Presheaf Models for Concurrency

Gian Luca Cattani and Glynn Winskel

BRICS*, Computer Science Department, Aarhus University, Denmark

Abstract. This paper studies presheaf models for concurrent computation. An aim is to harness the general machinery around presheaves for the purposes of process calculi. Traditional models like synchronisation trees and event structures have been shown to embed fully and faithfully in particular presheaf models in such a way that bisimulation, expressed through the presence of a span of open maps, is conserved. As is shown in the work of Joyal and Moerdijk, presheaves are rich in constructions which preserve open maps, and so bisimulation, by arguments of a very general nature. This paper contributes similar results but biased towards questions of bisimulation in process calculi. It is concerned with modelling process constructions on presheaves, showing these preserve open maps, and with transferring such results to traditional models for processes. One new result here is that a wide range of left Kan extensions, between categories of presheaves, preserve open maps. As a corollary, this also implies that any colimit-preserving functor between presheaf categories preserves open maps. A particular left Kan extension is shown to coincide with a refinement operation on event structures. A broad class of presheaf models is proposed for a general process calculus. General arguments are given for why the operations of a presheaf model preserve open maps and why for specific presheaf models the operations coincide with those of traditional models.

1 Introduction

The paper [8] presented an abstract view of bisimulation, inspired by [7], applicable to models for concurrency presented as categories, following the lines of [17]. 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 labelled event structures, and later on Petri nets [12], in which concurrency or parallelism of actions is expressed by some relation of independence.

This paper takes up the suggestion of [8] 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, they are an intuitively appealing model of nondeterministic computation. Starting with a category of path objects (or observations) in which morphisms stand for an

* **B**asic **R**esearch **in** **C**omputer **S**cience, Centre of the Danish National Research Foundation.

extension of one path by another, nondeterministic computations are represented essentially by gluing together computation paths in a manner reminiscient of the way a powerdomain is built from a domain as a completion of its finite elements. More accurately, forming presheaves is equivalent to adjoining all colimits to a category, which corresponds to more than just adding directed colimits—the reason why nondeterministic branching is also introduced.

As was argued in [8] 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. 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 in [9]. To highlight the gain of working at a more abstract level than is common in concurrency theory, we mention an important result of this paper, Lemma 3, which shows that a broad class of operations, obtained as left Kan extensions, automatically preserve open maps. In particular, it specialises to show that the refinement, obtained as a Kan extension, preserves open maps and so bisimulation. Lemma 3 can also be read as saying that any colimit-preserving functor between two presheaf categories preserves open maps (see Corollary 4).

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, through results like Lemma 6 and Proposition 19, transfer these preservation properties to concrete models like synchronisation trees and event structures.[1] Working with presheaves also avoids some obstructions to a treatment of weak bisimulation on independence models, 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" [13,5]. The paper [16] is a further step in this programme.

[1] The paper [4] shows that any "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 [8].

2 Traditional Models

We focus on three traditional models for concurrency: transition systems, synchronisation trees and event structures (see [17] 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*. As usual, a transition (s, a, s') is written as $s \xrightarrow{a} s'$.

Let

$$T_0 = (S_0, i_0, L_0, tran_0) \text{ and } T_1 = (S_1, i_1, L_1, tran_1)$$

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
- $\lambda : L_0 \to_* L_1$, a partial function[2], which together satisfy

$$\begin{aligned} &(s, a, s') \in tran_0 \;\&\; \lambda(a) \text{ defined} \\ &\qquad\qquad\qquad \Rightarrow (\sigma(s), \lambda(a), \sigma(s')) \in tran_1, \text{ and} \\ &(s, a, s') \in tran_0 \;\&\; \lambda(a) \text{ undefined} \Rightarrow \sigma(s) = \sigma(s'). \end{aligned}$$

Transition systems with morphisms form a category TS in which the composition of morphisms is defined componentwise.

A *synchronisation tree* is a transition system whose graph has the form of a tree with root the initial state. We write ST for the subcategory 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 $(E, \leq, Con, l)$ consisting of a set E, of *events* which are partially ordered by $\leq$, the *causal dependency relation*, a *consistency relation* Con consisting of finite subsets of events, and a *labelling function* $l : E \to L$, which satisfy

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

[2] We treat partial functions in the same way as in [17] using $*$ to represent undefined.

for all events e, e' and their subsets X, Y.

Two events $e, e' \in E$ are said to be *concurrent* (causally independent) iff

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

A set, x, of events in E is said to be a *configuration* if it is

downwards-closed: $\forall e, e'.\ e' \leq e \in x \Rightarrow e' \in x$, and

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

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,
$\eta : E \to_* E'$ is a partial function on events, $\lambda : L \to_* L'$ 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 $e_1 = e_2$.

Let ES be the category of event structures with morphisms, as above, composed componentwise. The definition of morphism on event structures is given rather abruptly—see [17] for motivation.

The categories TS, ST and ES are rich in categorical constructions. In particular, all the functors projecting to labelling sets in $\mathbf{Set}_*$, the category of sets with partial functions, are cofibrations. While the projections to $\mathbf{Set}_*$ associated with TS and ST are fibrations, the projection of ES is not; there are cartesian liftings of total functions but not of strictly partial functions.

