrow{a,K}_{\mathrm{P0}} \delta_K(O) \rhd k'$ and $\sigma \colon O \to O'$ injective and poset-reflecting. By the rule in Definition 9 there is a transition in $\mathrm{LTS_{DD}}$

$$k \xrightarrow{a,K'}_{\mathrm{DD}} k \qquad K = max_O(K') \ ;$$

If we apply (7) and then (8) to this transition, we get

$$(k\sigma){\downarrow}_{O'} \xrightarrow{a,K'\sigma{\downarrow}_{O'}}_{\mathrm{DD}} (k'\sigma^+){\downarrow}_{\delta_{K'\sigma{\downarrow}_{O'}}(O')} \ . \tag{9}$$

Observe that

$$\begin{aligned} max_{O'}(K'\sigma{\downarrow}_{O'}) &= max_{O'}(K'\sigma) \\ &= max_O(K')\sigma \\ &= K\sigma \end{aligned} \tag{10}$$

The first equation holds because ${\downarrow}_{O'}$ only adds to $K'\sigma$ events that are smaller w.r.t. $\prec_{O'}$. The second equation holds because σ is poset-reflecting: this prevents some maximal, thus unrelated, events in K' to become non-maximal in $K'\sigma$ due to additional causal relations involving them in O'.

From (10) it follows

$$\begin{aligned} \delta_{K'\sigma{\downarrow}_{O'}}(O') &= \delta_{max_{O'}(K'\sigma{\downarrow}_{O'})}(O') \\ &= \delta_{K\sigma}(O') \end{aligned} \tag{11}$$

where the first equation come from transitivity: adding causal relations from 1 to $K'\sigma$ and its past causes in O' is equivalent to adding relations to only maximal events in $K'\sigma$, because transitivity will take care of adding the missing ones.

We conclude by applying (11) to the continuation of (9), which becomes of the required form, and then the rule of Definition 9 to infer the required transition. The computation of the maximal causes yields the desired result, thanks to (10). □

Proof (of Proposition 2)

($\implies$) We prove that the following relation is a causal bisimulation

$$R = \{(k, k') \mid \exists O \colon O \rhd k \sim_{\text{P0}} O \rhd k'\}$$

Suppose $O \rhd k \xrightarrow{a,K}_{\text{P0}} O' \rhd k''$ and the simulating transition is $O \rhd k' \xrightarrow{a,K}_{\text{P0}} O' \rhd k'''$. Then we can recover simulating transitions in LTS$_{\text{DD}}$ as follows

$$k \xrightarrow{a,K\downarrow_O}_{\text{DD}} k'' \qquad k' \xrightarrow{a,K\downarrow_O}_{\text{DD}} k'''$$

and, since $O' \rhd k'' \sim_{\text{P0}} O' \rhd k'''$, we have $(k'', k''') \in R$.

($\impliedby$) We prove that the following relation is a poset-indexed causal bisimulation

$$R_O = \{(O \rhd k, O \rhd k') \mid k \sim_{\text{DD}} k'\}$$

Suppose $k \xrightarrow{a,K}_{\text{DD}} k''$ and $k' \xrightarrow{a,K}_{\text{DD}} k'''$. Then the rule in Definition 9, from both transitions, generates transitions in LTS$_{\text{P0}}$ with the same label, and the same source and target posets. If the latter is $\delta_M(O)$, then from $k'' \sim_{\text{DD}} k'''$ it follows $(\delta_M(O) \rhd k'', \delta_M(O) \rhd k''') \in R_{\delta_M(O)}$.
$\square$

We need the following lemmata, whose proofs are straightforward.

Lemma 2 *For each $O \rhd k$, $\mu_{O \rhd k}$ is injective and poset-reflecting.*

Lemma 3 *Let $[\![O \rhd k]\!] = O' \rhd k'$, then $(k'\mu_{O \rhd k})\downarrow_O = k$.*

An immediate consequence is the following one.

