BRICS

Basic Research in Computer Science

BRICS RS-99-36 Cattani & Winskel: Presheaf Models for CCS-like Languages

Presheaf Models for CCS-like Languages

Gian Luca Cattani
Glynn Winskel

BRICS Report Series **RS-99-36**

ISSN 0909-0878 **November 1999**

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

http://www.brics.dk
ftp://ftp.brics.dk
This document in subdirectory RS/99/36/

Presheaf Models for **CCS**-like Languages

Gian Luca Cattani[*]
Computer Laboratory
University of Cambridge
England
`Luca.Cattani@cl.cam.ac.uk`

Glynn Winskel
BRICS[†]
University of Aarhus
Denmark
`gwinskel@brics.dk`

November 1999

Abstract

The aim of this paper is to harness the mathematical machinery around presheaves for the purposes of process calculi. Joyal, Nielsen and Winskel proposed a general definition of bisimulation from open maps. Here we show that open-map bisimulations within a range of presheaf models are congruences for a general process language, in which CCS and related languages are easily encoded. The results are then transferred to traditional models for processes. By first establishing the congruence results for presheaf models, abstract, general proofs of congruence properties can be provided and the awkwardness caused through traditional models not always possessing the cartesian liftings, used in the break-down of process operations, are side-stepped. The abstract results are applied to show that hereditary history-preserving bisimulation is a congruence for CCS-like languages to which is added a refinement operator on event structures as proposed by van Glabbeek and Goltz.

[*] This author is supported by EPSRC grant GR/L62290: Calculi for Interactive Systems: Theory and Experiment.

[†] **B**asic **R**esearch **in** **C**omputer **S**cience, a centre of the Danish National Research Foundation.

Contents

Introduction

Joyal and Moerdijk's open maps [18] led to an abstract view of bisimulation [19], applicable to models for concurrency once they are presented as categories, along the lines of the handbook chapter [29]. A central idea was to define bisimulation through a span of open maps and explore its consequences over models for concurrency ranging from "interleaving" models like transition systems to "independence" models like event structures, and later on Petri nets [23], in which concurrency or parallelism of actions is expressed by some relation of independence.

This paper takes up the suggestion of [19] to study presheaf models for concurrent computation. There are several reasons for doing this.

One reason is that, once one passes the barrier of unfamiliarity, presheaves form an intuitively appealing model of nondeterministic computation. Nondeterministic computations are identified with presheaves over a path category; objects of the path category specify the possible shapes of computation paths while its morphisms specify how one path can be extended to another. Because the presheaf category is equivalent to the category obtained by freely adjoining all colimits to the original path category, an individual presheaf is obtained, in effect, by identifying subpaths within a collection of computation paths.

As was argued in [19] presheaf models are promising generalisations of existing models. This is because well-known models like synchronisation trees and labelled event structures embed fully and faithfully into appropriate presheaf categories, and, for general reasons, presheaves support operations such as those coming from Kan extensions [21, 2]. One particular Kan extension, resulting in a functor between presheaves over pomsets, was advanced as a good candidate for an operation of refinement of the kind proposed for event structures. Here it is shown that this Kan extension acts, when restricted to presheaves associated with event structures, in the same way as the refinement operation on event structures proposed by Goltz and van Glabbeek in [12]. To highlight the gain of working at a more abstract level than is common in concurrency theory, we can often exploit an important general result [6, 3] that any colimit-preserving functor between presheaf categories automatically preserves open maps. In particular, this result specialises to show that the refinement, obtained as a Kan extension, preserves open maps and so open-map bisimulation.

One point of approaching models for concurrency as categories is that operations fundamental to process calculi appear automatically, as built out of universal constructions. An obvious question is whether these universal constructions preserve open maps and therefore bisimulation. Our approach here is to prove that operations on presheaves preserve open maps and then transfer these preservation properties to concrete models like synchronisation trees and event structures, through canonical embeddings using results such as Proposition 5.5.[1] Working with presheaves also avoids some obstructions to a treatment of weak bisimulation on independence models [10], though this topic is not dealt with here.

A more general, and probably the most important, motivation for presheaf models is the hope they give of making concurrency less separate a study. Through presheaf models we are trying to bring concurrency theory within domain theory, though with the proviso that this should be understood liberally enough to include generalisations of domain theory like those envisaged in "axiomatic domain theory" [24, 9, 11]. The use of presheaves as models for concurrency is not confined to languages expressible in **Proc**. For instance, following the work in [28] a treatment of presheaf categories as domains has been devised [7, 4] and applied

[1] The paper [8] shows that any "$\mathbb{P}$-factorisable functor" preserves open maps and so bisimulation. In contrast we aim to take advantage of the preservation properties of universal constructions—a strategy proposed in the conclusion of [19].

to the constructions of presheaf models for more sophisticated languages [5, 3]. Other applications, to nondeterministic dataflow, fairness and weak bisimulation, can be found in [16, 15, 10].

Specifically, this paper builds on the analysis of the handbook chapter [29], with open map bisimulation in mind, to axiomatise a presheaf based semantics of the "general purpose" process language **Proc** treated there. On top of the categorical structure which is needed for the semantics of **Proc**, phrased in terms of fibrations, we require the satisfaction of two conditions borrowed from categorical logic, the Beck-Chevalley condition and the Fröbenius Reciprocity law. These two conditions are important in proving the general congruence results of bisimulation that are expected. The analysis of the constructions involved in the semantics of **Proc** together with the corresponding preservation properties of open map bisimulation form the core of the paper. As an application of the general congruence results, we show that hereditary history-preserving bisimulation [19, 1] is a congruence for **Proc** and conclude by proving that the refinement of event structures proposed in [12] preserves hereditary history-preserving bisimulation.

The paper is organised as follows. In the first section we recall the basic definitions and results from concurrency and category theory that we will need in order to make the paper reasonably self-contained. Section 2 introduces the language **Proc** and discusses the appropriate notion of categorical model for it. Section 3 defines presheaf models for **Proc** and in the following section the relevant semantic constructions are discussed and proved to preserve bisimulation. Section 5 and 6 present the two applications to hereditary history-preserving bisimulation and event structures discussed above.

An earlier extended abstract of this paper appeared as [6].

1 Preliminaries

This section recalls three fundamental models for concurrent computation and provides the categorical background.

1.1 Traditional models

We focus on three traditional models for concurrency: transition systems, synchronisation trees and event structures (see [29] for more background). A *transition system* is a structure (S, i, L, tran) where

- S is a set of *states* with *initial state* i,

- L is a set of *labels*,

- *tran* $\subseteq S \times L \times S$ is the *transition relation.* Usually, a transition (s, a, s') is written as $s \xrightarrow{a} s'$.

Let $T_0 = (S_0, i_0, L_0, \mathit{tran}_0)$ and $T_1 = (S_1, i_1, L_1, \mathit{tran}_1)$ be transition systems. A *morphism* $f : T_0 \to T_1$ is a pair $f = (\sigma, \lambda)$ where

- $\sigma : S_0 \to S_1$, such that $\sigma(i_0) = i_1$, and
- $\lambda : L_0 \rightharpoonup L_1$, a partial function, which together satisfy

$$\begin{aligned} &(s, a, s') \in \mathit{tran}_0 \ \&\ \lambda(a) \text{ defined} \\ &\qquad\qquad \Longrightarrow\ (\sigma(s), \lambda(a), \sigma(s')) \in \mathit{tran}_1, \text{ and} \\ &(s, a, s') \in \mathit{tran}_0 \ \&\ \lambda(a) \text{ undefined } \Longrightarrow\ \sigma(s) = \sigma(s'). \end{aligned}$$

A *synchronisation tree* is a transition system whose transition graph has the form of a tree with root the initial state.

Definition 1.1 (The Categories $\mathcal{TS}$ and $\mathcal{ST}$) *Define $\mathcal{TS}$ to be the category of objects transition systems and arrows transition systems morphisms. The composition of arrows is defined componentwise.*

Define $\mathcal{ST}$ to be the full subcategory of $\mathcal{TS}$ of Synchronisation Trees.

Transition systems and synchronisation trees are often called "interleaving models" because they represent parallel/concurrent composition by nondeterministically interleaving the actions of processes. In contrast, event structures represent a class of "independence models" (among them Petri nets) in which concurrency is represented directly as a form of causal independence.

Define a *(labelled) event structure* to be a structure $(E, \leq, \mathit{Con}, l)$ consisting of a set E, of *events* which are partially ordered by $\leq$, the *causal dependency relation*, a *consistency relation* Con which is a non-empty family of finite subsets of events, and a *labelling function* $l : E \to L$, which satisfy

$$\begin{aligned} &\{e' \mid e' \leq e\} \text{ is finite,} \\ &\{e\} \in \mathit{Con}, \\ &Y \subseteq X \in \mathit{Con} \implies Y \in \mathit{Con}, \\ &X \in \mathit{Con} \ \&\ e \leq e' \in X \implies X \cup \{e\} \in \mathit{Con}, \end{aligned}$$

for all events e, e' and their subsets X, Y. Two events $e, e' \in E$ are said to be *concurrent* (causally independent) iff

$$(e \not\leq e' \ \&\ e' \not\leq e \ \&\ \{e, e'\} \in \mathit{Con}).$$

A set, x, of events in E is said to be a *configuration* if it is *downwards-closed:* $\forall e, e'.\ e' \leq e \in x \implies e' \in x$, and

consistent: $\forall X.\ X$ finite & $X \subseteq x \implies X \in Con$.

A *morphism* of event structures consists of

$$(\eta, \lambda) : E \to E',$$

where $E = (E, \leq, Con, l), E' = (E', \leq', Con', l')$ are event structures, $\eta : E \rightharpoonup E'$ is a partial function on events, $\lambda : L \rightharpoonup L'$ is a partial function on labelling sets such that

(i) $l' \circ \eta = \lambda \circ l$,

(ii) If x is a configuration of E, then ηx is a configuration of E' and if for $e_1, e_2 \in x$ their images are both defined with $\eta(e_1) = \eta(e_2)$, then $e_1 = e_2$.

Definition 1.2 (The Category of Event Structures) *Define $\mathcal{ES}$ to be the category of objects event structures and arrows event structures morphisms. The composition of arrows is defined componentwise.*

The definition of morphism on event structures is given rather abruptly—see [29] for motivation. The categories $\mathcal{TS}, \mathcal{ST}$ and $\mathcal{ES}$ are related by coreflections: the inclusion functor $\mathcal{ST} \hookrightarrow \mathcal{TS}$ has a right adjoint unfolding transition systems to trees; the functor $\mathcal{ST} \to \mathcal{ES}$ identifying a synchronisation tree with an event structure has a right adjoint serialising an event structure to a synchronisation tree.

The categories of models described above can be fibred with respect to their labelling sets.

Definition 1.3 *For $\mathcal{M} \in \{\mathcal{ST}, \mathcal{TS}, \mathcal{ES}\}$ and L a set of labels, define $\mathcal{M}_L$ to be the subcategory of $\mathcal{M}$ of those objects labelled in L and morphisms which have the identity on L as relabelling part.*

1.2 Bisimulation from open maps

We now describe the characterisation of bisimulation via open maps proposed in [19], to which we refer for a more detailed discussion. We need to fix on an idea of the computation paths in our model. For instance a (computation) path of a transition system with labelling set L is reasonably taken to be a finite sequence of transitions that the transition system can perform. It has the shape of a string of labels in L.

Definition 1.4 (Finite strings regarded as a category) *Let L be a set. Define L^+ to be the partial order of finite non-empty strings over L regarded as a category. It is convenient to identify strings in L^+ with the equivalent subcategory of $\mathcal{ST}_L$ consisting of those special synchronisation trees consisting of a finite single branch of nonzero length.*
Define L^* to be the extension of L^+ to all finite strings.

To take account of the added independence structure of event structures, the shape of their computation paths is taken to be a finite *pomset* [25].

Definition 1.5 (Labelled pomsets) *Let L be a set. The category $\mathbf{Pom}_L$ is taken to be the subcategory of $\mathcal{ES}_L$, for a labelling set L, consisting of those finite non-empty event structures in which all subsets of events are in the consistency relation. In other words the objects P of $\mathbf{Pom}_L$ are triples $P = (P, \leq, l)$ where P is a finite non-empty set, $\leq$ is a partial order on P and $l : P \to L$ is a function. A morphism $f : P \to Q$ in $\mathbf{Pom}_L$ is given by a injective function that preserve the labelling and send downward closed sets of P to downward closed sets of Q.*
Define $(\mathbf{Pom}_L)_\perp$ to be the extension of $\mathbf{Pom}_L$ to include the empty pomset as well.

Both the operation of adjoining the empty string, taking L^+ to L^*, and the empty pomset to $\mathbf{Pom}_L$ are special cases of the "lifting" construction on categories which will meet in Definition 1.8)—this accounts for the notation in the definition above.

We can obtain a general definition of bisimulation from *open maps.* Roughly speaking open maps are morphisms with the property that any extension of a computation path in the range can be matched by an extension in its domain.

Definition 1.6 ($\mathbb{P}$-open maps) *Assume a category of models $\mathcal{M}$ and a choice of path category, a subcategory $\mathbb{P} \hookrightarrow \mathcal{M}$ consisting of path objects together with morphisms expressing how they can be extended. Let $f : M \to M'$ be an arrow in $\mathcal{M}$. We say that f is a* $\mathbb{P}$-open map *if, whenever, for $m : P \to Q$ a morphism in $\mathbb{P}$, a "square"*

$$\begin{tikzcd}
P \arrow[r, "p"] \arrow[d, "m"'] & M \arrow[d, "f"] \\
Q \arrow[r, "q"] & M'
\end{tikzcd}$$

in $\mathcal{M}$ commutes, i.e., *$q \circ m = f \circ p$, meaning the path $f \circ p$ in M' can be extended via m to a path q in M', then there is a (not necessarily unique) morphism p' such that in the diagram*

$$\begin{tikzcd}
P \arrow[r, "p"] \arrow[d, "m"'] & M \arrow[d, "f"] \\
Q \arrow[r, "q"] \arrow[ur, "p'"] & M'
\end{tikzcd}$$

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* M *which matches* q*.*

Two objects M_1, M_2 *of* $\mathcal{M}$ *are said to be* $\mathbb{P}$-bisimilar *iff there is a* span *of* $\mathbb{P}$*-open morphisms* f_1, f_2*:*

$$\begin{tikzcd} & M \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\ M_1 & & M_2 \ . \end{tikzcd}$$

In the case of traditional models we obtain known equivalences. In $\mathcal{ST}_L$, L^*-bisimulation coincides with Park and Milner's strong bisimulation; for event structures $\mathcal{ES}_L$, $(\mathbf{Pom}_L)_\perp$-bisimulation coincides with hereditary history-preserving bisimulation due to Bednarczyk refining ideas of van Glabbeek and Goltz, Rabinovitch and Traktenbrot [1, 12, 26].

The following easy-to-prove preservation property will be used later.

Proposition 1.7 *Suppose* $\mathcal{M}$ *has products. Let* $f_1 : M_1 \to N_1$ *and* $f_2 : M_2 \to N_2$ *be* $\mathbb{P}$*-open maps. Then* $f_1 \times f_2 : M_1 \times M_2 \to N_1 \times N_2$ *is* $\mathbb{P}$*-open.*

1.3 Presheaves as non-deterministic processes

Presheaf categories are the central mathematical notion around which this paper is built. Let $\mathbb{P}$ be a category. We shall call $\mathbb{P}$ a *path category* because we are thinking of its objects as specifying computation path shapes and its morphisms as path extensions. The category of *presheaves over* $\mathbb{P}$, often denoted by $\widehat{\mathbb{P}}$ or by $\mathbf{Set}^{\mathbb{P}^{op}}$, is the category whose objects are contravariant functors from $\mathbb{P}$ to $\mathbf{Set}$ (the category of sets and functions) and whose arrows are the natural transformations between such functors. The category $\widehat{\mathbb{P}}$ is the "free cocompletion" of $\mathbb{P}$, in the sense that it can be obtained to within equivalence of categories, by freely adding all possible colimits of diagrams in $\mathbb{P}$. The category $\widehat{\mathbb{P}}$ is very rich in structure and is well known to be an example of a topos (for more details on toposes and functor categories, see [22]).

A category of presheaves, $\widehat{\mathbb{P}}$, is accompanied by a functor, the *Yoneda embedding*, $\mathrm{y}_{\mathbb{P}} : \mathbb{P} \to \widehat{\mathbb{P}}$, which fully and faithfully embeds $\mathbb{P}$ in the category of presheaves. Given any object P of $\mathbb{P}$, the presheaf $\mathrm{y}_{\mathbb{P}}(P)$ is the contravariant hom-functor, $\mathbb{P}[-, P]$, which to any Q, an object of $\mathbb{P}$, associates the set $\mathbb{P}[Q, P]$ of arrows of $\mathbb{P}$ from Q to P, and to any arrow $f : Q \to Q'$ associates the function that by precomposition with f sends any arrow $g : Q' \to P$ to $g \circ f : Q \to P$.

Via the Yoneda embedding we can regard $\mathbb{P}$ as essentially a full subcategory of $\widehat{\mathbb{P}}$. We have a situation fitting that needed for defining open maps and bisimulation in $\widehat{\mathbb{P}}$, along the lines of Section 1.2. The Yoneda Lemma (see *e.g.* [21]) provides a natural bijection between $\widehat{\mathbb{P}}[y_{\mathbb{P}}(P), F]$ and $F(P)$. This justifies the intuition that a presheaf $F : \mathbb{P}^{\mathrm{op}} \to \mathbf{Set}$ can be thought of as specifying for a typical path object P the set, $F(P)$, of computation paths of shape P. The presheaf F acts on a morphism $m : P \to Q$ in $\mathbb{P}$ to give a function $F(m) : F(Q) \to F(P)$ saying how Q-paths restrict to P-paths.