The categories TS, ST and ES are related by coreflections: the inclusion functor $\mathsf{ST} \hookrightarrow \mathsf{TS}$ has a right adjoint unfolding transition systems to trees; the functor $\mathsf{ST} \to \mathsf{ES}$ identifying a synchronisation tree with an event structure has a right adjoint serialising an event structure to a synchronisation tree. The coreflections cut down to coreflections between the fibres. Preservation properties of adjoints help in relating semantics in the different models. We normally refer to a fibre over some set L, by adding the subscript L to the name of the category, e.g., ES_L stand for the subcategory of those event structures whose set of labels is L and whose morphisms (η, λ) always have $\lambda = id_L$. The categories of models have products, central in giving semantics to parallel compositions. Because the projection $\mathsf{TS} \to \mathbf{Set}_*$ forms a fibration, products in TS are expressible as products in the fibre over the product of their labelling sets in the base category $\mathbf{Set}_*$. Such a decomposition of product is not possible in ES because there are not cartesian liftings of (partial) projections like $L \times_* M \to_* M$.

Constructions in the categories support a process language **Proc**. Its syntax is given by

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

where a is a label, Λ is a subset of labels and Ξ is a total function from labels to labels. All the operations, apart from prefixing, at, have a categorical status, as being built out of universal constructions. Most simply, the term nil denotes the initial object, $t_0 \times t_1$ the product of those objects denoted by t_0, t_1, and recursive definitions $rec\ x.t$ are treated through the help of ω-colimits. The other operations make use of the categories being fibred over $\mathbf{Set}_*$. Restriction $t\lceil\Lambda$ is defined via cartesian lifting over an inclusion associated with Λ, and relabelling $t\{\Xi\}$ via cocartesian liftings over Ξ. The sum of two processes $t_0 \oplus t_1$ is defined by a fibre coproduct of the result of sending them over a union of their labelling sets by cocartesian liftings. See [17] for details.

3 Bisimulation from Open Maps

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 takes the shape of a string of labels in L. Strings L^* form a (partial order) category in which a morphism represents the extension of one string s to another s' (s is an initial prefix of s'). It is convenient to identify strings L^* with the subcategory of ST consisting of those special synchronisation trees consisting of a single branch. To take account of the added independence structure of event structures, the shape of their computation paths is taken to be a finite *pomset*. The category Pom_L is taken to be the subcategory of ES_L, for a labelling set L, consisting of those finite event structures in which all finite subsets of events are in the consistency relation.

We can obtain a general definition of bisimulation from *open maps*[3] , which roughly speaking are morphisms with the property that any extension of a computation path in the range can be matched by an extension in its domain.

Assume a category of models M (this can be a fibre in any of the categories of models we are considering) and a choice of path category, a subcategory $\mathsf{P} \hookrightarrow \mathsf{M}$ consisting of path objects (these could be branches, or pomsets) together with morphisms expressing how they can be extended.

Let $f : X \to Y$ be an arrow in M. We say that f is P-*open* if for every commuting diagram

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

[3] The notion of open map has been originally developed by Joyal and Moerdijk (cf. [7]) and its application to concurrency to express bisimulation was developed in [8].

with m a morphism in P, there is a morphism p' such that in the diagram

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

the two "triangles" commute. This can be understood as meaning that the path p can be extended via m to a path p' in X which matches q.

Two objects X_1, X_2 of M are said to be P-*bisimilar* iff there is a *span* of P-open morphisms f_1, f_2:

$$\begin{tikzcd} & X \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\ X_1 & & X_2 \end{tikzcd}$$

In the case of traditional models we obtain known equivalences: for transition systems TS_L or synchronisation trees ST_L, L^*-bisimulation coincides with Park and Milner's strong bisimulation; for event structures ES_L, Pom_L bisimulation coincides with strong history-preserving bisimulation due to Bednarczyk refining ideas of van Glabbeek and Goltz, Rabinovitch and Traktenbrot [2,9,14].

4 Presheaf Models

Let P be a path category. The category of *presheaves over* P, often denoted by $\widehat{\mathsf{P}}$ or by $\mathsf{Set}^{\mathsf{P}^{op}}$, is the category whose objects are contravariant functors from P to Set (the category of sets and functions) and whose arrows are the natural transformations between such functors. The category $\widehat{\mathsf{P}}$ is the "free cocompletion" of P, in the sense that it can be obtained to within equivalence of categories, by freely adding all possible colimits of diagrams in P. The category $\widehat{\mathsf{P}}$ 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 [11]).

A category of presheaves, $\widehat{\mathsf{P}}$, is accompanied by a functor, the *Yoneda embedding*, $y_{\mathsf{P}} : \mathsf{P} \to \widehat{\mathsf{P}}$, which fully and faithfully embeds P in the category of presheaves. Given any object P of P, the presheaf $y_{\mathsf{P}}(P)$ is the contravariant hom-functor, $\mathsf{P}[-, P]$, which to any Q, an object of P, associates the set $\mathsf{P}[Q, P]$ of arrows of 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 P essentially as a full subcategory of $\widehat{\mathsf{P}}$, and we have a situation fitting that needed for defining open maps and bisimulation in $\widehat{\mathsf{P}}$, along the lines of Section 3. The Yoneda Lemma (see e.g. [10]) provides a natural bijection between $\widehat{\mathsf{P}}[y_{\mathsf{P}}(P), F]$ and $F(P)$. This justifies the intuition that a presheaf $F : \mathsf{P}^{op} \to \mathbf{Set}$ 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 P to give a function $F(P) \leftarrow F(Q) : F(m)$ saying how Q-paths restrict to P-paths.