Lemma 4 $O \blacktriangleright k \xrightarrow[h]{a,K}_{\text{IC}} O' \rhd k'$ *is generated by* $O \rhd k \xrightarrow{a,K}_{\text{P0}} \delta_K(O) \rhd (k'h)\downarrow_{\delta_K(O)}$ *via the rule in Definition 12.*

Proof **Point (i).** We prove that the following relation is an immediate causes bisimulation

$$R = \{ \ ([\![O \rhd k]\!], \sigma, [\![O \rhd k']\!]) \mid O \rhd k \sim_{\text{P0}} O \rhd k',$$
$$\sigma(n) = m \iff \mu_{O \rhd k}(n) = \mu_{O \rhd k'}(m)\}$$

We only show how the derivation of a simulating transition for one of $[\![O \rhd k]\!]$. The symmetric case is analogous. Let $[\![O \rhd k]\!] = \tilde{O} \blacktriangleright \tilde{k}$, and suppose it has the following transition

$$\tilde{O} \blacktriangleright \tilde{k} \xrightarrow[h]{a,K}_{\text{IC}} \tilde{O}' \blacktriangleright \tilde{k}_1 \tag{12}$$

Then, by Lemma 4, we have

$$\tilde{O} \rhd \tilde{k} \xrightarrow{a,K}_{\text{P0}} \delta_K(\tilde{O}) \rhd \tilde{k}_2 \qquad \tilde{k}_2 = (\tilde{k}_1 h)\downarrow_{\delta_K(\tilde{O})} \tag{13}$$

Now, let $\mu_1 = \mu_{O \rhd k}$: by Lemma 2, we can apply (i) of Proposition 3 to the last transition and get

$$O \rhd (\tilde{k}\mu_1)\downarrow_O = k \xrightarrow{a,K\mu_1}_{\text{P0}} \delta_{K\mu_1}(O) \rhd \tilde{k}_3 \qquad \tilde{k}_3 = (\tilde{k}_2\mu_1^+)\downarrow_{\delta_{K\mu_1}(O)} \tag{14}$$

where $(\tilde{k}\mu_1)\downarrow_O = k$ is due to 3. Therefore we have $[\![\delta_{K\mu_1}(O) \rhd \tilde{k}_3]\!] = \tilde{O}' \blacktriangleright \tilde{k}_1$, and the associated map $\tilde{O}' \to \delta_{K\mu_1}(O)$ is given by composing those applied to the continuations of (13) and (14), namely

$$\mu_{\delta_{K\mu_1}(O) \rhd \tilde{k}_3} = \mu_1^+ \circ h \tag{15}$$

Suppose (14) can be simulated by $O \rhd k'$ as follows

$$O \rhd k' \xrightarrow{a,K\mu_1}_{\text{P0}} \delta_{K\mu_1}(O) \rhd k'' \ . \tag{16}$$

Let $[\![O \triangleright k']\!] = \tilde{O}' \blacktriangleright \tilde{k}'$ and $\mu_2 = \mu_{O \triangleright k'}$. Then, by Lemma 2, we can apply (ii) of Proposition 3 to (16) and μ_2, obtaining

$$\tilde{O}' \triangleright \tilde{k}' \xrightarrow{a,\tilde{K}}_{\text{PO}} \delta_{\tilde{K}}(\tilde{O}') \triangleright \tilde{k}'' \tag{17}$$

such that $\tilde{K}\mu_2 = K\mu_1$ and $(\tilde{k}''\mu_2^+)\!\downarrow_{\delta_{K\mu_1}(O)} = k''$. Let $h' = \mu_{\delta_{\tilde{K}}(\tilde{O}') \triangleright \tilde{k}''}$. Applying the rule of Definition 12 to (17) we get

$$\tilde{O} \blacktriangleright \tilde{k} \xrightarrow[h']{a,\tilde{K}}_{\text{IC}} [\![\delta_{\tilde{K}}(\tilde{O}') \triangleright \tilde{k}'']\!] \tag{18}$$