It turns out that to rule out the unfortunate case of the arrows from the initial, always empty, presheaf to any other presheaf being open, it is more convenient to define open map bisimulation for presheaves with respect to a *strict* extension of the Yoneda embedding.

Definition 1.8 *If $\mathbb{P}$ is a category, define $\mathbb{P}_\perp$ to be the category $\mathbb{P}$ to which a new initial object, $\perp$, has been added. Define $\mathrm{y}^\circ_{\mathbb{P}} : \mathbb{P}_\perp \hookrightarrow \widehat{\mathbb{P}}$ to be the strict,* i.e., *initial-object preserving, extension of the Yoneda embedding $\mathrm{y}_{\mathbb{P}}$.*

As we shall see, $\mathrm{y}^\circ_{\mathbb{P}} : \mathbb{P}_\perp \hookrightarrow \widehat{\mathbb{P}}$ is the universal arrow associated with another characterisation of the presheaf category $\widehat{\mathbb{P}}$, as the free connected-colimit preserving completion of $\mathbb{P}_\perp$.[2]

Definition 1.9 (Open maps for presheaves) *If $\mathbb{P}$ is a small category and $f : X \to Y$ is an arrow between presheaves over $\mathbb{P}$, define f to be $\mathbb{P}$-open (or simply open when no confusion arises) if it is open according to Definition 1.6 with respect to the embedding $\mathrm{y}^\circ_{\mathbb{P}}$.*

By considering open maps with respect to $\mathrm{y}^\circ_{\mathbb{P}}$ rather than $\mathrm{y}_{\mathbb{P}}$ we obtain that open maps are necessarily epimorphic (as are L^*- and $(\mathbf{Pom}_L)_\perp$-open maps in contrast to L^+- and $\mathbf{Pom}_L$-open maps for transition systems and event structures, respectively).

Proposition 1.10 *An arrow between presheaves over a category $\mathbb{P}$ is open if and only if it is epimorphic (*i.e.*, pointwise a surjective function) and open with respect to the Yoneda embedding, $\mathrm{y}_{\mathbb{P}}$.*

A model, like a transition system or a labelled event structure, gives rise to a presheaf. For a category of models $\mathcal{M}$ and a choice of path category forming a subcategory $\mathbb{P} \hookrightarrow \mathcal{M}$, there is a *canonical functor* from the category of models $\mathcal{M}$ to the category of presheaves $\widehat{\mathbb{P}}$. The functor, $c_{\mathcal{M}} : \mathcal{M} \to \widehat{\mathbb{P}}$, takes an object M of $\mathcal{M}$ to the presheaf $\mathcal{M}[-, M]$—more intuitively, it takes the model M to the presheaf which for each path object P yields the set of paths $\mathcal{M}[P, M]$ from P into M. The canonical functor takes a morphism $f : M \to M'$ in $\mathcal{M}$ to the natural transformation, $\mathcal{M}[-, f] : \mathcal{M}[-, M] \to \mathcal{M}[-, M]$, whose component at an object

[2] In earlier work [19, 6], we have made use of *rooted* presheaves of a category $\mathbb{P}_\perp$. A rooted presheaf is a presheaf X over $\mathbb{P}_\perp$ in which the set $X(\perp)$ is a singleton. Note that the subcategory of rooted presheaves over $\mathbb{P}_\perp$ is isomorphic to the category of presheaves $\widehat{\mathbb{P}}$, and that this is the connected-colimit completion of $\mathbb{P}_\perp$.

P of $\mathbb{P}$ is the function $\mathcal{M}[P, M] \to \mathcal{M}[P, M']$ taking p to $f \circ p$—a path $p : P \to M$ in M is taken to a path $f \circ p : P \to M$ in M. More generally for $F : \mathbb{P} \to \mathcal{M}$ any functor, one can define $c_F : \mathcal{M} \to \widehat{\mathbb{P}}$ to be $c_F(M) \overset{\text{def}}{=} \mathcal{M}[F(-), M]$ on objects and to act by composition on morphisms.

The canonical functors from synchronisation trees $\mathcal{ST}_L$ to presheaves $\widehat{L^+}$ and from event structures $\mathcal{ES}_L$ to $\widehat{\mathbf{Pom}_L}$ are full and faithful, so *canonical embeddings*. Generally, a canonical functor $c_{\mathcal{M}}$ is full and faithful whenever the path category is dense in the category of models $\mathcal{M}$ (see MacLane [21], P. 243), as remarked in [19]. Because such canonical embeddings are also dense they preserve all existing limits [30]. On the other hand, we cannot expect $c_{\mathcal{M}}$ to preserve general colimits in $\mathcal{M}$. The following proposition asserts that colimits of certain diagrams are preserved (we shall need it in Section 6).

Proposition 1.11 *Let $\mathcal{M}$ be a category, $\mathbb{P}$ be a category with a functor $F : \mathbb{P} \to \mathcal{M}$. Let $\Delta : \mathbb{D} \to \mathcal{M}$ be another functor from a small category $\mathbb{D}$ satisfying the following property of "density with respect to F": If $(M, \delta_D : \Delta(D) \to M)$ is a colimiting cone for Δ, then for any $P \in \mid \mathbb{P} \mid$ and $p : F(P) \to M$, there exists a $D \in \mid \mathbb{D} \mid$ and $d : F(p) \to \Delta(d)$ such that:*

- *$p = \delta_D d$.*
- *For any other factorisation*

$$\begin{tikzcd}
F(P) \arrow[rr, "p"] \arrow[dr, "d'"'] & & M \\
& \Delta(D') \arrow[ur, "\delta_{D'}"'] &
\end{tikzcd}\ ,$$

there exists $m : D \to D'$ such that

$$\Delta(m)d = d' \qquad \textit{and} \qquad \delta_{D'}\Delta(m) = \delta_D\ .$$

Then $c_F(M) \cong \operatorname{colim} c_F\Delta$.

Remark: Proposition 1.11 above can be made into an "if and only if" statement if we replace the condition on m by one saying that any two factorisations are connected by a chain of spans

$$D = D_0 \xleftarrow{m_1} D_1 \xrightarrow{m_2} D_2 \leftarrow \cdots \xrightarrow{m_n} D_n = D'$$

in $\mathbb{D}$ with $p_i : F(P_i) \to \Delta(D_i)$ such that:

$$\begin{array}{rclrcl}
\delta_{D_{i-1}} m_i & = & \delta_{D_i} \text{ (for } i \text{ odd)} & \delta_{D_i} m_i & = & \delta_{D_{i-1}} \text{ (for } i \text{ even)} \\
m_i p_i & = & p_{i-1} \text{ (for } i \text{ odd)} & m_i P_{i-1} & = & p_i \text{ (for } i \text{ even)}\ .
\end{array}$$

Open map bisimulation is clearly preserved whenever the canonical functors are full and faithful and is reflected too in the case of synchronisation trees and event structures.

Proposition 1.12 (Joyal-Nielsen-Winskel)

(i) *Two synchronisation trees, over labelling set L, are L^*-bisimilar (*i.e.*strong bisimilar) iff their corresponding presheaves, under the canonical embedding, are related by a span of L^+-open maps.*

(ii) *Two event structures, over labelling set L, are $(\mathbf{Pom}_L)_\perp$-bisimilar (equivalently, hereditary history-preserving bisimilar) iff their corresponding presheaves, under the canonical embedding, are related by a span of $\mathbf{Pom}_L$-open maps.*

Having established the link between categories of models and categories of presheaves over appropriate path categories, we can look for general constructions on presheaves, useful in modelling processes, that preserve openness and hence bisimulation. Left Kan extensions will be among such constructions. Before we introduce this powerful operation we mention a simple property, straightforward to verify, of the canonical embeddings $c_{\mathcal{ST}_L}$ of synchronisation trees and $c_{\mathcal{ES}_L}$ of event structures, that we shall need in Section 5.

Proposition 1.13 *The canonical embeddings, $c_{\mathcal{ST}_L}$ and $c_{\mathcal{ES}_L}$ preserve coproducts and initial objects.*

1.4 Left Kan extensions

We introduce the notion of Left Kan extension, a construction which we use extensively.

Definition 1.14 (Left Kan Extensions) *If $\mathcal{C} \xleftarrow{G} \mathcal{A} \xrightarrow{F} \mathcal{B}$ are functors, one says that a pair (H, α) is a* left Kan extension of G along F *if*

- *$H : \mathcal{B} \to \mathcal{C}$ is a functor*
- *$\alpha : G \Rightarrow HF$ is a natural transformation satisfying the following universal property:*

  for every other pair (K, β) with $\beta : G \Rightarrow KF$ there exists a unique $\gamma : H \Rightarrow K$ such that $\beta = \gamma_F \cdot \alpha$.

By the usual abuse of language we will often call the functor H the left Kan extension of G along F and write $\mathrm{Lan}_F(G)$ *to indicate it.*

Note that the triangle

$$\begin{tikzcd}
\mathcal{A} \arrow[r, "F"] \arrow[dr, "G"'] & \mathcal{B} \arrow[d, "\mathrm{Lan}_F(G)"] \\
& \mathcal{C}
\end{tikzcd}$$

need not commute, not even up to natural isomorphism. Still, this happens in many cases of interest.

Proposition 1.15 *If F is full and faithful and $(\mathrm{Lan}_F(G), \alpha)$ exists then α is a natural isomorphism.*

If $\mathcal{C}$ is cocomplete and $\mathcal{A}$ is essentially small, then $\mathrm{Lan}_F(G)$ always exists for any F and G and can be computed "pointwise" (see [2]) as a colimit. Left Kan extensions compose in the sense that if (H, α) is the left Kan extension of G along F for $\mathcal{C} \xleftarrow{G} \mathcal{A} \xrightarrow{F} \mathcal{B}$ and (K, β) is the left Kan extension of H along F', for $\mathcal{C} \xleftarrow{H} \mathcal{B} \xrightarrow{F'} \mathcal{D}$, then $(K, \beta_F \cdot \alpha)$ is the left Kan extension of G along $F'F$:

$$\mathrm{Lan}_{F'F}(G) \cong \mathrm{Lan}_{F'}(\mathrm{Lan}_F(G)) \ .$$

Let $\mathbb{P}$ be a category. We mentioned earlier that $\widehat{\mathbb{P}}$ is the free colimit completion of $\mathbb{P}$. In more detail:

Proposition 1.16 *The Yoneda embedding $\mathrm{y}_{\mathbb{P}} : \mathbb{P} \to \widehat{\mathbb{P}}$ satisfies the universal property that for any functor $F : \mathbb{P} \to \mathcal{E}$, where $\mathcal{E}$ is a cocomplete category, there is a colimit-preserving functor $G : \widehat{\mathbb{P}} \to \mathcal{E}$, unique to within isomorphism, such that $F \cong G \circ \mathrm{y}_{\mathbb{P}}$:*

$$\begin{tikzcd}
\mathbb{P} \arrow[r, "\mathrm{y}_{\mathbb{P}}"] \arrow[dr, "F"'] & \widehat{\mathbb{P}} \arrow[d, "G"] \\
& \mathcal{E}
\end{tikzcd}$$

To within isomorphism, the functor G is given as the left Kan extension $\mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(F)$. Conversely, if $G : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ is a colimit-preserving functor then, to within isomorphism, G is $\mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(G \circ \mathrm{y}_{\mathbb{P}})$.

In fact, $\mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(F)$ has right adjoint a functor taking Y in $\mathcal{E}$ to the presheaf $\mathcal{E}[F(\text{-}), Y]$; the right adjoint is also a left Kan extension, being $\mathrm{Lan}_F(\mathrm{y}_{\mathbb{P}})$ to within isomorphism.

A key result that we use in this paper is the following theorem which entails that the left Kan extensions described above preserve open maps, a powerful tool in showing operations preserve bisimulation:

Theorem 1.17 *Let $F : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ be a colimit preserving functor and let f be a $\mathbb{P}$-open map, then $F(f)$ is a $\mathbb{Q}$-open map.*

A proof of the above theorem appears in [3]. We shall not be interested only in colimit-preserving functors and in [3] a similar preservation property for a larger class of functors is also proved.

Theorem 1.18 *Let $F : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ be a connected-colimit preserving functor,* i.e., *a functor which preserves colimits of connected diagrams, and let f be a $\mathbb{P}$-open map, then $F(f)$ is a $\mathbb{Q}$-open map.*

The operations that we use for modelling process constructors on presheaves all fall within the class of connected-colimit preserving functors. This will enable us to establish that open-map bisimulation in presheaf models is a congruence for a general process language, once and for all.

We present the connection between the strict extension of the Yoneda embedding $\mathrm{y}^\circ_{\mathbb{P}}$ and connected-colimit preserving functors.

Proposition 1.19 (See [3]) *Let $\mathbb{P}$ be a small category. With respect to the embedding $\mathrm{y}^\circ_{\mathbb{P}} : \mathbb{P}_\perp \to \widehat{\mathbb{P}}$, the category $\widehat{\mathbb{P}}$ is the free connected-colimit completion of $\mathbb{P}_\perp$. Moreover if $F : \mathbb{P}_\perp \to \mathcal{E}$ is a functor, with $\mathcal{E}$ a category with connected-colimits, then the universal extension of F is $\mathrm{Lan}_{\mathrm{y}^\circ_{\mathbb{P}}}(F) : \widehat{\mathbb{P}} \to \mathcal{E}$.*

Recall the Kan extensions arising in an important special case. A functor $F : \mathbb{P} \to \mathbb{Q}$ gives rise to a triple of adjoint functors $F_! \dashv F^* \dashv F_*$ (see [22]) that can be described as

$$F_! = \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(\mathrm{y}_{\mathbb{Q}}F), \quad F^* = \mathrm{Lan}_{\mathrm{y}_{\mathbb{Q}}F}(\mathrm{y}_{\mathbb{P}}), \quad F_* = \mathrm{Lan}_{F^*\mathrm{y}_{\mathbb{Q}}}(\mathrm{y}_{\mathbb{Q}}) \ .$$

Instantiating this to $F = I : \mathbb{P} \hookrightarrow \mathbb{P}_\perp$, the obvious embedding of $\mathbb{P}$ in $\mathbb{P}_\perp$, we deduce the following.

Proposition 1.20 *If $I : \mathbb{P} \hookrightarrow \mathbb{P}_\perp$ is the obvious embedding of $\mathbb{P}$ into $\mathbb{P}_\perp$, then I_* preserves connected-colimits.*

Proof: As we wrote $I_* = \mathrm{Lan}_{I^*\mathrm{y}_{\mathbb{P}_\perp}}(\mathrm{y}_{\mathbb{P}})$. A straightforward calculation shows that $I^*\mathrm{y}_{\mathbb{P}_\perp} = \mathrm{y}^\circ_{\mathbb{P}}$, hence by Proposition 1.19, I_* is connected-colimit preserving. □

Notice that I_* can be defined more concretely as acting on a presheaf X in the following way:

$$I_*(X)(P) = \begin{cases} X(P) & \text{if } P \neq \perp \\ \{*\} & \text{otherwise.} \end{cases}$$

Later, we shall make great use of the following definition.

Definition 1.21 *Let $F : \mathbb{P}_\perp \to \mathbb{Q}_\perp$ be a functor that preserves the initial object. Consider the composite functor $\mathbb{P} \xrightarrow{I_{\mathbb{P}}} \mathbb{P}_\perp \xrightarrow{F} \mathbb{Q}_\perp \xrightarrow{\mathrm{y}^\circ_{\mathbb{Q}}} \widehat{\mathbb{Q}}$. Define $F^l \overset{\mathrm{def}}{=} \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(\mathrm{y}^\circ_{\mathbb{Q}} F I_{\mathbb{P}}) : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ and $F^r \overset{\mathrm{def}}{=} \mathrm{Lan}_{\mathrm{y}^\circ_{\mathbb{Q}} F I_{\mathbb{P}}}(\mathrm{y}_{\mathbb{P}})$.*

Proposition 1.22 *With respect to Definition 1.21 above we have that:*

1. *$F^l \dashv F^r$.*

2. *F^r is connected-colimit preserving.*

3. *F^l is colimit-preserving and $F^l \cong I^*_{\mathbb{Q}}(F I_{\mathbb{P}})_! \cong I^*_{\mathbb{Q}} F_! I_{\mathbb{P},!}$, where $I_{\mathbb{Q}} : \mathbb{Q} \hookrightarrow \mathbb{Q}_\perp$.*

Proof:

1. By Proposition 1.16 (with F instantiated to $\mathrm{y}^\circ_{\mathbb{Q}} F I_{\mathbb{P}}$).

2. Since left Kan extensions compose, we have that $F^r \overset{\mathrm{def}}{=} \mathrm{Lan}_{\mathrm{y}^\circ_{\mathbb{Q}} F I_{\mathbb{P}}}(\mathrm{y}_{\mathbb{P}}) \cong \mathrm{Lan}_{\mathrm{y}^\circ_{\mathbb{Q}}}(\mathrm{Lan}_{F I_{\mathbb{P}}}(\mathrm{y}_{\mathbb{P}}))$ and hence, by Proposition 1.19, we can deduce that F^r preserves connected colimits.