From the Yoneda Lemma one obtains directly an alternative characterisation of open maps in a presheaf category $\widehat{\mathsf{P}}$. A morphism $h : F \to G$, between presheaves F, G, is *open* iff for all morphisms $m : P \to Q$ in $\mathbf{P}$, the square

$$\begin{tikzcd} F(P) \arrow[d, "h_P"'] & F(Q) \arrow[l, "Fm"'] \arrow[d, "h_Q"] \\ G(P) & G(Q) \arrow[l, "Gm"] \end{tikzcd}$$

is a quasi-pullback, i.e. whenever $p \in F(P)$ and $q \in G(Q)$ satisfy $h_P(p) = (Gm)(q)$, then there exists $p' \in F(Q)$ such that $(Fm)(p') = p$ and $h_Q(p') = q$.

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

As remarked in [8], the canonical functors are full and faithful embeddings (we will often refer to them in the following as *canonical embeddings*) for synchronisation trees and event structures with respect to appropriate path categories; this follows by a general result of [10], P. 243, because the path categories are dense in the categories of these models.

Theorem 1.

(i) The canonical functor from ST_L *to* $\widehat{L^*}$ *is full, faithful and dense.*

(ii) The canonical functor from ES_L *to* $\widehat{\mathsf{Pom}_L}$ *is full, faithful and dense.*

In the situation where the path category P has an initial object 0, a *rooted presheaf* is a presheaf F in which $F(0)$ is a singleton. The full subcategory of rooted presheaves of $\widehat{L^*}$ is equivalent to the category ST_L. In fact, it is not hard to see that a presheaf X in $\widehat{L^*}$ corresponds to a collection of synchronisation trees in which the set of "roots" corresponds bijectively to the set $X(\epsilon)$, where ϵ is the empty string, the initial object in L^*. While the canonical functor from ES_L always yields a rooted presheaf, not all rooted presheaves in $\widehat{\mathsf{Pom}_L}$ are obtained in this way. Full subcategories of rooted presheaves play an important role in our approach. Bisimulation in the subcategories of rooted presheaves coincides with bisimulation in the categories of concrete models:[4]

[4] Alternatively, one can restrict to surjective open maps in the full category of presheaves to obtain a similar correspondence.

Proposition 2.

(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 open maps in the full subcategory of rooted presheaves of $\widehat{L^*}$.*

(ii) *Two event structures, over labelling set L, are Pom_L-bisimilar (i.e. strong history-preserving bisimilar) iff their corresponding presheaves, under the canonical embedding, are related by a span of open maps in the full subcategory of rooted presheaves of $\widehat{\mathsf{Pom}_L}$.*

Kan extensions We consider here a powerful operation, which we use extensively.

Let P be a small category. We mentioned earlier that $\widehat{\mathsf{P}}$ is the free colimit completion of P. In more detail, the Yoneda embedding $y_{\mathsf{P}} : \mathsf{P} \to \widehat{\mathsf{P}}$ satisfies the universal property that for any functor $F : \mathsf{P} \to \mathcal{E}$, where $\mathcal{E}$ is a cocomplete category, there is a colimit-preserving functor $G : \widehat{\mathsf{P}} \to \mathcal{E}$, unique to within isomorphism, such that $F \cong G \circ y_{\mathsf{P}}$:

$$\begin{tikzcd} \mathsf{P} \arrow[r, "y_{\mathsf{P}}"] \arrow[dr, "F"'] & \widehat{\mathsf{P}} \arrow[d, "G"] \\ & \mathcal{E} \end{tikzcd}$$

The functor G is the[5] *left Kan extension of F along y_{P}*, and is written as $Lan_{y_{\mathsf{P}}}(F)$. In fact, $Lan_{y_{\mathsf{P}}}(F)$ is left adjoint to the functor taking Y in $\mathcal{E}$ to the presheaf $\mathcal{E}[F(\text{-}), Y]$. The functor $Lan_{y_{\mathsf{P}}}(F)$ is colimit-preserving, but a converse holds too: If $G : \widehat{\mathsf{P}} \to \widehat{\mathsf{Q}}$ is a colimit-preserving functor then, to within isomorphism, G is $Lan_{y_{\mathsf{P}}}(G \circ y_{\mathsf{P}})$. [6] Here we shall only be interested in left Kan extensions where $\mathcal{E}$ is a presheaf category $\widehat{\mathsf{Q}}$.

A general example of Kan extensions of which we will make great use is the following. Let P and Q be two small categories and let $F : \mathsf{P} \to \mathsf{Q}$ be a functor between them. Then, there exists a triple of adjoint functors $F_! \dashv F^* \dashv F_*$:

$$\begin{tikzcd} \widehat{\mathsf{P}} \arrow[r, bend left, "F_!"] \arrow[r, bend right, "F_*"'] & \widehat{\mathsf{Q}} \arrow[l, "F^*"'] \end{tikzcd}$$

[5] As usual in Category Theory, the notion of Kan extension is defined up to isomorphism.

[6] For more on Kan extensions and colimit-preserving functors, we refer to [3], Vol.1 and [10], Chapter 10.