It is not difficult to check that $[\![\delta_{\tilde{K}}(\tilde{O}') \triangleright \tilde{k}'']\!] = [\![\delta_{K\mu_1}(O) \triangleright k'']\!]$: the intuition is that taking immediate causes from $O \triangleright k'$, then adding a new event, and taking again immediate causes, has the same result as adding the event straight away and then taking immediate causes. Therefore $\mu_{\delta_{K\mu_1}(O) \triangleright k''}$ can be expressed as composition of the maps from the continuation of (18) to that of (17), and from the latter to that of (16), namely

$$\mu_{\delta_K(O) \triangleright k''} = \mu_2^+ \circ h' \tag{19}$$

Now we shall check that (12) and (18) are actually simulating transitions. From $\tilde{K}\mu_2 = K\mu_1$ and the definition of σ (recall that $\mu_1 = \mu_{O \triangleright k}$ and $\mu_2 = \mu_{O \triangleright k'}$) it follows $\tilde{K} = K\sigma$. Let σ' be defined by

$$\sigma'(n) = m \iff \mu_1^+(h(n)) = \mu_2^+(h'(m))$$

The right equation is equivalent to $\mu_{\delta_{K\mu_1}(O) \triangleright k''}(n) = \mu_{\delta_{K\mu_1}(O) \triangleright \tilde{k}_3}(m)$, by (15) and (19), so $([\![\delta_{K\mu_1}(O) \triangleright k'']\!], \sigma', [\![\delta_{K\mu_1}(O) \triangleright \tilde{k}_3]\!]) \in R$. It remains to check that it is a proper bisimulation triple. Take n, m such that $\sigma'(n) = m$. We have

$$\begin{aligned} h(n) = 1 &\iff \mu_1^+(h(n)) = 1 \\ &\iff \mu_2^+(h'(m)) = 1 \\ &\iff h'(m) = 1 \end{aligned}$$

where the first and last implication follow from the definition of μ_1^+ and μ_2^+. For $h(n), h(m) > 1$, expanding the definition of μ_1^+ and μ_2^+ we get $\mu_1(h(n) - 1) = \mu_2(h'(m) - 1)$, hence $\sigma(h(n) - 1) = h'(m) - 1$, by the definition of R.

Point (ii). We prove that the following family of relations is a poset-indexed causal bisimulation

$$R_{\hat{O}} = \{(\hat{O} \triangleright \hat{k}, \hat{O} \triangleright \hat{k}') \mid O \blacktriangleright k \sim_{\text{IC}}^{\sigma} O' \blacktriangleright k', \text{(ii.a) and (ii.b) hold}\}$$

To ease notation, let $\mu_1 = \mu_{\hat{O} \triangleright \hat{k}}$ and $\mu_2 = \mu_{\hat{O} \triangleright \hat{k}'}$. We only show that each transition of $\hat{O} \triangleright \hat{k}$ can be simulated by one of $\hat{O} \triangleright \hat{k}'$. The proof for the symmetric statement is analogous. Suppose $\hat{O} \triangleright \hat{k}$ has the following transition

$$\hat{O} \triangleright \hat{k} \xrightarrow{a,K}_{\text{PO}} \delta_K(\hat{O}) \triangleright \hat{k}' \tag{20}$$

Then, by (ii) of Proposition 3 and Lemma 2, we can rename it via μ_1 and get

$$O \triangleright k \xrightarrow{a,\tilde{K}}_{\text{PO}} \delta_{\tilde{K}}(O) \triangleright \tilde{k}' \qquad \tilde{k}' = (\hat{k}'\mu_1^+)\!\downarrow_{\delta_{\tilde{K}}(O)}$$

where $\tilde{K}\mu_1 = K$. Applying the rule in Definition 12, we get

$$O \blacktriangleright k \xrightarrow[h]{a,\tilde{K}}_{\text{IC}} [\![\delta_{\tilde{K}}(O) \triangleright \tilde{k}']\!]$$