3. As remarked earlier, left Kan extensions with respect to the Yoneda embedding correspond to colimit preserving functors, to within isomorphism. Recall from the proof of Proposition 1.20, that $\mathrm{y}^\circ_{\mathbb{Q}} = I^*_{\mathbb{Q}} \mathrm{y}_{\mathbb{Q}_\perp}$. Notice moreover that colimit-preserving functors, such as $I^*_{\mathbb{Q}}$, preserve left Kan extensions (see [2]). Thus we can establish the following chain of isomorphisms:

$$\begin{aligned} F^l \overset{\mathrm{def}}{=} \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(\mathrm{y}^\circ_{\mathbb{Q}} F I_{\mathbb{P}}) \cong \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(I^*_{\mathbb{Q}} \mathrm{y}_{\mathbb{Q}_\perp} F I_{\mathbb{P}}) &\cong I^*_{\mathbb{Q}} \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(\mathrm{y}_{\mathbb{Q}_\perp} F I_{\mathbb{P}}) \\ &\overset{\mathrm{def}}{=} I^*_{\mathbb{Q}}(F I_{\mathbb{P}})_! \cong I^*_{\mathbb{Q}} F_! I_{\mathbb{P},!} \ . \end{aligned}$$

□

1.5 Fibred categories

Indexing structure plays a fundamental role in the categorical analysis of models for concurrency [29, 27]. In the context of categorical models for **CCS**-like languages we will consider presheaf categories indexed by a category of labelling sets. There is a tight correspondence between indexed categories and fibrations; the former represent the class of fibrations for which a definite (coherent) choice of a cleavage has been made. We introduce in this section the basic terminology of fibred category theory[3] together with pointers to the related notion of *elementary existential doctrine* [20] of which the presheaf models of Section 3 will be an example.

[3] A more detailed introduction to fibrations can be found in, *e.g.*, [17].

Definition 1.23 (Cartesian arrows) *Let $\pi : \mathcal{E} \to \mathcal{B}$ be a functor. An arrow in $\mathcal{E}$, $f : e' \to e$ is* cartesian *(with respect to π) if for every other arrow $g : e'' \to e$ such that $\pi(g) = \beta\alpha$ with $\beta = \pi(f)$, there exists a unique $h : e'' \to e'$ with $g = fh$ and $\pi(h) = \alpha$:*

$$\begin{tikzcd}
e'' \arrow[drr, bend left, "g"] \arrow[dr, dotted, "\exists ! h"'] & & \\
& e' \arrow[r, "f"'] & e
\end{tikzcd}$$

$$\begin{tikzcd}
\pi(e'') \arrow[r, "\alpha"] & \pi(e') \arrow[r, "\beta"] & \pi(e) \quad .
\end{tikzcd}$$

Definition 1.24 (Fibrations) *A functor $\pi : \mathcal{E} \to \mathcal{B}$ is a* fibration *if for every $\beta : b' \to b$ in $\mathcal{B}$ and $e \in |\mathcal{E}|$ such that $\pi(e) = b$, there exists a cartesian arrow, f, of codomain e such that $\pi(f) = \beta$. The arrow f is called a* cartesian lifting of e with respect to β*.*

Definition 1.25 *If $\pi : \mathcal{E} \to \mathcal{B}$ is a functor, an arrow $f : e' \to e$ of $\mathcal{E}$ is said to be* vertical *if $\pi(f) = 1_e$.*

Definition 1.26 *If $\pi : \mathcal{E} \to \mathcal{B}$ is a functor and b an object of $\mathcal{B}$. Define the fibre over b (with respect to π) to be the subcategory $\mathcal{E}_b$ of $\mathcal{E}$ of those objects e and arrows f such that $\pi(e) = b$ and $\pi(f) = 1_b$.*

If π is a fibration then a choice of cartesian arrows induces *cartesian lifting functors* between the fibres:

Proposition 1.27 (Cartesian lifting functors) *Let $\pi : \mathcal{E} \to \mathcal{B}$ a fibration. Let $\beta : b' \to b$ be an arrow in $\mathcal{B}$. For every object $e \in |\mathcal{E}_b|$ let $\beta^*_e : \beta^*(e) \to e$ be a chosen cartesian lifting of e with respect to β. This choice induces the following cartesian lifting functors $\beta^* : \mathcal{E}_b \to \mathcal{E}_{b'}$:*

- **On objects** $e \mapsto \beta^*(e)$ *as chosen above*
- **On arrows** $(f : \overline{e} \to e) \mapsto \beta^*(f)$ *that is defined to be the unique arrow such that the following square commutes:*

$$\begin{tikzcd}
\beta^*(\overline{e}) \arrow[r, "\beta^*_{\overline{e}}"] \arrow[d, "\beta^*(f)"'] & \overline{e} \arrow[d, "f"] \\
\beta^*(e) \arrow[r, "\beta^*_e"'] & e.
\end{tikzcd}$$

A choice of cartesian arrows for a fibration is called a *cleavage* and a fibration with a chosen cleavage is called a *cloven* fibration. If the choice of the cleavage is functorial, *i.e.*, $(1_b)^*_e = 1_e$ and for $b'' \xrightarrow{\alpha} b' \xrightarrow{\beta} b$, $(\beta\alpha)^*_e = \alpha^*_{\beta^*_e}$, then the fibration is said to be *split*.

We will make extensive use of the dual notion of cofibration:

Definition 1.28 (Cofibrations and Bifibrations) *A functor $\pi : \mathcal{E} \to \mathcal{B}$ is a* cofibration *if the dual functor $\pi^{\mathrm{op}} : \mathcal{E}^{\mathrm{op}} \to \mathcal{B}^{\mathrm{op}}$ is a fibration. A functor that is both a fibration and a cofibration is called a* bifibration*.*

Dually, one talks of cocartesian arrows, cocartesian liftings, cocartesian lifting functors and (functorial) cocleavages.

1.6 The Grothendieck construction

Cloven fibrations are equivalent to indexed categories.[4] In fact any indexed category gives rise (via a construction due to Grothendieck [13]) to a cloven fibration and vice versa any cloven fibration induces an indexed category.

Definition 1.29 (Indexed categories) *Let $\mathcal{B}$ be any category. A $\mathcal{B}$-indexed category in $\mathcal{CAT}$ is given by a pseudo-functor $F : \mathcal{B}^{\mathrm{op}} \to \mathcal{CAT}$, that is F associates to each object of b, a category $F(b)$, to any arrow $\beta : b' \to b$ a functor $\beta^* : F(b) \to F(b')$ with natural isomorphisms, $\phi_b : 1_{F(b)} \xrightarrow{\simeq} (1_b)^*$ and $\phi_{(\alpha,\beta)} : \alpha^*\beta^* \xrightarrow{\simeq} (\beta\alpha)^*$ for any $b \in\mid \mathcal{B} \mid$ and for any two arrows $b'' \xrightarrow{\alpha} b' \xrightarrow{\beta} b$ of $\mathcal{B}$ satisfying the coherence conditions given by commutativity of the following diagrams:*

$$\begin{tikzcd}
\alpha^*(1_b)^* \arrow[dr, "\phi_{\alpha,1_b}"'] & \alpha^* \arrow[l, "\alpha^*\phi_b"'] \arrow[r, "(\phi_b)_{\alpha^*}"] \arrow[d, "1_{\alpha^*}"] & (1_{b'})^*\alpha^* \arrow[dl, "\phi_{1_{b'},\alpha}"] \\
 & \alpha^* &
\end{tikzcd}$$

$$\begin{tikzcd}
\alpha^*\beta^*\gamma^* \arrow[r, "(\phi_{\alpha,\beta})_{\gamma^*}"] \arrow[d, "\alpha^*\phi_{\beta,\gamma}"'] & (\beta\alpha)^*\gamma^* \arrow[d, "\phi_{\beta\alpha,\gamma}"] \\
\alpha^*(\gamma\beta)^* \arrow[r, "\phi_{\alpha,\gamma\beta}"'] & (\gamma\beta\alpha)^*\ ,
\end{tikzcd}$$

with γ being another arrow of $\mathcal{B}$, $\gamma : b \to b'''$.

Definition 1.30 (Grothendieck construction) *Given a $\mathcal{B}$-indexed category $F : \mathcal{B}^{\mathrm{op}} \to \mathcal{CAT}$, define the following category $Groth(F)$:*

[4] See for example [14] for a precise account of this statement.

- **Objects:** *Pairs $\langle c, b\rangle$, with $c \in\mid F(b)\mid$ and $b \in\mid\mathcal{B}\mid$.*
- **Arrows:** *A pair $\langle f, \beta\rangle$ is an arrow from $\langle c', b'\rangle$ to $\langle c, b\rangle$ if $\beta : b' \to b$ is an arrow in $\mathcal{B}$ and $f : c' \to F(\beta)(c)$ is an arrow in $F(b')$. If $\langle f, \alpha\rangle : \langle c'', b''\rangle \to \langle c', b'\rangle$ and $\langle g, \beta\rangle : \langle c', b'\rangle \to \langle c, b\rangle$ then their composite is the pair $\langle h, \beta\alpha\rangle$ where h is the following arrow*

  $$c \xrightarrow{f} F(\alpha)c' \xrightarrow{F(\alpha)g} F(\alpha)F(\beta)c'' \xrightarrow{\phi_{\alpha,\beta}} F(\beta\alpha)c'' \ .$$

  The coherence conditions of Definition 1.29 ensure associativity of composition.

The obvious projection $\pi : Groth(F) \to \mathcal{B}$ that projects any pair onto its the second component is a fibration. A cartesian lifting for $\langle c, b\rangle$ with respect to $\beta : b' \to b$ is given by the pair $\langle 1_{F(\beta)c}, \beta\rangle$.

Our main example of a bifibration will be given by an elementary existential doctrine in the sense of Lawvere [20] whose categories of attributes will be presheaf categories. We will consider the following two conditions on fibrations, which usually arise in the context of categorical logic [22, 20].

Definition 1.31 *Let $\mathcal{P} : \mathcal{B}^{\mathrm{op}} \to \mathcal{CAT}$ be a pseudo-functor. If $\beta : b' \to b$ is an arrow in $\mathcal{B}$, we write β^* for $\mathcal{P}(\beta)$. Suppose that for any β, β^* has a left adjoint $\beta_!$.*

- **Beck-Chevalley Condition:** *Say that $\mathcal{P}$ satisfies the* Beck-Chevalley condition *if for every pullback square in $\mathcal{B}$*

  $$\begin{tikzcd} b''' \arrow[r, "\overline{\beta}"] \arrow[d, tail, "\overline{\alpha}"'] & b' \arrow[d, tail, "\alpha"] \\ b'' \arrow[r, "\beta"'] & b \end{tikzcd}$$

  with α monic, the following square commutes up to a natural isomorphism:

  $$\begin{tikzcd} \mathcal{P}(b''') \arrow[d, "\overline{\alpha}_!"'] & \mathcal{P}(b') \arrow[l, "\overline{\beta}^*"'] \arrow[d, "\alpha_!"] \\ \mathcal{P}(b'') & \mathcal{P}(b) \arrow[l, "\beta^*"] \ . \end{tikzcd}$$

- **Fröbenius Reciprocity Law:** *Suppose now that for every $b \in\mid\mathcal{B}\mid$, $\mathcal{P}(b)$ has binary products. Say that $\mathcal{P}$ satisfies the* Fröbenius Reciprocity Law *if* *for every $\beta : b' \to b$ and $c \in \mid \mathcal{P}(b) \mid$ the following square commutes up to a natural isomorphism:*

$$\begin{tikzcd}
b' \arrow[d, "\beta"'] & \mathcal{P}(b') \arrow[d, "\beta_!"'] & \mathcal{P}(b') \arrow[l, "\beta^*(c)\times -"'] \arrow[d, "\beta_!"] \\
b & \mathcal{P}(b) & \mathcal{P}(b) \arrow[l, "c\times -"'] \ .
\end{tikzcd}$$

We round off this section with a few facts about fibrations and indexed categories that we need later.

Proposition 1.32 *Let $\mathcal{P} : \mathcal{B}^{\mathrm{op}} \to \mathcal{CAT}$ be a pseudo-functor satisfying the Beck-Chevalley condition, then for any monic arrow $\beta : b' \rightarrowtail b$,*

$$\beta^* \beta_! \cong 1_{\mathcal{P}(b')} \ .$$

If $\mathcal{P}$ satisfies also the Fröbenius reciprocity law, then for any monic arrow $\beta : b' \rightarrowtail b$, the functor $\beta_!$ preserves products.

Proof: For the easy proof of the first statement see [22] P.175. The second, categorical folklore it seems, can be proved easily, as follows. Let c', d' be two objects of $\mathcal{P}(b')$, then

$$\begin{array}{rcll} \beta_!(c' \times d') & \cong & \beta_!(\beta^*\beta_! c' \times d') & \text{(from the property above)} \\ & \cong & \beta_! c' \times \beta_! d' & \text{(by Fröbenius reciprocity law)} \end{array}$$

□

We use the last result to prove a fact about products (Proposition 1.34) that will be useful later in Section 4 to prove that parallel compositions respect bisimulation. We first need a lemma.

Lemma 1.33 *Let $\mathcal{P} : \mathcal{B}^{\mathrm{op}} \to \mathcal{CAT}$ be a pseudo-functor, satisfying both the Beck-Chevalley condition and the Fröbenius Reciprocity law. Let the following square be a pullback of monomorphisms in $\mathcal{B}$:*

$$\begin{tikzcd}
a \arrow[r, tail, "\beta"] \arrow[d, tail, "\alpha"'] & b \arrow[d, tail, "\gamma"] \\
c \arrow[r, tail, "\delta"'] & d \ .
\end{tikzcd}$$

If X and Y are two objects of $\mathcal{P}(d)$ such that $\delta_!\delta^*X \cong X$ and $\gamma_!\gamma^*Y \cong Y$, then

$$X \times Y \cong \delta_!\alpha_!\alpha^*\delta^*(X \times Y) \ .$$

Proof:

$$\begin{aligned}
X \times Y &\cong \delta_!\delta^* X \times Y && \text{(by hypothesis)}\\
&\cong \delta_!(\delta^* X \times \delta^* Y) && \text{(by Fröbenius)}\\
&\cong \delta_!\delta^* X \times \delta_!\delta^* Y && \text{(by Proposition 1.32)}\\
&\cong X \times \delta_!\delta^* Y && \text{(by hypothesis)}\\
&\cong X \times \delta_!\delta^*\gamma_!\gamma^* Y && \text{(by hypothesis)}\\
&\cong X \times \delta_!\alpha_!\beta^*\gamma^* Y && \text{(by Beck-Chevalley)}\\
&\cong X \times \delta_!\alpha_!\alpha^*\delta^* Y && \text{(since } \delta\alpha \cong \gamma\beta\text{)}\\
&\cong \delta_!\alpha_!(\alpha^*\delta^* X \times \alpha^*\delta^* Y) && \text{(by Fröbenius)}\\
&\cong \delta_!\alpha_!\alpha^*\delta^*(X \times Y) && \text{(since } \alpha^* \text{ and } \delta^* \text{ are right adjoints).}
\end{aligned}$$

□

Proposition 1.34 *Let $\mathcal{P} : \mathcal{B}^{\mathrm{op}} \to \mathcal{CAT}$ be a pseudo-functor, satisfying both the Beck-Chevalley condition and the Fröbenius Reciprocity law. If a diagram*

$$\begin{tikzcd}
a & c \arrow[l, "\pi_a"'] \arrow[r, "\pi_b"] & b \\
d \arrow[u, tail, "i"] & f \arrow[l, "l"] \arrow[r, "r"'] \arrow[u, tail, "k"] & e \arrow[u, tail, "j"']
\end{tikzcd}$$

is a limiting cone in $\mathcal{B}$, then for any object $X \in \mid \mathcal{P}(d) \mid$ and $Y \in \mid \mathcal{P}(e) \mid$, there is an isomorphism in $\mathcal{P}(c)$,

$$k_!(l^* X \times r^* Y) \cong \pi_a^* i_! X \times \pi_b^* j_! Y \ .$$

Proof: Observe first of all that the limit of the diagram

$$\begin{tikzcd}
a & c \arrow[l, "\pi_a"'] \arrow[r, "\pi_b"] & b \\
d \arrow[u, tail, "i"] & & e \arrow[u, tail, "j"']
\end{tikzcd}$$

is obtained by taking three pullbacks, *i.e.*, the limiting cone can be constructed as follows,

$$\begin{tikzcd}
a & & c \arrow[ll, "\pi_a"'] \arrow[rr, "\pi_b"] & & b \\
d \arrow[u, tail, "i"] & \cdot \arrow[l, "\pi_d"] \arrow[ur, tail, "\delta"] & & \cdot \arrow[r, "\pi_e"'] \arrow[ul, tail, "\gamma"'] & e \arrow[u, tail, "j"'] \\
& & f \arrow[ul, tail, "\alpha"] \arrow[ur, tail, "\beta"'] & &
\end{tikzcd}$$

where all the quadrilaterals in the diagram above are pullbacks. Without loss of generality we can assume that $l = \pi_d \alpha$, $r = \pi_l \beta$ and $k = \delta\alpha = \beta\gamma$. We deduce the following natural isomorphism:

$$\begin{array}{rcll} \delta_! \delta^* \pi_a^* i_! & \cong & \delta_! \delta^* \delta_! \pi_d^* & \text{(by Beck-Chevalley)} \\ & \cong & \delta_! \pi_d^* & \text{(by Proposition 1.32)} \\ & \cong & \pi_a^* i_! & \text{(by Beck-Chevalley).} \end{array}$$

Similarly one deduces that $\gamma_! \gamma^* \pi_b^* j_! \cong \pi_b^* j_!$. Hence,

$$\begin{array}{rcll} \pi_a^* i_! X \times \pi_b^* j_! Y & \cong & \delta_! \alpha_! \alpha^* \delta^* (\pi_a^* i_! X \times \pi_b^* j_! Y) & \text{(by Lemma 1.33)} \\ & \cong & k_! (\alpha^* \delta^* \pi_a^* i_! X \times \alpha^* \delta^* \pi_b^* j_! Y) & \\ & \cong & k_! (\alpha^* \delta^* \delta_! \pi_d^* X \times \alpha^* \alpha_! \beta^* \pi_e^* Y) & \text{(by Beck-Chevalley)} \\ & \cong & k_! (\alpha^* \pi_d^* X \times \beta^* \pi_e^* Y) & \text{(by Proposition 1.32)} \\ & \cong & k_! (l^* X \times r^* Y)\ . & \end{array}$$

□

Proposition 1.35 *Let $\pi : \mathcal{E} \to \mathcal{B}$ be a fibration (cofibration). Let Δ be a class of diagram shapes (*i.e.*, a class of categories). Suppose that for every object b of $\mathcal{B}$, the fibre $\mathcal{E}_b$ has limits (colimits) of diagrams of shape δ for every $\delta \in \Delta$ and suppose that $\mathcal{B}$ has limits (colimits) of diagrams of shape δ for every $\delta \in \Delta$ too. Then $\mathcal{E}$ has limits (colimits) of diagrams of shape δ for every $\delta \in \Delta$.*

Proof: We simply give the description of how to build a limiting cone in $\mathcal{E}$ for a diagram of shape $\delta \in \Delta$. Let $F : \delta \to \mathcal{E}$ be a functor. Consider $\pi F : \delta \to \mathcal{B}$. By assumption there exists a limiting cone for πF. Let $b = \lim \pi F$ and for any $d \in\mid \delta \mid$, let $\beta_d : \pi F(d) \to b$ be the corresponding edge of the cone. Let $(\beta_d^* : \beta_d^*(F(d)) \to F(d))_{d \in |\delta|}$ be a family of cartesian arrows. This family induces a functor, $\beta_{(-)}^* F : \delta \to \mathcal{E}_b$. By assumption there exists a limiting cone to such a functor. Let $(e, f_d : e \to \beta_d^* F(d))$ be such a cone, then by post-composing with the corresponding cartesian arrows one obtains a limiting cone $(e, \beta_d^* f_d : e \to F(d))$ in $\mathcal{E}$. □

Morphisms of transition systems, synchronisation trees consist of pairs, one component of which is a partial function between labelling sets. By projecting to this component we obtain functors $p_{\mathcal{TS}} : \mathcal{TS} \to \mathbf{Set}_*$, $p_{\mathcal{ST}} : \mathcal{ST} \to \mathbf{Set}_*$ and $p_{\mathcal{ES}} : \mathcal{ES} \to \mathbf{Set}_*$ from the categories of models to $\mathbf{Set}_*$, the category of sets and partial functions.