The functor $F_!$ can be described as $Lan_{y_\mathsf{P}}(y_\mathsf{Q} \circ F)$. The functor F^* works by composition, i.e.,

$$F^*(X)(P) = X(FP) \qquad (F^*(\alpha))_P = \alpha_{FP}$$

for any X presheaf over Q and α arrow in $\widehat{\mathsf{Q}}$. The functor F_*, the right adjoint to F^*, is a right Kan extension, a concept dual to that of left Kan extension. From our earlier remarks, since F^* has a right adjoint, it is colimit preserving and therefore is itself a left Kan extension.

A key result of this paper is the following lemma which states that the left Kan extensions described above preserve open maps, a powerful tool in showing operations preserve bisimulation:

Lemma 3. *Let $F : \mathsf{P} \to \widehat{\mathsf{Q}}$ be a functor where P, Q are small categories. The left Kan extension $Lan_{y_P}F$ preserves open maps:*

If h is a P-open map in $\widehat{\mathsf{P}}$, then $(Lan_{y_P}F)(h)$ is a Q-open map in $\widehat{\mathsf{Q}}$.

Proof (Indication). Suppose h is an open map in $\widehat{\mathsf{P}}$. The proof proceeds by showing that the commuting squares associated with the naturality of $(Lan_{y_P}F)(h)$ are quasi-pullbacks. The application $((Lan_{y_P}F)(X))(Q)$, of $Lan_{y_P}F$ on a presheaf X at an object Q, is defined as a colimit in **Set** of a diagram on the category of elements of X (see e.g.[11]). The proof depends on the explicit presentation of such colimits as disjoint unions of sets subject to an equivalence relation, determined by how the diagram is connected by morphisms. In showing that the naturality squares of $(Lan_{y_P}F)(h)$ are quasi-pullbacks we use the fact that the functor between categories of elements, induced by the open map h, reflects connectivity by morphisms. $\square$

Since, as we said, all colimit-preserving functors G from a presheaf category $\widehat{\mathsf{P}}$ to a presheaf category $\widehat{\mathsf{Q}}$ are obtained to within natural isomorphism as left Kan extensions $Lan_{y_P}(G \circ y_P)$, we deduce the following general preservation property:

Corollary 4. *Assume $G : \widehat{\mathsf{P}} \to \widehat{\mathsf{Q}}$ is a colimit-preserving functor. If h is a P-open map in $\widehat{\mathsf{P}}$, then $G(h)$ is a Q-open map in $\widehat{\mathsf{Q}}$.*

In particular, in the case of a functor $F : \mathsf{P} \to \mathsf{Q}$ between small categories, we have the following:

Lemma 5. *Let $F : \mathsf{P} \to \mathsf{Q}$ be a functor between small categories. Let $F_! \dashv F^*$ be the adjunction described above between $\widehat{\mathsf{P}}$ and $\widehat{\mathsf{Q}}$. Then*

(i) If h is a P-open map, then $F_!(h)$ is Q-open map.
(ii) If h is a Q-open map, then $F^*(h)$ is P-open map.

If P and Q have initial objects then $F_!$ preserves rooted presheaves. Moreover, if F preserves the initial objects then F^* preserves rooted presheaves.

Thus we have the pleasing situation that open maps and bisimulation are preserved along both the directions of the adjunction $F_! \dashv F^*$.

As will be seen, Lemma 5 is useful in showing that bisimulation is a congruence with respect to the operations of process calculi. Some familiar adjunctions reappear as special instances. Suppose that F above is understood as the functor $L^* \to \mathsf{Pom}_L$ identifying a string with a (linear) pomset. The adjunction $F_! \dashv F^*$ extends, via the canonical embeddings, the familiar coreflection between synchronisation trees ST_L and event structures ES_L. Lemma 5 implies the intuitively clear fact that bisimulation is preserved by the inclusion of synchronisation trees in event structures, and that the right adjoint serialising an event structure to a synchronisation tree preserves bisimulation.

To illustrate the power of these general results we consider a form of refinement on event structures, which arises as a left Kan extension. One well-known operation of refinement on event structures is that of van Glabbeek and Goltz (refer to [9] for their definition) where a function θ "refining" labels in L to finite pomsets over M is extended to an operation $R(\theta)$ on event structures—roughly, a copy of the pomset $\theta(a)$ is plugged in for each occurrence of the label a and the pomsets' events inherit causal dependency and conflict from the host event structure. Such a refining function θ extends in the same way to a functor $R(\theta)' : \mathsf{Pom}_L \to \mathsf{Pom}_M$. As remarked in [8], the functor $R(\theta)'_!$, 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 $R(\theta)'_!$ act like the operation of refinement $R(\theta)$ on event structures? More precisely, if we let $c_L : \mathsf{ES}_L \to_* \widehat{\mathsf{Pom}}_L$ and $c_M : \mathsf{ES}_M \to_* \widehat{\mathsf{Pom}}_M$ denote the canonical embeddings, do we have

$$R(\theta)'_!(c_L(E)) \cong c_M(R(\theta)(E)) \ ?$$

Yes, by the following general lemma (instantiate $R(\theta)$ for F and read Pom_L for P_L, ES_L for E_L, *etc.*):

Notation: For any category C, we denote its class of objects by $|\,\mathsf{C}\,|$.