Now suppose that this transition can be simulated by $O' \blacktriangleright k'$ as follows

$$O' \blacktriangleright k' \xrightarrow[h']{a,\tilde{K}\sigma}_{\text{IC}} O'' \blacktriangleright k'' \ ,$$

and $(\llbracket \delta_{\tilde{K}}(O) \rhd \tilde{k}' \rrbracket, \sigma', O'' \blacktriangleright k'')$. Applying Lemma 4 we get

$$O' \rhd k' \xrightarrow{a, \tilde{K}\sigma}_{\text{P0}} \delta_{\tilde{K}\sigma}(O') \rhd k_1'' \qquad k_1'' = (k''h')\downarrow_{\delta_{\tilde{K}\sigma}(O')}$$

and then, from (i) of Proposition 3 with renaming μ_2, and applying Lemma 3 to the resulting source process, we get

$$\hat{O} \rhd \hat{k}' \xrightarrow{a, (\tilde{K}\sigma)\rho} \delta_K(\hat{O}) \rhd k_2'' \qquad k_2'' = (k_1''\mu_2{}^+)\downarrow_{\delta_K(\hat{O})} \tag{21}$$

From hypothesis (ii.b) we have $(\tilde{K}\sigma)\mu_2 = \tilde{K}\mu_1 = K$, so (20) and (21) have the same label. Finally, we have to check (ii.a) and (ii.b) on their continuations:

(ii.a) We have $\llbracket \delta_{\tilde{K}}(O) \rhd \tilde{k}' \rrbracket = \llbracket \delta_K(\hat{O}) \rhd \hat{k}' \rrbracket$ and $\llbracket \delta_K(\hat{O}) \rhd (\tilde{k}''\rho)\downarrow_{\delta_K(\hat{O})} \rrbracket = O'' \blacktriangleright k''$, as already explained in point (i) for analogous transitions;

(ii.b) Let $\tilde{\mu}_1 = \mu_{\delta_K(\hat{O}) \rhd \hat{k}'}$ and $\mu_{\delta_K(\hat{O}) \rhd \hat{k}}$. We have to check $\tilde{\mu}_1|_{dom(\sigma')} = \tilde{\mu}_2 \circ \sigma'$. Take $n \in dom(\sigma')$. Then, by the equivalence between continuations exhibited in the proof of (ii.a), we have $\tilde{\mu}_i = \mu_i{}^+ \circ h$, $i = 1, 2$, so

$$\tilde{\mu}_1(n) = \tilde{\mu}_2(\sigma'(n)) \iff \mu_1{}^+(h(n)) = \mu_2{}^+(h'(\sigma'(n)))$$

We have two cases, by the property relating σ' with h, h' and σ:
 $-\ h(n) = h'(\sigma(n)) = 1$, then $\mu_1{}^+(h(n)) = 1 = \mu_2{}^+(h'(\sigma'(n)))$;
 $-\ h(n), h'(\sigma(n)) > 1$, then $\sigma(h(n) - 1) = h'(\sigma'(n)) - 1$, and so

$$\begin{aligned} &\mu_2(\sigma(h(n) - 1)) = \mu_2(h'(\sigma'(n)) - 1) && \text{(applying } \mu_2 \text{ on both sides)} \\ \iff\ &\mu_1(h(n) - 1) = \mu_2(h'(\sigma'(n)) - 1) && \text{(by hypothesis } \mu_1(n) = \mu_2(\sigma(n))) \\ \iff\ &\mu_1{}^+(h(n)) = \mu_2{}^+(h'(\sigma'(n))) && \text{(by definition of } \mu_1{}^+ \text{ and } \mu_2{}^+, \\ &&& \text{adding 1 to both members)} \end{aligned}$$

$\square$

Proof (of Proposition 4) **FinPos** is not small, but we recover smallness by skeletality. Monomorphisms are stable under pullbacks, and it can be easily verified that poset-reflection is preserved by pullbacks and by their mediating morphisms. $\square$

Proof (of Theorem 3) For accessibility: $\mathscr{P}_f$ is known to be accessible; $\mathcal{L}$ is accessible, because it can be regarded as a constant endofunctor on $\mathbf{Set}^{\mathbf{P}_m}$; Δ is accessible, because it has a right adjoint, namely the functor computing right Kan extensions along δ. The thesis follows from accessibility being preserved by composition and products.