Proposition 1.36 (Implicit in [29]) *The functors $p_{\mathcal{TS}} : \mathcal{TS} \to \mathbf{Set}_*$ and $p_{\mathcal{ST}} : \mathcal{ST} \to \mathbf{Set}_*$ are bifibrations. The functor $p_{\mathcal{ES}} : \mathcal{ES} \to \mathbf{Set}_*$ is a cofibration. There exist cartesian liftings of all monomorphisms.*

Notation: If L and M are two sets, we write $L \times_* M$ for the categorical product of L and M as objects of $\mathbf{Set}_*$. Concretely $L \times_* M$ can be realised as the disjoint union of sets $L + M + L \times M$, while $L \times M$ as the usual cartesian product of sets.

2 A general process language and its categorical models

We introduce the process language **Proc** of [29] within which a class of **CCS**-like languages can be expressed. The distinctive feature of **Proc** is that its parallel composition operator is a general product out of which different parallel compositions can be constructed with the help of restriction and relabelling operations. As noted in [29], if the terms of **Proc** are to be interpreted in categories of labelled structures (such as labelled transition systems, event structures) it is convenient to regard such categories as fibred over the labelling sets, more precisely over the category $\mathbf{Set}_*$ of sets and partial maps. Viewed in this way we obtain the universal characterisations of the operators associated with restriction and relabelling in terms of (co)cartesian liftings. Prefixing requires a more *ad hoc* treatment.

Define the terms of the language **Proc**:

$$t ::= \mathbf{Nil} \mid at \mid t_0 \oplus t_1 \mid t_0 \times t_1 \mid t\lceil\Lambda \mid t\{\Xi\} \mid x \mid rec\, x.t$$

where a is a label, Λ is a subset of labels, while Ξ is a total function from labels to labels, and x is drawn from some infinite set of a variables that we denote by $Vars$.

The handbook chapter [29] presented an analysis of the categorical status of the operations involved in the semantics of **Proc**. The structure, left somewhat implicit there, leads to an axiomatisation of the categories which are models of **Proc**. We emphasise the role of partial relabelling functions as substitution operators and impose upon them the Fröbenius reciprocity law and the Beck-Chevalley condition of Section 1.5

Definition 2.1 (Models for Proc) *A categorical model for* **Proc** *is given by a functor* $\pi : \mathcal{M} \to \mathbf{Set}_*$ *such that:*

- $\mathcal{M}$ *has binary products* $(\times)$.
- *For every set* L*, the fibre* $\mathcal{M}_L$ *has initial object* (0_L)*, binary coproducts* $(+_L)$ *and colimits of* ω*-chains.*
- *For every inclusion* $i : L \hookrightarrow M$ *of sets, there exists a cartesian lifting functor* $i^* : \mathcal{M}_M \to \mathcal{M}_L$.
- *For every total function* $f : L \to M$*, there exists a cocartesian lifting functor* $f_! : \mathcal{M}_L \to \mathcal{M}_M$.
- *For every set* L *and label* $a \in L$*, there exists a prefixing endofunctor*

$$Pre_{a,L} : \mathcal{M}_L \to \mathcal{M}_L$$

which preserves ω-colimits as well as existing cocartesian lifting functors for partial maps $f : L \rightharpoonup M$, that are defined on a, i.e., *if f is a partial map from L to M such that $f(a)$ is defined and such that $f_!$ exists, then the following square commutes, up to coherent isomorphism:*

$$\begin{tikzcd}
\mathcal{M}_L \arrow[r, "Pre_{a,L}"] \arrow[d, "f_!"'] & \mathcal{M}_L \arrow[d, "f_!"] \\
\mathcal{M}_M \arrow[r, "Pre_{f(a),M}"'] & \mathcal{M}_M .
\end{tikzcd}$$

- *Whenever applicable,* i.e., *whenever the required (co)cartesian arrows exist, the Fröbenius reciprocity law and Beck-Chevalley condition of Definition 1.31 hold.*

In [29] several models were considered, ranging from 'interleaving' models, like transition systems and synchronisation trees, to 'non-interleaving' models, like event structures, Petri nets or transition systems with independence. Here we recall briefly, how the structure required in the definition above is used to give semantics to terms of **Proc** .

We first derive some properties of models. In fact a model for **Proc**, as described in Definition 2.1, is not necessarily a cofibration. Still it has enough cocartesian liftings for us to deduce the following corollary (of the proof) of Proposition 1.35:

Corollary 2.2 *If $\pi : \mathcal{M} \to$ **Set**$_*$ is a model for* **Proc**, *then $\mathcal{M}$ has an initial object, binary coproducts and colimits of ω-chains.*

2.1 Denotational semantics of Proc

The operation $\oplus$ will be used to model the nondeterministic sum. This operation contrasts with the categorical sum in that the labelling set of the nondeterministic sum is obtained as a union rather than a disjoint union of the labelling sets of the two components.

Definition 2.3 *Let $\pi : \mathcal{M} \to$ **Set**$_*$ be a model for* **Proc** . *If M is an object of $\mathcal{M}_L$ and N an object of $\mathcal{M}_{L'}$, define $M \oplus N \in |\mathcal{M}_{L \cup L'}|$ to be*

$$M \oplus N = i_{L,!}(M) +_{L \cup L'} i_{L',!}(N) \ .$$

For any two sets L, L', this construction induces a functor $(- \oplus -) : \mathcal{M}_L \times \mathcal{M}_{L'} \to \mathcal{M}_{L \cup L'}$. However, because of the choice of taking the union and not the disjoint set of labelling sets, this does not lift to a functor $\mathcal{M} \times \mathcal{M} \to \mathcal{M}$; it does not have a well defined action on arrows. In fact it is not possible to define what $f \oplus g$ must be for two arbitrary arrows $f : M \to M'$ and $g : N \to N'$ of $\mathcal{M}$. A definition is possible, however, when $\pi(f)$ and $\pi(g)$ agree on $\pi(M) \cap \pi(N)$. This occurs for arrows in a common fibre when $\pi(f)$ and $\pi(g)$ will be the same identity function. More generally one can define a bifunctor $\oplus : \mathcal{M}_{in} \times \mathcal{M}_{in} \to \mathcal{M}_{in}$ where $\mathcal{M}_{in}$ is the subcategory of $\mathcal{M}$ of those arrows, f, such that $\pi(f)$ is an inclusion of sets.

Let $\pi : \mathcal{M} \to \mathbf{Set}_*$ be a model as in Definition 2.1. We describe the denotational semantics of **Proc** inductively on the structure of the well typed terms, assuming an environment $\rho : Vars \to |\mathcal{M}|$, a function to the objects of $\mathcal{M}$:

Nil: $[\![\mathbf{Nil}]\!]_\rho = 0_\emptyset$ an initial object of $\mathcal{M}$

Variables: $[\![x]\!]_\rho = \rho(x)$

Sum: $[\![t_1 \oplus t_2]\!]_\rho = [\![t_1]\!]_\rho \oplus [\![t_2]\!]_\rho$

Product: $[\![t_1 \times t_2]\!]_\rho = [\![t_1]\!]_\rho \times [\![t_2]\!]_\rho$

Restriction: $[\![t_1 \lceil \Lambda]\!]_\rho = i^*([\![t_1]\!]_\rho)$, where $i : \Lambda \cap \pi([\![t_1]\!]_\rho) \hookrightarrow \pi([\![t_1]\!]_\rho)$

*Relabelling:*For $\Xi : L \to M$ a relabelling function, $[\![t_1[\Xi]]\!]_\rho = \Xi'_!([\![t_1]\!]_\rho)$, where $\Xi' : \pi([\![t_1]\!]_\rho) \to \pi([\![t_1]\!]_\rho) \cup M$ is defined as $\Xi'(a) = \begin{cases} a & \text{if } a \notin L \\ \Xi(a) & \text{if } a \in L \end{cases}$.

Prefixing: $[\![at]\!]_\rho = Pre_{a,L\cup\{a\}}(i_!([\![t]\!]_\rho))$, where $\pi([\![t]\!]_\rho) = L$ and $i : L \hookrightarrow L \cup \{a\}$.

Recursion: Let t be any term, and let x be a variable (possibly free in t). Given any environment ρ the term t and the variable x determine an endofunctor

$$\begin{array}{rccl} t^x_\rho : & \mathcal{M}_{in} & \to & \mathcal{M}_{in} \\ & M & \mapsto & [\![t]\!]_{\rho[M/x]} \ . \end{array}$$

From t^x_ρ, the following ω-chain is derivable:

$$\begin{array}{rccll} T : & \omega & \to & \mathcal{M} & \\ & 0 & \mapsto & [\![t]\!]_{\rho[[\![\mathbf{Nil}]\!]_\rho/x]} & \\ & n & \mapsto & [\![t]\!]_{\rho[T^{n-1}/x]} & \text{for } n > 0 \end{array}$$

Define $[\![recx.t]\!]\rho = \operatorname{colim} T$. Since all the constructions involved in the denotation of a term t are ω-colimits preserving functors, then $\operatorname{colim} T$ is a fixed point for t^x_ρ.

The interpretation of the terms of **Proc** as objects of $\mathcal{M}$ has not involved Fröbenius reciprocity or the Beck-Chevalley condition. As we shall see in the next section these two conditions come into play when equipping models with a notion of bisimulation. Then the presence of these conditions will matter: they constrain the action that the cartesian arrows have on objects of $\mathcal{M}$ and are needed in showing that bisimulation is a congruence.

3 Presheaf models for Proc

As we saw in the previous section, the denotation of a term of **Proc** is given mainly in terms of universal constructions: (co)products, (co)cartesian liftings and fixed points. (The *ad hoc* treatment of prefixing is the exception.) Since we have an abstract definition of bisimulation via open maps we can hope to obtain abstract proofs of congruence with respect to bisimulation. But this task seems impossible with respect to the general definition of model for **Proc**. However, by specialising to presheaf models we can exploit the richness of constructions there to obtain a very general congruence theorem. The canonical embeddings from traditional models to presheaf categories preserve and reflect bisimulation as well as the process algebra operations. Consequently, the congruence results for presheaf models can be transferred to traditional models, as we do in Section 5 for the event structure model of **Proc**.

We begin by describing what we need to build a presheaf model for **Proc**. We require a family of path categories, indexed by sets of labels. Partial relabelling functions $L \rightharpoonup M$ will be associated with functors mapping computation paths over L to (possibly empty) computation paths over M. We will again need functors to represent prefixing, but this time they will be derived from more basic functors associated with lifting.

Definition 3.1 (Path structure for Proc) *A* path structure *for* **Proc** *consists of a functor $\mathbb{P}_{(-)}$ from* **Set**$_*$ *to* **Cat**, *the category of small categories, which sends $\lambda : L \rightharpoonup M$ to $\overline{\lambda} : (\mathbb{P}_L)_\perp \to (\mathbb{P}_M)_\perp$ such that:*

1. *The functors $\overline{\lambda}$, for $\lambda : L \rightharpoonup M$, preserve initial objects.*
2. *For each set L and element $a \in L$, there is an explicitly given* prefixing *functor $pre_{a,L} : (\mathbb{P}_L)_\perp \to \mathbb{P}_L$ satisfying commutativity of the following diagram, where I_M, I_L are the obvious embeddings:*

$$\begin{tikzcd}
(\mathbb{P}_L)_\perp \arrow[r, "pre_{a,L}"] \arrow[d, "\overline{\lambda}"'] & \mathbb{P}_L \arrow[r, hook, "I_L"] & (\mathbb{P}_L)_\perp \arrow[d, "\overline{\lambda}"] \\
(\mathbb{P}_M)_\perp \arrow[r, "pre_{\lambda(a),M}"'] & \mathbb{P}_M \arrow[r, hook, "I_M"'] & (\mathbb{P}_M)_\perp
\end{tikzcd}$$

for any $\lambda : L \rightharpoonup M$ that is defined on a.

A process with labelling set L is to denote a presheaf over $\mathbb{P}_L$.

With the Grothendieck construction of Section 1.6 in mind, one sees that a path structure for **Proc** defines a split cofibration in **Set**$_*$.

Example 3.2

1. *Define* $(-)^+ : \mathbf{Set}_* \to \mathbf{Cat}$ *to be the functor that associates to each set* L *the partial ordered set (regarded as a category)* L^+ *of finite non-empty strings of elements of* L*, and to each partial map* $\lambda : L \rightharpoonup M$ *the monotone map (*i.e.*, the functor)* $\overline{\lambda} : L^* \to M^*$ *that pointwise relabels every string over* L *to a string over* M *according to* λ*, sending every letter on which* λ *is undefined to the empty string (*ε*). The prefixing functors are defined by usual prefixing of strings,* i.e.*,* $pre_{a,L}(\sigma) = a\sigma$*.*

2. *Define* $\mathbf{Pom}_{(-)} : \mathbf{Set}_* \to \mathbf{Cat}$ *to be the functor that associates to each set* L *the category of non-empty pomsets labelled in* L*. If* $\lambda : L \rightharpoonup M$*, then* $\overline{\lambda} : (\mathbf{Pom}_L)_\perp \to (\mathbf{Pom}_M)_\perp$ *is the following functor:*

   - **On objects:** *Given a (possibly empty) pomset* $P = (P, \leq, l)$*,* $\overline{\lambda}(P) = (P', \leq', l')$ *with* $P' = \{e \in P \mid \lambda(l(e)) \text{ is defined}\}$*,* $\leq' = \leq \cap (P' \times P')$ *and* $l'(e) = \lambda(l(e))$*.*
   - **On arrows:** *If* $f : P \to Q$ *is an arrow in* $(\mathbf{Pom}_L)_\perp$*,* $\overline{\lambda}(f)$ *is simply the restriction of* f *to* P' *and* Q'*.*

   The prefixing functors are again the obvious ones, i.e.*, the prefixing* $pre_{a,L}(P)$ *of a pomset* P *is obtained by adding a new event, labelled 'a', which is placed below all the events of* P *in the causal order relation.*

Recall from the Definition 1.21 that from every functor $F : \mathbb{P}_\perp \to \mathbb{Q}_\perp$, that preserves the initial object, between small categories one can derive a pair of adjoint functors $F^l \dashv F^r : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$, with F^r being connected-colimit preserving. Hence, using the Grothendieck construction, from a path structure for **Proc** we can derive a bifibration, whose fibres will the be the categories $\widehat{\mathbb{P}_L}$.

3.1 The Grothendieck construction for presheaf models

Given a path structure $\mathbb{P}_{(-)}$, we can glue together all the fibres, consisting of categories of presheaves over $\mathbb{P}_L$, to form a fibration over $\mathbf{Set}_*$ which we call $Groth(\mathbb{P}_{(-)})$:

Objects: pairs $\langle X, L\rangle$ with $L \in |\mathbf{Set}_*|$ and X a presheaf over $\mathbb{P}_L$,

Arrows: pairs $\langle f, \lambda\rangle : \langle X, L\rangle \to \langle Y, M\rangle$ with $\lambda : L \rightharpoonup M$ and $f : X \to \overline{\lambda}^r(Y)$.