Lemma 6. *Let P_L and P_M be two small categories. Let $i_L : \mathsf{P}_L \to \mathsf{E}_L$ and $i_M : \mathsf{P}_M \to \mathsf{E}_M$ be two dense full and faithful embeddings. Let $c_L : \mathsf{E}_L \to \widehat{\mathsf{P}_L}$ and $c_M : \mathsf{E}_M \to \widehat{\mathsf{P}_M}$ be the associated canonical embeddings.*
Suppose $F : \mathsf{E}_L \to \mathsf{E}_M$ is a functor. Let $F' = F \circ i_L$. Then

$$Lan_{y_L}(c_M \circ F') \circ c_L = c_M \circ F$$

if for any $E \in |\,\mathsf{E}_L\,|$ and $Q \xrightarrow{g} FE$, there exist $P \in |\,\mathsf{P}_L\,|$, $g_Q : Q \to F'P$ and $f_P : i_L(P) \to E$ with $g = Ff_P \circ g_Q$ and such that for any other factorisation $g = Ff_{P'} \circ g'_Q$ there exists an $\alpha : P \to P'$ such that $f'_P \circ \alpha = f_P$ and $g'_Q = F_{\mathsf{P}}\alpha \circ g_Q$:

$$\begin{tikzcd}
Q \arrow[rr, "g"] \arrow[dr, "g_Q"] \arrow[ddr, "g'_Q"'] & & F(E) \\
& FP \arrow[ur, "F(f_P)"] \arrow[d, dotted, "F(\alpha)"] & \\
& FP' \arrow[uur, "F(f_{P'})"'] &
\end{tikzcd}$$

Remark: The "factorisation" condition of the lemma implies that the functor $F : \mathsf{E}_L \to \mathsf{E}_M$ preserves the canonical colimits associated with the objects of E_L (which is equivalent to the conclusion of the lemma). The "factorisation" condition can be weakened to a necessary condition by requiring that any two factorisations are connected by a chain of morphisms like α above.

As a left Kan extension, the operation $R(\theta)_!$ on presheaves preserves open maps and so bisimulation by Lemma 5, and consequently so does refinement $R(\theta)$ on event structures—an example where we transfer abstract, general properties of presheaves to a concrete model.

5 Presheaf Models for Proc

In giving a presheaf semantics to **Proc** we must address a minor clumsiness. Terms of **Proc** can have different labelling sets, whereas the presheaf categories up to now have been with respect to a particular labelling set. However we can "glue" all the presheaf categories together using the Grothendieck-fibration construction (see [3], vol.2).

To emphasise its generality, we give semantics to **Proc** with respect to a general class of presheaf models defined as follows.

A *presheaf model* for **Proc** consists of a functor $\mathsf{P}_{(-)}$ from $\mathbf{Set}_*$ to $\mathbf{Cat}$, the category of small categories, which sends $\lambda : L \to_* M$ to $\bar{\lambda} : \mathsf{P}_L \to \mathsf{P}_M$ such that:

- For each set L, the category P_L has an initial object; the functors $\bar{\lambda}$, for $\lambda : L \to_* M$, preserve initial objects.
- For each set L and element a, there is an explicitly given *prefixing* functor $a(-) : \mathsf{P}_L \to \mathsf{P}_{L\cup\{a\}}$.

A process with labelling set L is to denote a rooted presheaf over P_L. Of course, there are further conditions that one could expect a presheaf model to satisfy, but this definition suffices for our purposes.

5.1 The Grothendieck Construction

Given a presheaf model on $\mathsf{P}_{(-)}$, we can glue together all the fibres, consisting of categories of rooted presheaves over P_L, to form a fibration over $\mathbf{Set}_*$ which we call $Groth(\mathsf{P}_{(-)})$:

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

Arrows: pairs $\langle f, \lambda\rangle : \langle X, L\rangle \to \langle Y, M\rangle$ with $\lambda : L \to_* M$ and $f : X \to \bar{\lambda}^*(Y)$.

The composition of arrows is $\langle g, \mu\rangle \circ \langle f, \lambda\rangle = \langle \bar{\lambda}^*(g) \circ f, \mu \circ \lambda\rangle$. Clearly the projection $\langle X, L\rangle \mapsto L$ is the object part of a functor $\pi : Groth(\mathsf{P}_{(-)}) \to \mathbf{Set}_*$. 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.

Because of Lemma 5, for $\lambda : L \to_* M$ we have an adjunction $\bar{\lambda}_! \dashv \bar{\lambda}^*$ between presheaf categories $\widehat{P}_L$ and $\widehat{P}_M$ which cuts down to an adjunction between the fibres of rooted presheaves. The adjunctions ensure that the Grothendieck fibration is in fact a bifibration; the cocartesian lifting of λ with respect to X is $(\eta_X, \lambda) : X \to \bar{\lambda}_!(X)$ where $\eta_X : X \to \bar{\lambda}^*\bar{\lambda}_!(X)$ is the component of the unit of the adjunction at X.

By Lemma 5, the adjunctions preserve open maps across the fibres. Within the fibres the product and coproduct preserve open maps—by an elementary argument in the case of product, and making use of the disjointness of the coproduct of rooted presheaves.

Lemma 7. *For $\lambda : L \to_* M$, the functors $\bar{\lambda}_!$ and $\bar{\lambda}^*$ preserve open maps in the fibres of $Groth(\mathsf{P}_{(-)})$.*

Product and coproduct functors within the fibres of $Groth(\mathsf{P}_{(-)})$ preserve open maps.

5.2 Semantic Constructions in $Groth(\mathsf{P}_{(-)})$

As a preliminary to giving a presheaf semantics to **Proc**, we describe the main constructions involved and show that they preserve open maps.