As for preservation of weak-pullbacks, all the functors and operators we use preserve them. In particular Δ, having a left adjoint (computing left Kan extensions), preserves limits. $\square$

Proof (Proof of Proposition 5) The transition relation can be turned into a natural transformation in $\mathbf{Set}^{\mathbf{P}_m}$: (i) and (ii) guarantee that such relation obeys the naturality condition. The other direction clearly holds. $\square$

Proof (Proof of Proposition 6) Given a B-bisimulation (R, ρ), we shall show that $\{R(O)\}_{O \in |\mathbf{P}_m|}$ is a $\mathbf{P}_m$-indexed bisimulation. Take $O \in |\mathbf{P}_m|$ and a a pair $(p, q) \in R(O)$. Since it is also a B-coalgebra, by Proposition 5 there is a $\mathbf{P}_m$-ILTS $(R, \Longrightarrow_{\text{P0}})$ that has one transition of the form

$$O \rhd (p, q) \overset{l}{\Rightarrow}_{\text{P0}} O' \rhd (p', q')\ , (p', q') \in Rg'\ ,$$

for each transition of p. This gives the simulating transition of q required by (i) of Definition 17. Condition (ii) just amounts to say that R is a presheaf $\mathbf{Set}^{\mathbf{P}_m}$. $\square$

References

1. Jiří Adámek. Introduction to coalgebra. *Theory and Applications of Categories*, 14(8):157–199, 2005.
2. Jirí Adámek, Filippo Bonchi, Mathias Hülsbusch, Barbara König, Stefan Milius, and Alexandra Silva. A coalgebraic perspective on minimization and determinization. In *FoSSaCS*, pages 58–73, 2012.
3. Jiří Adámek and Jiří Rosický. *Locally Presentable and Accessible Categories*. Cambridge University Press, 1994.
4. Michele Boreale and Davide Sangiorgi. A fully abstract semantics for causality in the π-calculus. *Acta Inf.*, 35(5):353–400, 1998.
5. Roberto Bruni and Ugo Montanari. Dynamic connectors for concurrency. *Theor. Comput. Sci.*, 281(1-2):131–176, 2002.
6. Gian Luca Cattani and Peter Sewell. Models for name-passing processes: interleaving and causal. *Inf. Comput.*, 190(2):136–178, 2004.
7. Vincenzo Ciancia, Alexander Kurz, and Ugo Montanari. Families of symmetries as efficient models of resource binding. *Electr. Notes Theor. Comput. Sci.*, 264(2):63–81, 2010.
8. Vincenzo Ciancia and Ugo Montanari. Symmetries, local names and dynamic (de)-allocation of names. *Inf. Comput.*, 208(12):1349 – 1367, 2010.
9. Philippe Darondeau and Pierpaolo Degano. Causal trees: Interleaving + causality. In *Semantics of Systems of Concurrent Processes*, pages 239–255, 1990.
10. Gian Luigi Ferrari, Ugo Montanari, and Emilio Tuosto. Coalgebraic minimization of hd-automata for the π-calculus using polymorphic types. *Theor. Comput. Sci.*, 331(2-3):325–365, 2005.
11. Marcelo P. Fiore and Daniele Turi. Semantics of name and value passing. In *LICS*, pages 93–104, 2001.
12. Fabio Gadducci, Marino Miculan, and Ugo Montanari. About permutation algebras, (pre)sheaves and named sets. *Higher-Order and Symbolic Computation*, 19(2-3):283–304, 2006.
13. Paris C. Kanellakis and Scott A. Smolka. Ccs expressions, finite state processes, and three problems of equivalence. *Inf. Comput.*, 86(1):43–68, 1990.
14. Ugo Montanari and Marco Pistore. Minimal transition systems for history-preserving bisimulation. In *STACS*, pages 413–425, 1997.
15. Ugo Montanari and Marco Pistore. Structured coalgebras and minimal hd-automata for the π-calculus. *Theor. Comput. Sci.*, 340(3):539–576, 2005.
16. Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000.
17. Sam Staton. Name-passing process calculi: operational models and structural operational semantics. Technical Report 688, University of Cambridge, 2007.
18. Sam Staton. Relating coalgebraic notions of bisimulation. *Logical Methods in Computer Science*, 7(1), 2011.
19. James Worrell. Terminal sequences for accessible endofunctors. *Electr. Notes Theor. Comput. Sci.*, 19:24–38, 1999.