The composition of arrows is $\langle g, \mu\rangle \circ \langle f, \lambda\rangle = \langle \overline{\lambda}^r(g) \circ f, \mu \circ \lambda\rangle$. Clearly the projection $\langle X, L\rangle \mapsto L$ is the object part of a functor $\pi : Groth(\mathbb{P}_{(-)}) \to \mathbf{Set}_*$. Intuitively, the Grothendieck construction glues the various fibres together; it adds arrows between presheaves (possibly over different fibres), to allow for the possibility of a partial relabelling of actions. The adjunctions ensure that the Grothendieck fibration is in fact a bifibration [17]; the cocartesian lifting of λ with respect to X is $(\eta_X, \lambda) : X \to \overline{\lambda}^l(X)$ where $\eta_X : X \to \overline{\lambda}^r\overline{\lambda}^l(X)$ is the component of the unit of the adjunction at X. Since the fibres are presheaf categories they satisfy all the colimit completeness requirements in Definition 2.1. Moreover by applying Proposition 1.35 we can deduce that $Groth(\mathbb{P}_{(-)})$ has binary products.

Even if the functor $\mathbb{P}_{(-)}$ induces a split cofibration, whose fibres are the categories $(\mathbb{P}_L)_\perp$, for a set L, when extended to $Groth(\mathbb{P}_{(-)})$ this property is lost. On the other hand, since the $\overline{\lambda}^r$'s are defined by composition, $Groth(\mathbb{P}_{(-)})$ is a split fibration.

Definition 3.3 (Presheaf Models for Proc) *A* presheaf model *for* **Proc**, *consists of a path structure $\mathbb{P}_{(-)}$ as in Definition 3.1 satisfying the extra condition that the induced bifibration $Groth(\mathbb{P}_{(-)})$ satisfies both the Fröbenius Reciprocity law and the Beck-Chevalley condition.*

Presheaf models for **Proc**, $\mathbb{P}_{(-)}$, induce categorical models in the sense of Definition 2.1.

Theorem 3.4 *If $\mathbb{P}_{(-)}$ is a presheaf model for* **Proc**, *then $Groth(\mathbb{P}_{(-)})$ is a model in the sense of Definition 2.1, where the prefixing functors are defined as follows, for any $a \in L$:*

$$Pre_{a,L} : \widehat{\mathbb{P}_L} \xrightarrow{I_{L,*}} \widehat{(\mathbb{P}_L)_\perp} \xrightarrow{pre_{a,L,!}} \widehat{\mathbb{P}_L} \ ,$$

where we recall that $pre_{a,L,!} = \mathrm{Lan}_{\mathrm{y}_{(\mathbb{P}_L)_\perp}}(\mathrm{y}_{\mathbb{P}_L} pre_{a,L})$ and $I_{L,*}$ is defined as in Section 1.4.

To prove the theorem we need the following lemma:

Lemma 3.5 *Let $F : \mathbb{P}_\perp \to \mathbb{Q}_\perp$ be a functor and let $I_{\mathbb{P}} : \mathbb{P} \to \mathbb{P}_\perp$ and $I_{\mathbb{Q}} : \mathbb{Q} \to \mathbb{Q}_\perp$ be the obvious embeddings, then the following square commutes up to a natural isomorphism:*

$$\begin{tikzcd}
\widehat{\mathbb{P}} \arrow[r, "I_{\mathbb{P},*}"] \arrow[d, "F^l"'] & \widehat{\mathbb{P}_\perp} \arrow[d, "F_!"] \\
\widehat{\mathbb{Q}} \arrow[r, "I_{\mathbb{Q},*}"] & \widehat{\mathbb{Q}_\perp} \ .
\end{tikzcd}$$

Proof: Observe that all the arrows in the above square are connected-colimit preserving functors. It is now an easy calculation to show that

$$F_! I_{\mathbb{P},*} \mathrm{y}^\circ_{\mathbb{P}} \cong I_{\mathbb{Q},*} F^l \mathrm{y}^\circ_{\mathbb{P}} \ .$$

Thus $F_! I_{\mathbb{P},*}$ is isomorphic to $I_{\mathbb{Q},*} F^l$. □

Proof:[of Theorem 3.4] The only thing which we need to check is that the prefixing functors satisfy the condition given in Definition 2.1. The check is provided by the following chain of isomorphisms:

$$\begin{array}{rcll}
\overline{\lambda}^l Pre_{a,L} & = & \overline{\lambda}^l pre_{a,L,!} I_{L,*} & \text{(by definition of } Pre\text{)} \\
& \cong & I_M^* \overline{\lambda}_! I_{L,!} pre_{a,L,!} I_{L,*} & \text{(by Proposition 1.22)} \\
& \cong & I_M^* I_{M,!} pre_{\lambda(a),M,!} \overline{\lambda}_! I_{L,*} & \text{(by Definition 3.1)} \\
& \cong & pre_{\lambda(a),M,!} \overline{\lambda}_! I_{L,*} & \text{(since } I_M \text{ is an embedding)} \\
& \cong & pre_{\lambda(a),M,!} I_{M,*} \overline{\lambda}^l & \text{(by Lemma 3.5)} \\
& \cong & Pre_{\lambda(a),M} \overline{\lambda}^l & \text{(by definition of } Pre\text{).}
\end{array}$$

□

We can now equip our model with a canonical notion of bisimulation. We first bring two objects over a common fibre and then see whether they are open map bisimilar there.

Definition 3.6 (Open map bisimulation in $Groth(\mathbb{P}_{(-)})$**)** *Let* $\langle X, L\rangle$ *and* $\langle Y, M\rangle$ *be two objects in* $Groth(\mathbb{P}_{(-)})$*. We say that they are* (open map) bisimilar *if* $\overline{i_L}^l(\langle X, L\rangle)$ *and* $\overline{i_M}^l(\langle Y, M\rangle)$ *are related by a span of* $\mathbb{P}_{L\cup M}$*-open maps, where* i_L *and* i_M *are the set inclusions* $L \overset{i_L}{\hookrightarrow} L \cup M \overset{i_M}{\hookleftarrow} M$*.*

NOTATION: In the remainder of this paper, we shall write $\langle X, L\rangle \sim \langle Y, M\rangle$ to mean that they are open map bisimilar, and $X \sim_L Y$ to say that both X and Y are in $|\widehat{\mathbb{P}_L}|$ and that they are $\mathbb{P}_L$-open bisimilar. Hence we have that

$$\langle X, L\rangle \sim \langle Y, M\rangle \text{ if and only if } \overline{i_L}^l(X) \sim_{L\cup M} \overline{i_M}^l(Y)\ .$$

From the above definition we immediately see that moving objects across different fibres along cocartesian liftings preserves bisimulation. But we can deduce more: bisimulation is a congruence with respect to the process operations. Its proof relies mainly on Theorems 1.17 and 1.18. We begin by observing that for any $\langle X, L\rangle$ in $Groth(\mathbb{P}_{(-)})$, there exists a least subset L' of L with inclusion function $i : L' \hookrightarrow L$, such that, $\langle \overline{i}^r X, L'\rangle \sim \langle X, L\rangle$.

Proposition 3.7 *Let* $\langle X, L\rangle \in |Groth(\mathbb{P}_{-})|$*. Let* $M \overset{i}{\hookrightarrow} L \overset{j}{\hookleftarrow} N$ *be two inclusions, such that* $\langle \overline{i}^r X, M\rangle \sim \langle X, L\rangle \sim \langle \overline{j}^r X, N\rangle$*, then if*

$$\begin{tikzcd}
M \cap N \arrow[r, hook, "l"] \arrow[d, hook, "k"'] & M \arrow[d, hook, "i"] \\
N \arrow[r, hook, "j"'] & L
\end{tikzcd}$$

is the obvious pullback square of inclusions,

$$\langle \overline{j}^r \overline{k}^r X, M \cap N\rangle = \langle \overline{l}^r \overline{i}^r X, M \cap N\rangle \sim \langle X, L\rangle\ .$$

Proof: The equality holds obviously. For bisimilarity we need to show that $\overline{i}^l\overline{l}^l\overline{l}^r\overline{i}^r X \sim_L X$. By assumption $\overline{i}^l\overline{i}^r X \sim_L X$, hence by Theorem 1.18, $\overline{j}^r\overline{i}^l\overline{i}^r X \sim_N \overline{j}^r X$. By the Beck-Chevalley condition, $\overline{j}^r\overline{i}^l\overline{i}^r X \cong \overline{k}^l\overline{l}^r\overline{i}^r X$, hence by composing with $\overline{j}^l$, $\overline{j}^l\overline{k}^l\overline{l}^r\overline{i}^r X \sim_L \overline{j}^l\overline{j}^r X \sim_L X$. Since the square of inclusions is a commutative one, we know that $\overline{j}^l\overline{k}^l \cong \overline{i}^l\overline{l}^l$, thus

$$\overline{i}^l\overline{l}^l\overline{l}^r\overline{i}^r X \sim_L X \ .$$

□

Corollary 3.8 *For any $\langle X, L\rangle \in\mid Groth(\mathbb{P}_{-})\mid$, there exists a least subset L' of L such that $\langle \overline{i}^r X, L'\rangle \sim \langle X, L\rangle$, where $i : L' \hookrightarrow L$ is the inclusion function.*

Proof: Take L' to be equal to the intersection of all $M \subseteq L$, such that $\langle \overline{i_M}^r X, M\rangle \sim \langle X, L\rangle$, where $i_M : M \hookrightarrow L$ is the inclusion function. □

Definition 3.9 *For any $\langle X, L\rangle \in\mid Groth(\mathbb{P}_{-})\mid$, say that X* reduces to *M, if M is a subset of L, and $\langle X, L\rangle \sim \langle \overline{i}^r X, M\rangle$. If L' is the least subset of L for which X can be reduced to, say that X has* support *L'.*

That a presheaf X over paths $\mathbb{P}_L$ reduces to a presheaf over paths $\mathbb{P}_{L'}$ with $L' \subseteq L$ means that X is described to within bisimulation as a presheaf over $\mathbb{P}_{L'}$.

The above results yield the following characterisations of bisimilarity.

Proposition 3.10 *Two objects $\langle X, L\rangle, \langle Y, M\rangle \in\mid Groth(\mathbb{P}_{-})\mid$ are bisimilar iff they both reduce to $L \cap M$ and $\overline{i}^r X \sim_{L\cap M} \overline{j}^r Y$, for $L \overset{i}{\hookleftarrow} L \cap M \overset{j}{\hookrightarrow} M$.*

Corollary 3.11 *Two objects $\langle X, L\rangle, \langle Y, M\rangle \in\mid Groth(\mathbb{P}_{-})\mid$ are bisimilar iff they have the same set N as support with $L \overset{i}{\hookleftarrow} N \overset{j}{\hookrightarrow} M$ and $\overline{i}^r X \sim_N \overline{j}^r Y$.*

As we shall see all the operations involved in the semantics of **Proc** preserve bisimulation.

Before going into the semantics of **Proc** in presheaf models we recall the following instantiation of Proposition 1.32 to $Groth(\mathbb{P}_{(-)})$.

Proposition 3.12 *Let $\mathbb{P}_{(-)} : \mathbf{Set}_* \rightarrow \mathbf{Cat}$ be a presheaf model for* **Proc** *. Let $i : L \rightarrowtail M$ be a monomorphism in $\mathbf{Set}_*$,* i.e., *an injective (total) function. Then the following facts hold:*

- $\overline{i}^r\overline{i}^l \cong 1_{\widehat{\mathbb{P}_L}}$.
- $\overline{i}^l$ *preserves products in the fibres,* i.e., *for every two presheaves $X, Y \in\mid \widehat{\mathbb{P}_L}\mid$,*

$$\overline{i}^l(X \times Y) \cong \overline{i}^l(X) \times \overline{i}^l(Y) \ .^5$$

[5] We use the same symbol "×" to indicate the product of two objects in a category, irrespective of the fact that sometime, as here, products taken in different categories appear in the same expression. We believe the reader can disambiguate from the context.

4 Semantic constructions in $Groth(\mathbb{P}_{(-)})$

We analyse the constructions used in $Groth(\mathbb{P}_{(-)})$ to give the semantics of **Proc** according to Section 2.1 and show that they preserve bisimulation.

Products: As we have already said the category $Groth(\mathbb{P}_{(-)})$ has products. They can be constructed (*cf.* Proposition 1.35) using the products in the fibres as follows. Given $\langle X, L\rangle, \langle Y, M\rangle \in |Groth(\mathbb{P}_{(-)})|$. Define

$$\langle X, L\rangle \times \langle Y, M\rangle = \langle \overline{\pi_L}^r(X) \times \overline{\pi_M}^r(Y), L \times_* M\rangle$$

where $L \xleftarrow{\pi_L} L \times_* M \xrightarrow{\pi_M} M$ are the projections of the product in $\mathbf{Set}_*$.

Proposition 4.1 *If $\langle X, L\rangle$ is open map map bisimilar to $\langle X', L'\rangle$ and $\langle Y, M\rangle$ is open map bisimilar to $\langle Y', M'\rangle$ then the product $\langle X, L\rangle \times \langle Y, M\rangle$ is open map bisimilar to the product $\langle X', L'\rangle \times \langle Y', M'\rangle$.*

Proof: Let $N = L \times_* M$ and $N' = L' \times_* M'$. Consider the diagram

$$\begin{tikzcd}
L \arrow[d, hook, "i_L"'] & N \arrow[l, "\pi_L"'] \arrow[r, "\pi_M"] \arrow[d, hook, "i_N"'] & M \arrow[d, hook, "i_M"] \\
L \cup L' & N \cup N' \arrow[l, "\alpha"'] \arrow[r, "\beta"] & M \cup M' \\
L' \arrow[u, hook, "i'_L"] & N' \arrow[l, "\pi_{L'}"'] \arrow[r, "\pi_{M'}"] \arrow[u, hook, "i_{N'}"'] & M' \arrow[u, hook, "i_{M'}"'] ,
\end{tikzcd}$$

with α and β the obvious projecting partial functions. Observe that since the set $N \cup N'$ is included in $(L \cup L') \times_* (M \cup M')$, then both the upper diagram and the lower one are limiting cones of vertex N and N', respectively. Hence by Proposition 1.34 we have the following two isomorphisms in $\widehat{\mathbb{P}_{N\cup N'}}$:

$$\begin{aligned}
\overline{i_N}^l(\overline{\pi_L}^r X \times \overline{\pi_M}^r Y) &\cong \overline{\alpha}^r \overline{i_L}^l X \times \overline{\beta}^r \overline{i_M}^l Y \\
\overline{i_{N'}}^l(\overline{\pi_{L'}}^r X' \times \overline{\pi_{M'}}^r Y') &\cong \overline{\alpha}^r \overline{i_{L'}}^l X' \times \overline{\beta}^r \overline{i_{M'}}^l Y' \ .
\end{aligned}$$

Consequently, we have to prove that assuming $\overline{i_L}^l X \sim_{L\cup L'} \overline{i_{L'}}^l X'$ and $\overline{i_M}^l Y \sim_{M\cup M'} \overline{i_{M'}}^l Y'$ we have

$$\overline{\alpha}^r \overline{i_L}^l X \times \overline{\beta}^r \overline{i_M}^l Y \sim_{N\cup N'} \overline{\alpha}^r \overline{i_{L'}}^l X' \times \overline{\beta}^r \overline{i_{M'}}^l Y' \ .$$

By Theorem 1.18, we deduce that $\overline{\alpha}^r \overline{i_L}^l X \sim_{N\cup N'} \overline{\alpha}^r \overline{i_{L'}}^l X'$ and $\overline{\beta}^r \overline{i_M}^l Y \sim_{N\cup N'} \overline{\beta}^r \overline{i_{M'}}^l Y'$. Hence, by Proposition 1.7

$$\overline{\alpha}^r \overline{i_L}^l X \times \overline{\beta}^r \overline{i_M}^l Y \sim_{N\cup N'} \overline{\alpha}^r \overline{i_{L'}}^l X' \times \overline{\beta}^r \overline{i_{M'}}^l Y' \ .$$

□

Note that the above proof makes essential use of the Beck-Chevalley condition and the Fröbenius Reciprocity law (via Proposition 1.34).
Sum: Let $\langle X, L\rangle, \langle Y, M\rangle \in |Groth(\mathbb{P}_{(-)})|$. Define

$$\langle X, L\rangle \oplus \langle Y, M\rangle = \langle \overline{i_L}^{l}(X) + \overline{i_M}^{l}(Y), L \cup M\rangle$$

where $L \xrightarrow{i_L} L \cup M \xleftarrow{i_M} M$ are the obvious set inclusions.

Proposition 4.2 *The functor $\oplus$ preserves open map bisimulation; if $\langle X, L\rangle$ is open map bisimilar to $\langle X', L'\rangle$ and $\langle Y, M\rangle$ is open map bisimilar to $\langle Y', M'\rangle$ then $\langle X, L\rangle \oplus \langle Y, M\rangle$ is open map bisimilar to $\langle X', L'\rangle \oplus \langle Y', M'\rangle$*

Proof: The proof is a straightforward application of Theorem 1.17. Let $N = L \cup M$, $N' = L' \cup M'$, $L'' = L \cup L'$ and $N'' = N \cup N'$. Consider the following diagram of inclusions:

$$\begin{tikzcd}
L \arrow[r, hook, "j_L"] \arrow[d, hook, "i_L"'] & N \arrow[d, hook, "i_N"'] & M \arrow[l, hook', "j_M"'] \arrow[d, hook, "i_M"] \\
L'' \arrow[r, hook, "j_{L''}"] & N'' & M'' \arrow[l, hook', "j_{M''}"'] \\
L' \arrow[r, hook, "j_{L'}"'] \arrow[u, hook', "i_{L'}"] & N' \arrow[u, hook', "i_{N'}"'] & M' \arrow[l, hook', "j_{M'}"] \arrow[u, hook', "i_{M'}"']
\end{tikzcd}$$.