Products: The category $Groth(\mathsf{P}_{(-)})$ has products. Product can be constructed as follows. Given $\langle X, L\rangle, \langle Y, M\rangle \in |Groth(\mathsf{P}_{(-)})|$. Define

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

where $L \overset{\pi_L}{{}_*\!\leftarrow} L \times_* M \overset{\pi_M}{\to_*} M$ are the projections of the product in $\mathbf{Set}_*$. Because the product in $Groth(\mathsf{P}_{(-)})$ decomposes into functors preserving open maps by Lemma 7, we see that:

Proposition 8. *The functor $\times$ preserves open maps; if f is an open map in $\widehat{\mathsf{P}}_L$ and g is an open map in $\widehat{\mathsf{P}}_M$, then $f \times g$ is an open map in $\widehat{\mathsf{P}_{L\times_* M}}$.*

Sum: Let $\langle X, L\rangle, \langle Y, M\rangle \in |Groth(\mathsf{P}_{(-)})|$. Define

$$\langle X, L\rangle \oplus \langle Y, M\rangle = \langle \overline{i_{L!}}(X) + \overline{i_{M!}}(Y), L \cup M\rangle$$

where $L \overset{i_L}{\to} L \cup M \overset{i_M}{\leftarrow} M$ are the injections of the coproduct in $\mathbf{Set}_*$. Sum is built up from open-map preserving functors by Lemma 7, so:

Proposition 9. *The functor $\oplus$ preserves open maps.*

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

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

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

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

By Lemma 7, restriction preserves open maps:

Proposition 10. *If $f : X \to Y$ is an open map in the fibre over L, then $f{\restriction}\Lambda : X{\restriction}\Lambda \to Y{\restriction}\Lambda$ is open in the fibre over $\Lambda \cap L$.*

Relabelling: Let $\Xi : L \to M$ be total. Take $\langle X, N\rangle$ as usual, define $\Xi_N : N \to M \cup N$ with

$$\Xi_N(x) = \begin{cases} \Xi(x) & \text{if } x \in L \\ x & \text{otherwise} \end{cases}$$

Consider the truncation of Ξ_N to its image set, i.e. $\Xi_N : N \to \Xi_N N$. Define the relabelling to be

$$\langle X, N\rangle[\Xi] = \langle \overline{\Xi_{N!}}(X), \Xi_N N\rangle \ .$$

By Lemma 7, relabelling preserves open maps:

Proposition 11. *If $f : X \to Y$ is an open map in the fibre over L, then $f[\Xi] : X[\Xi] \to Y[\Xi]$ is open in the fibre over $\Xi_N N$.*

Prefixing: Suppose we have a label set L and an element a. By taking a left Kan extension we extend the prefixing functors to $a(-) : \widehat{\mathsf{P}}_L \to \widehat{\mathsf{P}}_{L\cup\{a\}}$. By Lemma 5:

Proposition 12. *If $f : X \to Y$ is open in the fibre over L, then $a(f) : a(X) \to a(Y)$ is open in the fibre over $L \cup \{a\}$.*

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

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

Here 0 is the unique, up to isomorphism, rooted presheaf over $\mathsf{P}_\emptyset$. Any $F^n(\langle 0, \emptyset\rangle)$ consists of a pair $\langle X_n, L_n\rangle$ with $X_n \in |\widehat{\mathsf{P}_{L_n}}|$, 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 $\widehat{\mathsf{P}_L}$ of all the cocartesian liftings of the X_n.

All our term constructors are continuous with respect to ω-chains, hence $\mathit{rec}(F)$ determines a fixed point. 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 $\mathit{rec}\ x.t$, $\mathit{rec}\ y.u$ from bisimulation between the open terms t and u. Such open terms give rise to endofunctors on $\mathit{Groth}(\mathsf{P}_{(-)})$. Thus, we start by extending the notion of open map, and therefore bisimulation, to functors.

Definition 13. Let α be a natural transformation between two functors whose codomain is $\mathit{Groth}(\mathsf{P}_{(-)})$. We say α is *open* if each of its components is open in a fibre $\widehat{\mathsf{P}_L}$.

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

Definition 14. Let F be an endofunctor of $Groth(\mathsf{P}_{(-)})$. Define $\omega_F : \omega \to Groth(\mathsf{P}_{(-)})$ as follows:

$$\omega_F(n) = F^n(0) \text{ and } \omega_F(n \leq m) = F^n(0_{F^{m-n}(0)})$$

where $0_{F^{m-n}(0)}$ is the unique arrow from the initial rooted presheaf to $F^{m-n}(0)$.

Proposition 15. *Let F, R be endofunctors of $Groth(\mathsf{P}_{(-)})$ 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.*

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

Proposition 16. *Let $\omega_F, \omega_R : \omega \to \widehat{\mathsf{P}}$ and $\omega_\alpha : \omega_R \dot{\to} \omega_F$, be as above, with ω_α open, then the arrow $\varinjlim \omega_\alpha : \varinjlim \omega_R \to \varinjlim \omega_F$, uniquely determined by the universal property of the colimit, is an open map in the fibre over the colimiting labelling set.*

Consequently, if two endofunctors F, G ranging over $Groth(\mathsf{P}_{(-)})$ 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 $Groth(\mathsf{P}_{(-)})$ wich preserves open maps by this section's 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.

5.3 Denotational Semantics

We define the denotational semantics, with respect to an environment

$$\rho : Vars \to |Groth(\mathsf{P}_{(-)})|,$$

inductively on the structure of the terms in **Proc**.

Nil: $[\![\mathbf{Nil}]\!]_\rho =$ the unique, up to isomorphism, rooted presheaf over $\mathsf{P}_\emptyset$.
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: Let Λ be a set. $[\![t_1 \restriction \Lambda]\!]_\rho = [\![t_1]\!]_\rho \restriction \Lambda$
Relabelling: Let $\Xi : L \to M$ be total. $[\![t_1[\Xi]]\!]_\rho = [\![t_1]\!]_\rho[\Xi]$
Prefixing: Let a be a label, then $[\![at]\!]_\rho = a([\![t]\!]_\rho)$
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

$$\begin{aligned} t^x_\rho : Groth(\mathsf{P}_{(-)}) &\to Groth(\mathsf{P}_{(-)}) \\ X &\mapsto [\![t]\!]_{\rho[X/x]} \end{aligned}$$

Define $[\![recx.t]\!]\rho = rec(t^x_\rho)$

All of the constructions of Section 5.2, those used in giving the semantics above, preserve open maps and so bisimulation. Hence, we can deduce that for any presheaf model, the bisimulation equivalence associated with spans of open maps is a congruence for the language **Proc**.

5.4 Concrete Models Revisited

The semantics and results specialise to the specific presheaf models on path categories L^* and Pom_L for sets L; as prefixing functors $a(-)$ we take the obvious prefixing of strings and pomsets by an occurrence of the element a. 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 $\mathsf{ST}_L \to \widehat{L^*}$ and $\mathsf{ES}_L \to \widehat{\mathsf{Pom}_L}$ extend to full and faithful embeddings from ES and ST to presheaf models by the next proposition. We give first some notation to facilitate its reading.

Notation: If $F : \mathsf{C} \to \mathsf{B}$ is a cofibration we write by $\alpha_!^{\mathsf{C}} : \mathsf{C}_{b'} \to \mathsf{C}_b$ for the functor which results from choosing a cocartesian arrow over $\alpha : b' \to b$, at every object of $\mathsf{C}_{b'}$. Hence we write $\alpha_{c'}^{\mathsf{C}}$ for the cocartesian arrow at c' projecting to α. For any other $f : c' \to c$ we write $f_{\alpha^{\mathsf{C}}}$ for the unique arrow such that $f = f_{\alpha^{\mathsf{C}}} \circ \alpha_{c'}^{\mathsf{C}}$

Proposition 17. *Let $F : \mathsf{C} \to \mathsf{B}$ and $G : \mathsf{D} \to \mathsf{B}$ be two cofibrations. For any $b \in |\,\mathsf{B}\,|$, let C_b and D_b be the fibres over b. Suppose that for any b, $i_b : \mathsf{C}_b \to \mathsf{D}_b$ is a full and faithful embedding and that for any $\alpha : b' \to b$ the following diagram commutes:*

$$\begin{tikzcd} \mathsf{C}_{b'} \arrow[r, "i'_b"] \arrow[d, "\alpha_!^{\mathsf{C}}"'] & \mathsf{D}_{b'} \arrow[d, "\alpha_!^{\mathsf{D}}"] \\ \mathsf{C}_b \arrow[r, "i_b"] & \mathsf{D}_b \end{tikzcd}$$

Then there is a full and faithful embedding $i : \mathsf{C} \to \mathsf{D}$ such that for any object $c \in |\,\mathsf{C}_b\,|$, $i(c) = i_b(c)$ and for any $f : c' \to c$ with $F(f) = b' \stackrel{\alpha}{\to} b$, $i(f) = i_b(f_{\alpha^{\mathsf{C}}}) \circ \alpha^{\mathsf{D}}_{i_b(c)}$

We have stated this result for cofibrations (rather than, equivalently, for fibrations) because event structures form a cofibration (and not a fibration) over $\mathbf{Set}_*$. The embeddings of event structures and synchronisation trees in the respective presheaf categories satisfy the property of the proposition above. Hence, specialising to the case of event structures:

Corollary 18. *The embedding $c : \mathsf{ES} \to Groth(\mathsf{Pom}_{(-)})$, given in Proposition 17 is dense, full and faithful.*

It follows that $c : \mathsf{ES} \to Groth(\mathsf{Pom}_{(-)})$ preserves limits ([18] P. 21) and so products, and it can be checked that coproducts are preserved, too.

Proposition 19. *The embedding of $c : \mathsf{ES} \to Groth(\mathsf{Pom}_{(-)})$ preserves products, coproducts, cocartesian liftings, and cartesian liftings of inclusions.*

A denotational semantics of **Proc** in event structures ES is given in [17]. Denotations are built up in the same way as in Section 5.2, using the same universal constructions; for example the denotation of $t_1 \times t_2$ is built up as a product in ES, while the denotation of a restriction is got from a cartesian lifting. With the help of Proposition 19 we can show that the embedding c preserves the semantics of **Proc**. The only construction needing a separate treatment is prefixing. We can prove that the "standard" event-structures semantics of **Proc** in [17]—write $\mathsf{ES}\,[\![t]\!]$ for the denotation of a term t as in event structure, coincides with that in the presheaf model on Pom_L—write $[\![t]\!]$ for the denotation of a term t in presheaves over pomsets. The proof is straightforward (but for some technicalities in the treatment of recursion).