By assumption $\overline{i_{L,!}}X \sim_{L''} \overline{i_{L',!}}X'$ and $\overline{i_{M,!}}X \sim_{M''} \overline{i_{M',!}}X'$, hence

$$\begin{aligned}
\overline{i_{N,!}}(\overline{j_{L,!}}X + \overline{j_{M,!}}Y) &\cong \overline{i_{N,!}j_{L,!}}X + \overline{i_{N,!}j_{M,!}}Y \\
&\cong \overline{j_{L'',!}i_{L,!}}X + \overline{j_{M'',!}i_{M,!}}Y \\
&\sim_{N''} \overline{j_{L'',!}i_{L',!}}X' + \overline{j_{M'',!}i_{M',!}}Y' \\
&\cong \overline{i_{N',!}j_{L',!}}X' + \overline{i_{N',!}j_{M',!}}Y' \\
&\cong \overline{i_{N',!}}(\overline{j_{L',!}}X' + \overline{j_{M',!}}Y') \ .
\end{aligned}$$

□

Remark: This sum construction is not the coproduct because of the choice of labelling set for the sum. It can be shown that, if $[i_L, i_M] : L + M \to L \cup M$ is the mediating map from the coproduct of sets, then

$$\langle X, L\rangle \oplus \langle Y, M\rangle \cong \overline{[i_L, i_M]}^{l}(\langle X, L\rangle + \langle Y, M\rangle).$$

Restriction: Let Λ be a set and let $\langle X, L\rangle \in |Groth(\mathbb{P}_{(-)})|$. Then consider the inclusion map $i : \Lambda \cap L \hookrightarrow L$ and define the restriction of X to $\Lambda \cap L$ to be

$$\langle X, L\rangle\lceil\Lambda = \langle \overline{i}^{r}(X), \Lambda \cap L\rangle \ .$$

Proposition 4.3 *The functor $(-)\lceil\Lambda$ preserves open map bisimulation; if $\langle X, L\rangle$ is open map bisimilar to $\langle X', L'\rangle$ then $\langle X, L\rangle\lceil\Lambda$ is open map bisimilar to $\langle X', L'\rangle\lceil\Lambda$*

Proof: Consider the diagram of inclusions

$$\begin{tikzcd}
\Lambda \cap L \arrow[r, hook, "k"] \arrow[d, hook, "j_L"'] & L \arrow[d, hook, "i_L"] \\
\Lambda \cap (L \cup L') \arrow[r, hook, "l"] & L \cup L' \\
\Lambda \cap L' \arrow[r, hook, "m"] \arrow[u, hook, "j_{L'}"] & L' . \arrow[u, hook, "i_{L'}"']
\end{tikzcd}$$

Both squares are readily seen to be pullbacks, therefore by Beck-Chevalley, $\overline{j_L}^l\overline{k}^r \cong \overline{l}^r\overline{i_L}^l$ and $\overline{j_{L'}}^l\overline{m}^r \cong \overline{l}^r\overline{i_{L'}}^l$. Assuming, $\overline{i_L}^l X \sim_{L\cup L'} \overline{i_{L'}}^l X'$, we can deduce

$$\begin{array}{rcll}
\overline{j_L}^l\overline{k}^r X & \cong & \overline{l}^r\overline{i_L}^l X & \\
 & \sim_{\Lambda\cap(L\cup L')} & \overline{l}^r\overline{i_{L'}}^l X' & \text{(by Theorem 1.18)} \\
 & \cong & \overline{j_{L'}}^l\overline{m}^r X' \ . &
\end{array}$$

□

Relabelling: Let $\Xi : L \to M$ be total. Take $\langle X, L\rangle$ as usual, define the relabelling to be

$$\langle X, L\rangle[\Xi] = \langle \overline{\Xi}^l(X), M\rangle \ .$$

Relabelling preserves bisimulation:

Proposition 4.4 *If $\langle X, L\rangle$ is open map bisimilar to $\langle X', L'\rangle$ then $\langle X, L\rangle[\Xi]$ is open map bisimilar to $\langle X', L'\rangle[\Xi]$*

Proof: Again this is a straightforward application of Theorem 1.17. Consider the commuting diagram

$$\begin{tikzcd}
L \arrow[r, "\Xi_L"] \arrow[d, hook, "i_L"'] & \Xi_L L \arrow[d, hook, "j_L"] \\
L \cup L' \arrow[r, "\Xi_{L\cup L'}"] & \Xi_{L\cup L'}(L \cup L') \\
L' \arrow[r, "\Xi_{L'}"] \arrow[u, hook, "i_{L'}"] & L' . \arrow[u, hook, "j_{L'}"']
\end{tikzcd}$$

Knowing that $\overline{i_{L,!}}X \sim_{L\cup L'} \overline{i_{L',!}}X'$, we obtain

$$\begin{array}{rcl} \overline{j_{L,!}}\,\overline{\Xi_{L,!}}X & \cong & \overline{\Xi_{L\cup L',!}}\,\overline{i_{L,!}}X \\ & \sim_{\Xi_{L\cup L'}L\cup L'} & \overline{\Xi_{L\cup L',!}}\,\overline{i_{L',!}}X' \\ & \cong & \overline{j_{L',!}}\,\overline{\Xi_{L',!}}X' \ . \end{array}$$

□

Prefixing: Prefixing is dealt with by using the functors $Pre_{a,L}$ defined as in Theorem 3.4. The preservation property of bisimulation is automatically ensured.

Proposition 4.5 *Let $a \in L \cap L'$. If $\langle X, L\rangle$ is open map bisimilar to $\langle X', L'\rangle$ then $Pre_{a,L}(\langle X, L\rangle)$ is open map bisimilar to $Pre_{a,L'}(\langle X', L'\rangle)$*

Recursion: Letting $F : Groth(\mathbb{P}_{(-)}) \to Groth(\mathbb{P}_{(-)})$ be a functor, define $rec(F)$ to be the colimit $\mathrm{colim}\,\omega_F$ where

$$\begin{array}{rrcl} \omega_F : & \omega & \to & Groth(\mathbb{P}_{(-)}) \\ & n & \mapsto & F^n(\langle 0, \emptyset\rangle). \end{array}$$

Here 0 is the unique, up to isomorphism, presheaf over $\mathbb{P}_\emptyset$. Any $F^n(\langle 0, \emptyset\rangle)$ consists of a pair $\langle X_n, L_n\rangle$ with $X_n \in |\widehat{\mathbb{P}_{L_n}}|$, and we can express the colimit as a pair $\langle X, L\rangle$, where L is the colimit in $\mathbf{Set}_*$ of the L_n and X is the colimit in $\widehat{\mathbb{P}_L}$ of all the cocartesian liftings of the X_n, along the edges of the cocone $i_n : L_n \to L$.

We already observed that the operations on $Groth(\mathbb{P}_{(-)})$ associated with the term constructors are functors but for the sum $(- \oplus -)$; the sum nevertheless becomes functorial if one restricts to $Groth(\mathbb{P}_{(-)})_{in}$, the subcategory of $Groth(\mathbb{P}_{(-)})$ with morphisms given by pairs $\langle f, i\rangle$ where i is an inclusion of sets. Having $F : Groth(\mathbb{P}_{(-)})_{in} \to Groth(\mathbb{P}_{(-)})_{in}$, we can define $rec(F)$ as above. Notice that $L = \cup_n L_n$ and every $i_n : L_n \to L$ is an inclusion of sets.

All our constructions are continuous with respect to ω-chains (particular kinds of connected diagram) and restrict to $Groth(\mathbb{P}_{(-)})_{in}$. Hence $rec(F)$ determines a fixed point where F is derived from a denotation of a term t as in Section 2.1. So, the construction above yields a denotation for a recursively defined process in terms of an ω-colimit of presheaves over a common path category. We would like to deduce the bisimulation of recursive processes $rec\ x.t$, $rec\ y.u$ from bisimulation between the open terms t and u. Such open terms give rise to endofunctors on $Groth(\mathbb{P}_{(-)})_{in}$ We start by extending the notion of open map, and therefore bisimulation, to functors. Following Definition 3.6 and 3.9, we first say when an arrow $\langle f, i\rangle$ in $Groth(\mathbb{P}_{(-)})_{in}$ (and hence in $Groth(\mathbb{P}_{(-)})$) is open.

Definition 4.6 *An arrow $\langle f, i\rangle : \langle X, L\rangle \to \langle Y, M\rangle$ in $Groth(\mathbb{P}_{(-)})_{in}$ is open if the transpose $f' : \overline{i}^l X \to Y$ of f, with respect to the adjunction $\overline{i}^l \dashv \overline{i}^r$, is $\mathbb{P}_M$-open.*

Proposition 4.7 *If $\langle f, i\rangle : \langle X, L\rangle \to \langle Y, M\rangle$ is open in the sense of Definition 4.6 above, then f is $\mathbb{P}_L$-open and Y reduces to L.*

Proof: By the adjunction $f = \overline{i}^r(f')\eta_X$, where η_X is the unit at X of the adjunction $\overline{i}^l \dashv \overline{i}^r$. But we know that η_X is an isomorphism, since i is a monomorphism (*cf.* Proposition 3.12), hence f is the composite of two open maps and therefore is open. So we have that $X \sim_L \overline{i}^r Y$, hence $\overline{i}^l X \sim_M \overline{i}^l\overline{i}^r Y$. Therefore $Y \sim_M \overline{i}^l X \sim_M \overline{i}^l\overline{i}^r Y$. □

An obvious question is whether the Proposition 4.7 above can be made into an "if and only if" statement. However, we seem to need an extra assumption on the presheaf model to obtain the converse implication. For instance, using Lemma 6(ii) of [19] we can obtain the following:

Proposition 4.8 *Let $\mathbb{P}_{(-)} : \mathbf{Set}_* \to \mathbf{Cat}$ be a presheaf model such that for any injective total function (*i.e.*, any monomorphism of $\mathbf{Set}_*$), $i : L \to M$ and for any two objects $P \in |\mathbb{P}_L|$ and $Q \in |\mathbb{P}_M|$,*

$$\mathbb{P}_M[Q, \overline{i}P] \neq \emptyset \textit{ only if } Q \cong \overline{i}P' \textit{ for some } P' \in |\mathbb{P}_L| \quad .$$

Then, for any $Y \in |\widehat{\mathbb{P}_M}|$ with support L, the counit, ε_Y, of the adjunction $\overline{i}^l \dashv \overline{i}^r$ is $\mathbb{P}_M$-open.

It is now seen that the proposition above induces the converse of Proposition 4.7, since $f' = \varepsilon_Y \overline{i}^l f$. Notice that both the presheaf models of Example 3.2 satisfy the condition required by Proposition 4.8.

Back to recursion:

Definition 4.9 *Let $F, G : \mathcal{C} \to Groth(\mathbb{P}_{(-)})_{in}$ be two functors. Let $\alpha : F \stackrel{\cdot}{\longrightarrow} G$ be a natural transformation. Say that α is open if for every $c \in |\mathcal{C}|$, α_c is open according to Definition 4.6.*

We consider two endofunctors F, G on $Groth(\mathbb{P}_{(-)})_{in}$ bisimilar *if there is another endofunctor R and a span of open natural transformations $\alpha : R \stackrel{\cdot}{\to} F$ and $\beta : R \stackrel{\cdot}{\to} G$ relating them.*

Proposition 4.10 *Let $\mathcal{C}$ be a category with initial object 0. Every natural transformation $\alpha : R \stackrel{\cdot}{\longrightarrow} F$, with $R, F : \mathcal{C} \to \mathcal{C}$ endofunctors induces a natural transformation $\omega_\alpha : \omega_R \stackrel{\cdot}{\longrightarrow} \omega_F$ where ω_R (ω_F) are defined inductively by:*

- $\omega_R(0) = 0$ *($\omega_F(0) = 0$)*
- $\omega_R(n+1) = R(\omega_R(n))$ *($\omega_F(n+1) = F(\omega_F(n))$*

- $\omega_R(0 \leq 1) = 0_{R0}$ *($\omega_F(0 \leq 1) = 0_{F0}$)*
- $\omega_R(n{+}1 \leq n{+}2) = R(\omega_R(n \leq n{+}1)$ *($\omega_F(n{+}1 \leq n{+}2) = F(\omega_F(n \leq n{+}1))$*

Proof: Inductively define

- $(\omega_\alpha)_0 \overset{\text{def}}{=} 1_0$
- $(\omega_\alpha)_{n+1} \overset{\text{def}}{=} \alpha_{F^n 0} R((\omega_\alpha)_n) = F((\omega_\alpha)_n)\alpha_{R^n 0}$ where the second equality holds by naturality of α.

To check that ω_α is a natural transformation, we need to show that the following square commutes for any $n \geq 0$:

$$\begin{tikzcd}
R^n 0 \arrow[r, "\omega_R(n \leq n+1)"] \arrow[d, "(\omega_\alpha)_n"'] & R^{n+1} 0 \arrow[d, "(\omega_\alpha)_{n+1}"] \\
F^n 0 \arrow[r, "\omega_F(n \leq n+1)"'] & F^{n+1} 0 \ .
\end{tikzcd}$$

The proof goes routinely, by induction. The base case follows immediately by initiality of 0. Assuming $n > 0$, we see that

$$\begin{aligned}
(\omega_\alpha)_{n+1}\omega_R(n \leq n+1) &= (\omega_\alpha)_{n+1} R(\omega_R(n-1 \leq n)) \\
&\qquad \text{(by definition of } \omega_R\text{)} \\
&= \alpha_{F^n 0} R((\omega_\alpha)_n) R(\omega_R(n-1 \leq n)) \\
&\qquad \text{(by definition of } \omega_\alpha\text{)} \\
&= \alpha_{F^n 0} R((\omega_\alpha)_n \omega_R(n-1 \leq n)) \\
&\qquad \text{(by functoriality of } R\text{)} \\
&= \alpha_{F^n 0} R(\omega_F(n-1 \leq n)(\omega_\alpha)_{n-1}) \\
&\qquad \text{(by inductive hypothesis)} \\
&= \alpha_{F^n 0} R(\omega_F(n-1 \leq n)) R((\omega_\alpha)_{n-1}) \\
&\qquad \text{(by functoriality of } R\text{)} \\
&= F(\omega_F(n-1 \leq n))\alpha_{F^{n-1}0} R((\omega_\alpha)_{n-1}) \\
&\qquad \text{(by naturality of } \alpha\text{)} \\
&= \omega_F(n \leq n+1)(\omega_\alpha)_n \\
&\qquad \text{(by definition).}
\end{aligned}$$

□

We deduce:

Proposition 4.11 *Let F, R be endofunctors of $Groth(\mathbb{P}_{(-)})_{in}$ and let $\alpha : R \dot{\to} X$ be a natural transformation. Then there is a natural transformation $\omega_\alpha : \omega_R \dot{\to} \omega_F$. Moreover if α is open and X preserve open morphisms, then ω_α is open.*

Proof: Proposition 4.10, instantiated to $Groth(\mathbb{P}_{(-)})_{in}$, proves the first part. The second part is immediately verified by inspecting the definition of ω_α. □

Open maps are preserved in passing to the colimit, in particular:

Proposition 4.12 *Let $F, R : \omega \to \widehat{\mathbb{P}}$ be functors and $\alpha : R \dot{\to} F$ a natural transformation such that for every n, α_n is a $\mathbb{P}$-open map. Then the map* $\operatorname{colim} \alpha : \operatorname{colim} R \to \operatorname{colim} F$*, uniquely determined by the universal property of colimits, is a $\mathbb{P}$-open map.*

Proof: Let the following be a commutative square with P and Q objects of $\mathbb{P}$:

$$\begin{tikzcd}
P \arrow[r, "p"] \arrow[d, "m"'] & \operatorname{colim} R \arrow[d, "\operatorname{colim} \alpha"] \\
Q \arrow[r, "q"'] & \operatorname{colim} F \, .
\end{tikzcd} \tag{1}$$

Since $\operatorname{colim} R$ and $\operatorname{colim} F$ are colimits of ω-chains there exists a number n and arrows

$$p_n : P \to R(n) \qquad \text{and} \qquad q_n : Q \to F(n)$$

such that the following diagram commutes:

$$\begin{tikzcd}
P \arrow[r, "p_n"] \arrow[d, "m"'] \arrow[rr, bend left, "p"] & R(n) \arrow[r, "R_{n,\infty}"] \arrow[d, "\alpha_n"] & \operatorname{colim} R \arrow[d, "\operatorname{colim} \alpha"] \\
Q \arrow[r, "q_n"'] \arrow[rr, bend right, "q"'] & F(n) \arrow[r, "F_{n,\infty}"'] & \operatorname{colim} F \, ,
\end{tikzcd}$$

where $R_{n,\infty}$ and $F_{n,\infty}$ are edges of the corresponding colimiting cones. By assumption, α_n is $\mathbb{P}$-open, hence there exists $r_n : Q \to R(n)$ splitting the leftmost square in two commutative triangles. Then

$$r \overset{\text{def}}{=} R_{n,\infty} r_n$$

is an arrow from Q to $\operatorname{colim} R$ that splits the Diagram (1) in two commutative triangles.