Theorem 20. *Let $\rho : Vars \to |\,\mathsf{ES}\,|$ be an environment function. Then for any term t of* **Proc**

$$c(\mathsf{ES}\,[\![t]\!]_\rho) \cong [\![t]\!]_{c\circ\rho}\ .$$

By Proposition 2(ii), open maps and bisimulation coincide, via the canonical embeddings, in ES_L and the fibre over L in $Groth(\mathsf{Pom}_{(-)})$. Hence we can transfer the congruence property deduced for the presheaf semantics to deduce, in particular, that strong history-preserving bisimulation is a congruence for the language **Proc**.

6 Relations with Domain Theory

One hope in moving to more abstract presheaf models is to be able to relate the theory of concurrency with the theory of domains, though this has to be understood in a generalised sense, in which domains can be proper categories, not just complete partial orders.

Finitely accessible categories [1], the category-analogue of algebraic complete partial orders, are obvious candidates for such generalised domains; roughly a finitely accessible category has a "basis" of "finite" (finitely presentable) objects from which every object is obtainable as a directed colimit. As proposed in [16], one can view the presheaf construction on the basis of a finitely accessible category as a form of powerdomain construction. In [16] it is indicated how the bicategory Prof of profunctors (or distributors, in the terminology of [3], vol.1) provides a framework in which one can solve domain equations for path categories. The bicategory of profunctors, or equivalently, the category where the objects are presheaf categories and the morphisms are colimit-preserving functors, should be thought of as a category of nondeterministic domains.

The framework in [16] is applied to give a denotational semantics to a value-passing process language with late semantics. More recently Ian Stark and the authors have been combining these results with those of [15,6] to produce presheaf models for the π-calculus with both early and late semantics. It is also possible to give a presheaf semantics to process languages which permit processes to be passed, and not just discrete values or names, though in this case we do not yet have a good understanding of the notion of bisimulation induced by open maps—in the other cases mentioned bisimulation in the model coincides with a traditional operational definition and is well understood. So far all the presheaf models coping with higher-order features are based on an "interleaving" treatment of concurrency, as presently we cannot see how to combine facility at higher-order with independence of the kind found in event structures and Petri nets.

References

1. Adámek, J., and Rosicky, J., *Locally Presentable and Accessible Categories.* London Mathematical Society Lecture Note Series, 189. Cambridge University Press, 1994.
2. Bednarczyk, M., *Hereditary history preserving bisimulation or what is the power of the future perfect in program logics.* Technical report, Polish Academy of Sciences, Gdansk, 1991.
3. Borceux, F., *Handbook of Categorical Algebra, vol.1,2.* Encyclopedia of Mathematics and its Applications, vol.50,51. Cambridge University Press, 1994.
4. Cheng, A., and Nielsen, M., *Open Maps, Behavioural Equivalences, and Congruences.* Report Series RS-96-2, BRICS, University of Aarhus. Denmark, January 1996.
5. Fiore, M.P., *Axiomatic Domain Theory in Categories of Partial Maps.* Ph.D Thesis. Distinguished Dissertations in Computer Science, Cambridge University Press, 1996.
6. Fiore, M.P., Moggi, E., and Sangiorgi, D., *A Fully-Abstract Model for the π-calculus* In Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science, pages. 43-54. IEEE Computer Society Press, 1996.
7. Joyal, A., and Moerdijk, I., *A completeness theorem for open maps.* In Annals of Pure and Applied Logic 70, 51-86, 1994.
8. Joyal, A., Nielsen, M., and Winskel, G., *Bisimulation from open maps.* In LICS '93 special issue of Information and Computation, vol.127, pp.164-185, 1996.
9. van Glabbeek, R.J., and Goltz, U., *Equivalence notions for concurrent systems and refinement of actions.* In Proceedings of Mathematical foundations of computer science 1989, pp.237–248, Lecture Notes in Computer Science vol.379, 1989.
10. Mac Lane, S., *Categories for the Working Mathematician.* Springer-Verlag, 1971.
11. Mac Lane, S., Moerdijk, I., *Sheaves in Geometry and Logic.* Springer-Verlag, 1992.
12. Nielsen, M., and Winskel, G., *Petri nets and bisimulations.* Theoretical Computer Science 153, pp.211-244, 1996.
13. Plotkin, G., *Algebraic Completeness and Compactness in an Enriched Setting.* Transparencies from the Invited Lecture given at the Workshop on Logic, Domains and Programming Languages. Darmstadt, 1995.
14. Rabinovitch, A., and Traktenbrot, B., *Behaviour structures and nets.* Fundamenta Informatica, 11(4), pp.357-404, 1988.
15. Stark, I., *A Fully Abstract Domain Model for the π-Calculus.* In Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science, pages. 36-42. IEEE Computer Society Press, 1996.
16. Winskel, G., *A Presheaf Semantics of Value-Passing Processes.* In Proceedings of the Seventh International Conference on Concurrency Theory: CONCUR'96, pp.98–114, Lecture Notes in Computer Science vol.1119, 1996.

17. Winskel, G., and Nielsen, M., *Models for Concurrency.* In the Handbook of Logic in Computer Science, vol.IV pp.1–148, ed. Abramsky, Gabbay and Maibaum, Oxford University Press, 1995.
18. Wyler, O., *Lecture Notes on Topoi and Quasitopoi.* World Scientific Publishing Co., 1991.

This article was processed using LATEXmacro package with LLNCS style.
Diagrams were drawn using Kristoffer Rose XY-pic package.