If every α_n is an epimorphic natural transformation, then obviously, since colimits in presheaf categories are calculated pointwise, $\operatorname{colim} \alpha$ is epimorphic as well. □

Since the calculation of colimits of ω-chains in $Groth(\mathbb{P}_{(-)})_{in}$ is reduced to calculating them in the colimiting fibre, the above proposition yields:

Proposition 4.13 *Let $\omega_F, \omega_R : \omega \to Groth(\mathbb{P}_{(-)})_{in}$ and $\omega_\alpha : \omega_R \dot{\to} \omega_F$, be as in Proposition 4.11, with ω_α open, then the (adjoint transpose of the) arrow*

$$\operatorname{colim} \omega_\alpha : \operatorname{colim} \omega_R \to \operatorname{colim} \omega_F \ ,$$

uniquely determined by the universal property of the colimit, is an open map in the fibre over the colimiting labelling set.

Proof: We have already remarked that colimits of ω-chains

$$\cdots \to \langle X_n, L_n \rangle \to \langle X_{n+1}, L_{n+1} \rangle \to \cdots \ ,$$

in $Groth(\mathbb{P}_{(-)})_{in}$ are obtained by taking first the union

$$L = \cup_{n \in \omega} L_n$$

of the all the labelling sets in the chain and then calculating the colimit of the chain induced in the fibre over L, by cocartesian liftings of all the X_n. We now make this explanation more precise, in order to show that the functor part of $\operatorname{colim} \alpha$ arises from a situation satisfying the hypothesis of Proposition 4.12. We need some notation first. Let's write $\langle R_n, L_n \rangle$ for $\omega_R(n)$, $\langle F_n, M_n \rangle$ for $\omega_F(n)$. For every n, let $\langle r_n, i_n \rangle$ and $\langle f_n, j_n \rangle$ be $\omega_R(n \leq n+1)$ and $\omega_F(n \leq n+1)$, where, for simplicity we already assume that $r_n : \overline{(i_n)}^l R_n \to R_{n+1}$ and $f_n : \overline{(j_n)}^l F_n \to F_{n+1}$, rather than taking their transposes. For every n, let $\langle \alpha_n, k_n \rangle$ be $(\omega_\alpha)_n$, where again we take $\alpha_n : \overline{(k_n)}^l R_n \to F_n$. Naturality of ω_α means that the following square commutes (where the indicated isomorphisms are uniquely determined by the universal property of left Kan extensions):

$$\begin{tikzcd}
\overline{(j_n k_n)}^l R_n = \overline{(k_{n+1} i_n)}^l R_n \arrow[r, "\simeq"] \arrow[d, "\cong"'] & \overline{(k_{n+1})}^l \overline{(i_n)}^l R_n \arrow[r, "\overline{(k_{n+1})}^l r_n"] & \overline{(k_{n+1})}^l R_{n+1} \arrow[dd, "\alpha_{n+1}"] \\
\overline{(j_n)}^l \overline{(k_n)}^l R_n \arrow[d, "\overline{(j_n)}^l \alpha_n"'] & & \\
\overline{(j_n)}^l F_n \arrow[rr, "f_n"'] & & F_{n+1} \ .
\end{tikzcd}$$

Now, writing

$$i_{n,\infty} : L_n \to L = \cup_{n \in \omega} L_n \ ,$$

for every n, and $\langle R_\infty, L\rangle$ for $\operatorname{colim}\omega_R$, we have that R_∞ can be calculated as the colimit of the following chain in $\widehat{\mathbb{P}_L}$:

$$\begin{tikzcd}
\cdots \arrow[r] & \overline{(i_{n,\infty})}^{l}R_n \arrow[r, "\simeq"] & \overline{(i_{n+1,\infty})}^{l}\overline{(i_n)}^{l}R_n \arrow[r, "\overline{(i_{n+1,\infty})}^{l}r_n"] & \overline{(i_{n+1,\infty})}^{l}R_{n+1} \arrow[r] & \cdots\ .
\end{tikzcd}$$

Similarly one calculates $\operatorname{colim} F$, moreover from the commutativity of the above diagram it follows that for every n, the following diagram commutes too (where $k : L \hookrightarrow M$ is the inclusion function):

$$\begin{tikzcd}
\overline{k}^{l}\overline{(i_{n,\infty})}^{l}R_n \arrow[r, "\simeq"] \arrow[d, "\cong"'] & \overline{k}^{l}\overline{(i_{n+1,\infty})}^{l}\overline{(i_n)}^{l}R_n \arrow[r, "\overline{k}^{l}\overline{(i_{n+1,\infty})}^{l}r_n"] & \overline{k}^{l}\overline{(i_{n+1,\infty})}^{l}R_{n+1} \arrow[d, "\cong"] \\
\overline{(j_{n,\infty}k_n)}^{l} = \overline{(ki_{n,\infty})}^{l}R_n \arrow[d, "\cong"'] & & \overline{(j_{n+1,\infty}k_{n+1})}^{l} = \overline{(ki_{n+1,\infty})}^{l}R_{n+1} \arrow[d, "\cong"] \\
\overline{(j_{n,\infty})}^{l}\overline{(k_n)}^{l}R_n \arrow[d, "\overline{(j_{n,\infty})}^{l}\alpha_n"'] & & \overline{(j_{n+1,\infty})}^{l}\overline{(k_{n+1})}^{l}R_{n+1} \arrow[d, "\overline{(j_{n+1,\infty})}^{l}\alpha_{n+1}"] \\
\overline{(j_{n,\infty})}^{l}F_n \arrow[r, "\simeq"] & \overline{(j_{n+1,\infty})}^{l}\overline{(j_n)}^{l}F_n \arrow[r, "\overline{(j_{n+1,\infty})}^{l}f_n"] & \overline{(j_{n+1,\infty})}^{l}F_{n+1}\ .
\end{tikzcd}$$

Hence $\alpha_\infty : \overline{k}^{l}R_\infty \to F_\infty$ is the unique mediating morphism between two colimiting cones connected by a natural transformation that is pointwise an open map, since every vertical arrow in the diagram above is either an isomorphism (hence an open map) or the transformation along a colimit preserving functor of an open map (hence an open map because of Theorem 1.17). This falls within the hypothesis of Proposition 4.12, and so α_∞ is $\mathbb{P}_M$-open. □

Consequently, if two endofunctors F, G ranging over $Groth(\mathbb{P}_{(-)})_{in}$ are bisimilar and preserve open maps, then the colimits $rec(F)$, $rec(G)$ are bisimilar. A term with a free variable, built-up from the constructions of this section, will determine an endofunctor on $Groth(\mathbb{P}_{(-)})_{in}$ which preserves open maps by this section propositions. It follows that if two open terms t and u are bisimilar, *i.e.*, induce bisimilar functors, then the recursive definitions $rec\ x.t$ and $rec\ y.u$ are bisimilar.

5 Concrete models revisited

We have already mentioned the full embeddings

$$\mathcal{ST}_L \hookrightarrow \widehat{L^+}$$
$$\mathcal{ES}_L \hookrightarrow \widehat{\mathbf{Pom}_L}$$

The first is actually an equivalence of categories. The second is a strict inclusion (for instance the terminal presheaf cannot be represented as an event structure) that not only preserves but reflects bisimulation (see Proposition 1.12). We consider the presheaf models $Groth((-)^+)$ and $Groth(\mathbf{Pom}_{(-)})$ of Example 3.2. We can now transfer the results from the presheaf models to the concrete models of synchronisation trees and event structures by noting that the canonical embeddings between fibres, $\mathcal{ST}_L \to \widehat{L^+}$ and $\mathcal{ES}_L \to \widehat{\mathbf{Pom}_L}$, extend to full and faithful embeddings from $\mathcal{ST}$ and $\mathcal{ES}$ to the corresponding presheaf models. In particular we again have that the embedding $\mathcal{ST} \hookrightarrow Groth((-)^+)$ is an equivalence. We illustrate then the situation with event structures.

We recalled in Proposition 1.36 that the functor $p_{\mathcal{ES}} : \mathcal{ES} \to \mathbf{Set}_*$ is a cofibration. We now provide a cocleavage which is functorial (*cf.* end of Section 1.5) and hence makes $p_{\mathcal{ES}} : \mathcal{ES} \to \mathbf{Set}_*$ into a split cofibration.

Let $E = (E, \leq, Con, l : E \to L)$ be an event structure and let $\alpha : L \rightharpoonup L'$ be a partial function. define $\alpha_!(E) = (E', \leq', Con', l' : E' \to L')$ to consist of:

- $E' = \{e \in E \mid \alpha(l(e)) \text{ is defined}\}$
- $\leq' = \leq \cap (E' \times E')$
- $Con' = \{x \in Con \mid x \subseteq E'\}$
- $l'(e') = \alpha(l(e'))$, for every $e' \in E'$.

It is straightforward to verify that $\alpha_!(E)$ is an event structure and that the pair $\langle 1', \alpha\rangle$, with $1' : E \rightharpoonup E'$ the truncation of the identity functions on events to E', is an event structure morphism.

Proposition 5.1 *Given an event structure $E = (E, \leq, Con, l : E \to L)$ and a partial function $\alpha : L \rightharpoonup L'$, the event structures morphism $\langle 1', \alpha\rangle$ is a cocartesian arrow.*

We will write $\alpha_{!,E}$ for $\langle 1', \alpha\rangle$. There is an induced cocartesian lifting functor $\alpha_! : \mathcal{ES}_L \to \mathcal{ES}_{L'}$. If we restrict the construction above to pomsets we obtain the functor $\overline{\alpha} : (\mathbf{Pom}_L)_\perp \to (\mathbf{Pom}_{L'})_\perp$ of Example 3.2.

It is easy to verify that this choice of cocartesian lifting functors is functorial:

Proposition 5.2 *Given an event structure $E = (E, \leq, Con, l : E \to L)$ and a partial function $\alpha : L \rightharpoonup L'$, the following hold:*

1. *If $\alpha = 1_L$, then $\alpha_{!,E}$ is the identity morphism $\langle 1_E, 1_L\rangle : E \to E$.*
2. *If $\beta : L' \to L''$ is another partial function, then $\beta_{!,\alpha_!(E)}\alpha_{!,E} = (\beta\alpha)_{!,E}$.*

Notation: If $\langle f, \alpha\rangle : E \to \overline{E}$ is an event structure morphism, write f_α for the unique function such that $\langle f, \alpha\rangle = \langle f_\alpha, 1'_L\rangle \alpha_{!,E}$. That is, f_α is the restriction of f to the elements of $\alpha_!(E)$ that, by the way, is equal to the set $\{e \in E \mid f(e) \text{ is defined}\}$. Call f_α the transpose of f.

With this notation in mind, define $c : \mathcal{ES} \to Groth(\mathbf{Pom}_{(-)})$ to be

- **On objects:** $c(E, \leq, Con, l : E \to L) = \langle c_L(E), L\rangle$
- **On arrows:** If $\langle f, \alpha\rangle : E \to E'$ with $\alpha : L \rightharpoonup L'$, then $c(\langle f, \alpha\rangle) = \langle c(f), \alpha\rangle$ where

  $$c(f) : c_L(E) = \mathcal{ES}_L[-, E] \to \mathcal{ES}_{L'}[\overline{\alpha}(-), E'] = \overline{\alpha}^r(c_{L'}(E'))$$

  is defined by composition and transposition (recall that on pomsets, $\alpha_!$ is another name for $\overline{\alpha}$), *i.e.*, $c(f)_P(p) = (fp)_\alpha$.

This defines a functor because from Proposition 5.2 one has that $(1_E p)_{1_L} = p$ and $(g(fp)_\alpha)_\beta = (gfp)_{\beta\alpha}$ and from these equalities one deduces that $c\langle 1_E, 1_L\rangle = \langle 1_E, 1_L\rangle$ and $c(\langle g, \beta\rangle\langle f, \alpha\rangle) = c(\langle gf, \beta\alpha\rangle)$. Moreover, again from Proposition 5.2, one sees that for any arrow $\langle f, 1_L\rangle : E \to E'$, $(fp)_{1_L} = fp$, hence c acts as c_L when restricted to $\mathcal{ES}_L$.

Proposition 5.3 *The functor $c : \mathcal{ES} \to Groth(\mathbf{Pom}_{(-)})$ is a dense full embedding.*

Proof: Straightforward from the fact that c extends the c_L's that were dense full embeddings and the fact that via cocartesian liftings, every arrow between objects of $Groth(\mathbf{Pom}_{(-)})$ in different fibres is uniquely determined by an arrow in a fibre. □

It is known that every dense full embedding preserves limits [30]. Moreover a direct calculation shows that c respects relabelling (*i.e.*, cocartesian liftings) and cartesian liftings of inclusions.

Proposition 5.4 *Let $\alpha : L \rightharpoonup M$ be a partial function, then there is a natural isomorphism*

$$c_M \alpha_! \cong \overline{\alpha}^l c_L \ .$$

Let $\lambda : L \hookrightarrow M$ be an inclusion map, then

$$c_L \lambda^* \cong \overline{\lambda}^r c_M \ ,$$

where λ^* is the right adjoint of $\lambda_!$ defined on objects as follows:

$$\lambda(E, \leq, Con, l) = (E', \leq', Con', l') \ ,$$

where $E' = \{e \in E | \forall e' \leq e \, \exists a \in L \, \lambda(a) = l(e')\}$, $\leq' = \leq \cap E' \times E'$, $Con' = \{x \in Con \mid x \subseteq E'\}$ *and* $l'(e) = a$, *where* a *is the unique element of* L, *such that* $\lambda(a) = l(e)$.

We have already noticed that c_L preserves coproducts (Proposition 1.13). After Proposition 1.35 we know that coproducts in a cofibred category are built using coproducts in the fibres and cocartesian liftings, hence c preserves coproducts. Summarising:

Proposition 5.5 *The embedding* $c : \mathcal{ES} \to Groth(\mathbf{Pom}_{(-)})$ *preserves all limits that exists in* $\mathcal{ES}$, *coproducts, cocartesian liftings and cartesian liftings of inclusion.*

A denotational semantics of **Proc** in $\mathcal{ES}$ was given in [29] and corresponds to the one described abstractly in Section 2.1. Proposition 5.5 above ensure that (after the embedding with c) the semantics in $\mathcal{ES}$ correspond to the one in $Groth(\mathbf{Pom}_{(-)})$, hence the following:

Theorem 5.6 *Let* $\mathcal{ES}[\![\cdot]\!]$ *and* $Groth(\mathbf{Pom}_{(-)})[\![\cdot]\!]$ *stand for the respective semantics of* **Proc**. *Let* $\rho : Vars \to |\mathcal{ES}|$ *be an environment function, then*

$$c(\mathcal{ES}[\![t]\!]_\rho) \cong Groth(\mathbf{Pom}_{(-)})[\![t]\!]_{c\circ\rho} \, .$$

By Proposition 1.12(ii), open maps and bisimulation coincide, via the canonical embeddings, in $\mathcal{ES}_L$ and the fibre over L in $Groth(\mathbf{Pom}_{(-)})$. Hence we can transfer the congruence property deduced for the presheaf semantics to deduce, in particular, that hereditary history-preserving bisimulation is a congruence for the language **Proc**.

Theorem 5.7 *Hereditary history-preserving bisimulation is a congruence for the language* **Proc** *with respect to the event structure semantics of the language* **Proc**.

6 Refinement for event structures

As a further example of an application of Theorem 1.17, we prove the refinement operator for event structures proposed in [12] preserves hereditary history-preserving bisimulation (abbreviated to hhpb).

Definition 6.1 (*cf.* [12], Section 2) *A* refinement function *is a function* r *from* L *to* $|\mathbf{Pom}_M|$, *objects of* $\mathbf{Pom}_M$; *so any element* $a \in L$ *is sent to a non-empty pomset* $r(a)$ *over* M.

Definition 6.2 (A refinement functor) *A refinement function as in the definition above induces a* refinement functor

$$R : \mathbf{Pom}_L \to \mathbf{Pom}_M$$

acting as follows:

- **On objects:** *If $P = (P, \leq_P, \lambda_P, L)$ is a pomset over L, then define $R(P) = (R(P), \leq_{R(P)}, \lambda_{R(P)}, M)$ with:*
  - $R(P) = \{(x, x') | x \in P \wedge x' \in r(\lambda_P(x))\}$
  - $(x, x') \leq_{R(P)} (y, y')$ *if either $x \leq_P y$ and $x \neq y$ or $x = y$ and $x' \leq_{r(\lambda_P(x))} y'$*
  - $\lambda_{R(P)}(x, x') = \lambda_{r(\lambda_P(x))}(x')$.
- **On arrows:** *If $f : P \to Q$ in $\mathbf{Pom}_L$, define $R(f)(x, x') = (f(x), x')$.*

One can see r as inducing a refinement functor, say $R^{\mathcal{ES}}$, on event structures as well. If $(E, \leq, Con, l)$ is an event structure over L, $R^{\mathcal{ES}}(E)$ is defined on E, $\leq$ and l as for pomsets, while $X \in Con_{R^{\mathcal{ES}}(E)}$ iff $\{x \in E \mid \exists x' .(x, x') \in X\} \in Con$.

As remarked in [19], the functor $R_!$, obtained as a left Kan extension, is a good candidate for the extension of this refinement to presheaves including those corresponding to event structures. But does the functor $R_!$ act like the operation of refinement $R^{\mathcal{ES}}$ on event structures? More precisely, if we let $c_L : \mathcal{ES}_L \to \widehat{\mathbf{Pom}}_L$ and $c_M : \mathcal{ES}_M \to \widehat{\mathbf{Pom}}_M$ denote the canonical embeddings, do we have that the following square commutes (up to a natural isomorphism)?

$$\begin{tikzcd}
\mathcal{ES}_L \arrow[r, "c_L"] \arrow[d, "R^{\mathcal{ES}}"'] & \widehat{\mathbf{Pom}_L} \arrow[d, "R_!"] \\
\mathcal{ES}_M \arrow[r, "c_M"'] & \widehat{\mathbf{Pom}_M}
\end{tikzcd}$$

We embark on proving that it does.

Lemma 6.3 *Let E be an event structure in $\mathcal{ES}_L$, let Q be a pomset over M and let $R^{\mathcal{ES}} : \mathcal{ES}_L \to \mathcal{ES}_M$ be a refinement functor. Then for any $q : Q \to R^{\mathcal{ES}}(E)$ there exists a pomset $P_q \in |\mathbf{Pom}_L|$ and a morphism $p : P \to E$ such that*

- *There exists a morphism $p_q : Q \to R^{\mathcal{ES}}(P) = R(P)$ such that $q = R^{\mathcal{ES}}(p)p_q$.*

- *For any other factorisation*

$$\begin{tikzcd}
Q \arrow[rr, "q"] \arrow[dr, "p'_q"'] & & R^{\mathcal{ES}}(E) \\
& R^{\mathcal{ES}}(P') \arrow[ur, "R^{\mathcal{ES}}(p')"'] &
\end{tikzcd}$$

  there exists a unique mediating morphism of pomsets, $m : P \to P'$, such that

$$p'_q = R^{\mathcal{ES}}(m)p_q \qquad \text{and} \qquad p'm = p \ .$$

Proof: Define $P = \{e \in E | \exists (e,f) \in R(E) \, \exists y \in Q \, q(y) = (e,f)\}$, with the order relation induced by Q, *i.e.*, $e \leq_P e'$ if either $e = e'$ or there exist $y \leq_Q y'$ with $q(y) = (e,f)$ and $q(y') = (e',f')$. The verification of the properties is straightforward. □

Proposition 6.4 *Let $i_L : \mathbf{Pom}_L \to \mathcal{ES}_L$ and $i_M : \mathbf{Pom}_M \to \mathcal{ES}_M$ be the inclusion functors, then*

$$R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(i_M \circ R).$$

Proof: Recall that $\mathbf{Pom}_L$ is dense in $\mathcal{ES}_L$, *i.e.*, for every $E \in |\mathcal{ES}_L|$,

$$E \cong \mathrm{colim}\, i_L/E \to \mathbf{Pom}_L \xrightarrow{i_L} \mathcal{ES}_L.$$

Using the Lemma 6.3 above it is not difficult to verify that

$$R^{\mathcal{ES}}(E) \cong \mathrm{colim}\, i_L/E \to \mathbf{Pom}_L \xrightarrow{R} \mathbf{Pom}_M \xrightarrow{i_M} \mathcal{ES}_M \ . \tag{2}$$

From this we can deduce that $R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(i_M \circ R)$. In fact $R^{\mathcal{ES}} \circ i_L = i_M \circ R$ and moreover if $F : \mathcal{ES}_L \to \mathcal{ES}_M$ is a functor and $\alpha : F \circ i_L \stackrel{\cdot}{\longrightarrow} R \circ i_m$ is a natural transformation, there exists a unique $\beta : R^{\mathcal{ES}} \stackrel{\cdot}{\longrightarrow} F$ such that

$$\beta_{i_L} = \alpha. \tag{3}$$

To show this, observe first of all that if β is a natural transformation satisfying (3), then for any $E \in |\mathcal{ES}_L|$ and $f : P \to E$,

$$\beta_E \circ R^{\mathcal{ES}}(f) = F(f) \circ \alpha_P \ . \tag{4}$$

In fact

$$\begin{array}{rcll}
\beta_E \circ R^{\mathcal{ES}}(f) & = & F(f) \circ \beta_{i_L(P)} & \text{(by naturality of } \beta\text{)} \\
& = & F(f) \circ \alpha_P & \text{(by equation (3)).}
\end{array}$$

But since, for any $E \in \mid \mathcal{ES}_L \mid$, (2) holds, there exists a unique $\beta_E : R^{\mathcal{ES}}(E) \to F(E)$ satisfying $\beta_E \circ R^{\mathcal{ES}}(f) = F(f) \circ \alpha_P$. Commutativity of the naturality squares follows as well from the universal property of colimits. We need to prove that for any $g : E \to E'$ in $\mathcal{ES}_L$, the following diagram commutes:

$$\begin{tikzcd}
R^{\mathcal{ES}}(E) \arrow[r, "\beta_E"] \arrow[d, "R^{\mathcal{ES}}(g)"'] & F(E) \arrow[d, "F(g)"] \\
R^{\mathcal{ES}}(E') \arrow[r, "\beta_{E'}"'] & F(E') \ .
\end{tikzcd}$$

It is enough to show that for any $f : P \to E$, $F(g) \circ \beta_E \circ R^{\mathcal{ES}}(f) = \beta_{E'} \circ R^{\mathcal{ES}}(g) \circ R^{\mathcal{ES}}(f)$. This follows by the following calculation:

$$\begin{aligned}
F(g) \circ \beta_E \circ R^{\mathcal{ES}}(f) &= F(g) \circ F(f) \circ \alpha_P && \text{(by equation (4))} \\
&= F(gf) \circ \alpha_P && \text{(by functoriality of } F\text{)} \\
&= \beta_{E'} \circ R^{\mathcal{ES}}(gf) && \text{(by equation (4))} \\
&= \beta_{E'} \circ R^{\mathcal{ES}}(g) \circ R^{\mathcal{ES}}(f) && \text{(by functoriality of } R^{\mathcal{ES}}\text{)}
\end{aligned}$$

□

We then have a functor $R : \mathbf{Pom}_L \to \mathbf{Pom}_M$ that can be extended as follows:

$$\begin{tikzcd}
\mathbf{Pom}_L \arrow[rr, "i_L"] \arrow[dr, "R"'] & & \mathcal{ES}_L \arrow[r, "c_L"] \arrow[dd, "R^{\mathcal{ES}}=\mathrm{Lan}_{i_L}(i_M R)"] & \widehat{\mathbf{Pom}_L} \arrow[dd, "\mathrm{Lan}_{\mathrm{y}_{\mathbf{Pom}_L}}(\mathrm{y}_{\mathbf{Pom}_M} R)=R_!"] \\
& \mathbf{Pom}_M \arrow[dr, "i_M"'] & & \\
& & \mathcal{ES}_M \arrow[r, "c_M"'] & \widehat{\mathbf{Pom}_M} \ .
\end{tikzcd}$$

We want to show that the square on the right commutes up to a natural isomorphism. We show first of all that

$$c_M R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(c_M i_M R) = \mathrm{Lan}_{i_L}(\mathrm{y}_{\mathbf{Pom}_M} R) \ .$$

From this in fact it will follow that (using that left Kan extensions compose, *cf.* Section 1.4),

$$\begin{aligned}
\mathrm{Lan}_{\mathrm{y}_{\mathbf{Pom}_L}}(\mathrm{y}_{\mathbf{Pom}_M} R) &\cong \mathrm{Lan}_{c_L}(\mathrm{Lan}_{i_L}(\mathrm{y}_{\mathbf{Pom}_M} R) \\
&\cong \mathrm{Lan}_{c_L}(c_M R^{\mathcal{ES}}) \ .
\end{aligned}$$

Hence, since c_L is full and faithful (*cf.* Proposition 1.15)

$$\mathrm{Lan}_{\mathrm{y}_{\mathbf{Pom}_L}}(\mathrm{y}_{\mathbf{Pom}_M} R) c_L \cong \mathrm{Lan}_{c_L}(c_M R^{\mathcal{ES}}) c_L \cong c_M R^{\mathcal{ES}} \ .$$

To prove that $c_M R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(c_M i_M R)$, we apply Proposition 1.11.

Proposition 6.5 *There is a natural isomorphism*

$$c_M R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(c_M i_M R) \ .$$

Proof: The proof is an immediate consequence of Proposition 1.11 and of Lemma 6.3. In fact, as we saw in the proof of Proposition 6.4, $\mathrm{Lan}_{i_L}(i_M R)$ can be expressed as the colimit (2) and Lemma 6.3 ensures that the conditions of Proposition 1.11 are met. □

Now we can use Theorem 1.17 to deduce that $R_!$ preserves open map bisimulation.

Proposition 6.6 *For any refinement function $r : L \to\mid \mathbf{Pom}_M \mid$, the associated refinement functor $R_! : \widehat{\mathbf{Pom}_L} \to \widehat{\mathbf{Pom}_M}$ preserves open map bisimulation; if X and Y are two $\mathbf{Pom}_L$-open bisimilar presheaves, then $R_!(X)$ and $R_!(Y)$ are $\mathbf{Pom}_M$-open bisimilar.*

As a consequence, using Proposition 1.12, we have:

Corollary 6.7 *For any refinement function $r : L \to\mid \mathbf{Pom}_M \mid$, the associated refinement functor $R^{\mathcal{ES}} : \mathcal{ES}_L \to \mathcal{ES}_M$ preserves hereditary history-preserving bisimulation; If E and E' are two hereditary history-preserving bisimilar event structures in $\mathcal{ES}_L$, then $R^{\mathcal{ES}}(E)$ and $R^{\mathcal{ES}}(E')$ are hereditary history-preserving bisimilar.*

Proof:

$$\begin{array}{rcl} E \text{ hhpb } E' & \Longrightarrow & c_L(E)\ \mathbf{Pom}_L\text{-open bisimilar to } c_L(E') \\ & \Longrightarrow & R_! c_L(E)\ \mathbf{Pom}_M\text{-open bisimilar to } R_! c_L(E') \\ & \Longleftrightarrow & c_M R^{\mathcal{ES}}(E)\ \mathbf{Pom}_M\text{-open bisimilar to } c_M R^{\mathcal{ES}}(E') \\ & \Longleftrightarrow & R^{\mathcal{ES}}(E) \text{ hhpb } R^{\mathcal{ES}}(E') \ . \end{array}$$

□

Acknowledgements Thanks are due to Gordon Plotkin who suggested improvements.

References

[1] M. Bednarczyk. Hereditary history preserving bisimulation or what is the power of the future perfect in program logics. Technical report, Polish Academy of Sciences, Gdansk, 1991.

[2] F. Borceux. *Handbook of categorical algebra I*, volume 50 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1994.

[3] G. L. Cattani. *Presheaf Models for Concurrency.* PhD thesis, University of Aarhus, 1999.

[4] G. L. Cattani, M. P. Fiore, and G. Winskel. A theory of recursive domains with applications to concurrency. In *LICS '98, Proceedings of the Thirteenth Annual IEEE Symposium on Logic in Computer Science*, pages 214–225. IEEE Computer Society Press, 1998.

[5] G. L. Cattani, I. Stark, and G. Winskel. Presheaf models for the π-calculus. In *Proceedings of the 7th International Conference on Category Theory and Computer Science, CTCS '97*, number 1290 in Lecture Notes in Computer Science, pages 106–126. Springer-Verlag, 1997.

[6] G. L. Cattani and G. Winskel. Presheaf models for concurrency. In D. van Dalen and M. Bezem, editors, *Computer Science Logic. 10th International Workshop, CSL '96, Annual Conference of the European Association for Computer Science Logic. Selected Papers*, volume 1258 of *Lecture Notes in Computer Science*, pages 58–75. Springer-Verlag, 1997.

[7] G. L. Cattani and G. Winskel. Profunctors, open maps and bisimulation. Manuscript in preparation, 1999.

[8] A. Cheng and M. Nielsen. Open maps, behavioural equivalences, and congruences. *Theoretical Computer Science*, 190(1):87–112, 1998.

[9] M. P. Fiore. *Axiomatic Domain Theory in Categories of Partial Maps.* Distinguished Dissertations in Computer Science. Cambridge University Press, 1996.

[10] M. P. Fiore, G. L. Cattani, and G. Winskel. Weak bisimulation and open maps (extended abstract). In *LICS '99, Proceedings of the Fourteenth Annual IEEE Symposium on Logic in Computer Science*, pages 67–76. IEEE Computer Society Press, 1999.

[11] M. P. Fiore, G. D. Plotkin, and A. J. Power. Complete cuboidal sets in axiomatic domain theory. In *LICS '97, Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 268–279. IEEE Computer Society Press, 1997.

[12] R. van Glabbeek and U. Goltz. Equivalence notions for concurrent systems and refinement of actions. In *Mathematical Foundations of Computer Science*

1989, number 379 in Lecture Notes in Computer Science, pages 237–248. Springer-Verlag, 1989.

[13] A. Grothendieck. *Revêtements étales et groupe fondamental*, volume 224 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1971. Séminaire de Géométrie Algébrique du Bois Marie 1960–1961 (SGA 1), Dirigé par Alexandre Grothendieck. Augmenté de deux exposés de M. Raynaud.

[14] C. Hermida. *Fibrations Logical Predicates and Indeterminates.* PhD thesis, University of Edinburgh, 1993. Available as Technical Report DAIMI-PB 462, Computer Science Department, University of Aarhus.

[15] T. T. Hildebrandt. A fully abstract presheaf semantics for SCCS with finite delay. In *Proceedings of the 8th International Conference on Category Theory and Computer Science, CTCS '99*, ENTCS. Elsevier, 1999. To appear.

[16] T. T. Hildebrandt, P. Panangaden, and G. Winskel. A relational model of non-deterministic dataflow. In D. Sangiorgi and R. de Simone, editors, *Proceedings of the 9th International Conference on Concurrency Theory, CONCUR '98*, volume 1466 of *Lecture Notes in Computer Science*, pages 613–628. Springer-Verlag, 1998.

[17] B. Jacobs. *Categorical Logic and Type Theory.* Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.

[18] A. Joyal and I. Moerdijk. A completeness theorem for open maps. *Annals of Pure and Applied Logic*, 70(1):51–86, 1994.

[19] A. Joyal, M. Nielsen, and G. Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, 1996.

[20] F. W. Lawvere. Equality in hyperdoctrines and comprehension schema as an adjoint functor. In *Applications of Categorical Algebra (Proc. Sympos. Pure Math., Vol. XVII, New York, 1968)*, pages 1–14. Amer. Math. Soc., 1970.

[21] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics.* Springer-Verlag, 1971.

[22] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory.* Springer-Verlag, 1992.

[23] M. Nielsen and G. Winskel. Petri nets and bisimulation. *Theoretical Computer Science*, 153:211–244, 1996.

[24] G. D. Plotkin. Algebraic completeness and compactness in an enriched setting. Invited lecture given at the Workshop on Logic, Domains, and Programming Languages. Darmstadt, 1995.

[25] V. Pratt. Modelling concurrency with partial orders. *International Journal of Parallel Processing*, 15:33–71, 1986.

[26] A. Rabinovitch and B. Traktenbrot. Behaviour structures and nets. *Fundamenta Informatica*, 11(4):357–404, 1988.

[27] G. Winskel. A category of labelled petri nets and compositional proof system. In *LICS '88, Proceedings of the Third Annual IEEE Symposium on Logic in Computer Science*, pages 142–154. IEEE Computer Society Press, 1988.

[28] G. Winskel. A presheaf semantics of value-passing processes (extended abstract). In U. Montanari and V. Sassone, editors, *CONCUR'96, Proceedings of the 7th International Conference on Concurrency Theory*, volume 1119 of *Lecture Notes in Computer Science*, pages 98–114. Springer-Verlag, 1996.

[29] G. Winskel and M. Nielsen. Models for concurrency. In *Handbook of logic in computer science, Vol. 4*, Oxford Sci. Publ., pages 1–148. Oxford Univ. Press, 1995.

[30] O. Wyler. *Lecture notes on topoi and quasitopoi.* World Scientific Publishing Co. Inc., Teaneck, NJ, 1991.

Recent BRICS Report Series Publications

RS-99-36 **Gian Luca Cattani and Glynn Winskel. *Presheaf Models for CCS-like Languages*. November 1999. ii+46 pp.**

RS-99-35 **Tibor Jordán and Zoltán Szigeti. *Detachments Preserving Local Edge-Connectivity of Graphs*. November 1999. 16 pp.**

RS-99-34 **Flemming Friche Rodler. *Wavelet Based 3D Compression for Very Large Volume Data Supporting Fast Random Access*. October 1999. 36 pp.**

RS-99-33 **Luca Aceto, Zoltán Ésik, and Anna Ingólfsdóttir. *The Max-Plus Algebra of the Natural Numbers has no Finite Equational Basis*. October 1999. 25 pp. To appear in *Theoretical Computer Science*.**

RS-99-32 **Luca Aceto and François Laroussinie. *Is your Model Checker on Time? — On the Complexity of Model Checking for Timed Modal Logics*. October 1999. 11 pp. Appears in Kutyłowski, Pacholski and Wierzbicki, editors, *Mathematical Foundations of Computer Science: 24th International Symposium*, MFCS '99 Proceedings, LNCS 1672, 1999, pages 125–136.**

RS-99-31 **Ulrich Kohlenbach. *Foundational and Mathematical Uses of Higher Types*. September 1999. 34 pp.**

RS-99-30 **Luca Aceto, Willem Jan Fokkink, and Chris Verhoef. *Structural Operational Semantics*. September 1999. 128 pp. To appear in Bergstra, Ponse and Smolka, editors, *Handbook of Process Algebra*, 1999.**

RS-99-29 **Søren Riis. *A Complexity Gap for Tree-Resolution*. September 1999. 33 pp.**

RS-99-28 **Thomas Troels Hildebrandt. *A Fully Abstract Presheaf Semantics of SCCS with Finite Delay*. September 1999. 37 pp. To appear in *Category Theory and Computer Science: 8th International Conference*, CTCS '99 Proceedings, ENTCS, 1999.**

RS-99-27 **Olivier Danvy and Ulrik P. Schultz. *Lambda-Dropping: Transforming Recursive Equations into Programs with Block Structure*. September 1999. 57 pp. To appear in the November 2000 issue of *Theoretical Computer Science*. This revised report supersedes the earlier BRICS report RS-98-54.**