BRICS

Basic Research in Computer Science

A Presheaf Semantics of Value-Passing Processes

Glynn Winskel

BRICS Report Series RS-96-44

ISSN 0909-0878 November 1996

See back inner page for a list of recent publications in the BRICS Report Series. 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 World Wide Web and anonymous FTP:

`http://www.brics.dk/`
`ftp://ftp.brics.dk/pub/BRICS`
This document in subdirectory `RS/96/44/`

A presheaf semantics of value-passing processes

Glynn Winskel

BRICS* – Computer Science Dept., University of Aarhus

Abstract

This paper investigates presheaf models for process calculi with value passing. Denotational semantics in presheaf models are shown to correspond to operational semantics in that bisimulation obtained from open maps is proved to coincide with bisimulation as defined traditionally from the operational semantics. Both "early" and "late" semantics are considered, though the more interesting "late" semantics is emphasised. A presheaf model and denotational semantics is proposed for a language allowing process passing, though there remains the problem of relating the notion of bisimulation obtained from open maps to a more traditional definition from the operational semantics. A tentative beginning is made of a "domain theory" supporting presheaf models.

Introduction

The papers [12, 4] explore presheaf models for concurrency. Here begins an investigation of the use of presheaves to model higher-order features, most dramatic in the situation of process calculi where processes can be communicated as values.

Something of higher-order appears even in value-passing process calculi where values lie in some discrete datatype like integers or booleans. As is customary, for value-passing calculi, we draw a distinction between "early" and "late" semantics. *Early* semantics coincides with that presented in [14] where a value-passing calculus is reduced to a value-free one by immediately instantiating the variable in an input action to its possible values, the resulting processes being set together in a nondeterministic sum. According to *late* semantics input actions contain bound variables which only become instantiated when a communication is made. Generally (see e.g. [15, 6, 16]),

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

a late semantics for value passing represents the result of input communication as an abstraction, denoting a function from values to processes. Whereas the usual models for concurrency, transition systems, labelled Petri nets and event structures and the like, accommodate early semantics for value-passing directly, following [14, 7, 8], the late semantics seems accomplished most smoothly in domain-theoretic settings, which readily support abstractions.

Two ways seem open to extending models for concurrency to higher-order features. One is to take existing models, most of these transitions systems in one disguise or another, and essentially decorate them with extra structure. Another is to develop a new class of models, some of which can be seen to correspond to existing models, and which at the same time are rich enough to support constructions of the kind we are used to seeing in domain theory. This paper follows the latter course in investigating presheaf models.

Presheaf models for concurrency have the advantage of including interleaving models like synchronisation trees and independence models like labelled event structures, as well as contributing a general definition of bisimulation based on open maps. As we will see, they also extend to higher-order, though presently many questions remain, chief among them being the problem of simultaneously combining higher-order features with independence of the kind seen in event structures and Petri nets. A more specific problem is that of obtaining a characterisation in terms of the operational semantics of the bisimulation obtained from open maps for a process-passing calculus. On the positive side, the usual definition of "late bisimulation" and "early bisimulation" for ordinary value-passing is reconciled with the definition of bisimulation obtained on presheaves via open maps.

1 The language VProc

VProc is a process language for passing values along channels, inspired by CCS. Its syntax:

$$t ::= nil \mid \tau.t \mid a!e.t \mid a?x.t \mid t_1 \mid t_2 \mid t_1 + t_2 \mid [e_1 = e_2]t \mid X \mid recX.t$$

where x ranges over value-variables Var, X over process-variables $Pvar$, a over channel names C, and e, e_1, e_2 over value-expressions. We will not go into the details of the form of value-expressions beyond remarking that they may contain free value-variables and when evaluated yield values in a set V.

For simplicity we assume that recursive definitions of processes $recX.t$ are guarded in the sense that all free occurrences of X in t lie under, though not necessarily immediately under, a prefix τ.-, $a!e$.- or $a?x$.-.

1.1 Late transition semantics for VProc

We specify the transitions a closed term can perform. A transition $t \xrightarrow{\alpha} t'$, where t is a closed term, is understood to mean that the process t can perform action α to become t'; actions α range over τ-actions τ, output actions $a!v$, where $a \in C$ and $v \in V$, and input actions $a?x$, where $a \in C$ and $x \in Var$.

τ rule: $\tau.t \xrightarrow{\tau} t$

Output rule: $a!e.t \xrightarrow{a!v} t$

where e, necessarily closed, evaluates to value v.

Input rule: $a?x.t \xrightarrow{a?y} t[y/x]$

where $y \in Var$ is assumed not captured by its substitution for x in t.

Parallel rules:

$$\frac{t_1 \xrightarrow{\alpha} t_1'}{t_1|t_2 \xrightarrow{\alpha} t_1'|t_2} \qquad \frac{t_2 \xrightarrow{\alpha} t_2'}{t_1|t_2 \xrightarrow{\alpha} t_1|t_2'}$$

In the first parallel rule t_2 must have no free variables in common with action α; a symmetric condition is enforced for the second parallel rule.

$$\frac{t_1 \xrightarrow{a!v} t_1' \quad t_2 \xrightarrow{a?y} t_2'}{t_1|t_2 \xrightarrow{\tau} t_1'|t_2'[v/y]} \qquad \frac{t_1 \xrightarrow{a?y} t_1' \quad t_2 \xrightarrow{a!v} t_2'}{t_1|t_2 \xrightarrow{\tau} t_1'[v/y]|t_2'}$$

Sum rules:

$$\frac{t_1 \xrightarrow{\alpha} t_1'}{t_1 + t_2 \xrightarrow{\alpha} t_1'} \qquad \frac{t_2 \xrightarrow{\alpha} t_2'}{t_1 + t_2 \xrightarrow{\alpha} t_2'}$$

Condition rule:

$$\frac{t \xrightarrow{\alpha} t'}{[e_1 = e_2]t \xrightarrow{\alpha} t'}$$

provided e_1 and e_2 evaluate to the same value.

Recursion rule:

$$\frac{t[recX.t/X] \xrightarrow{\alpha} t'}{recX.t \xrightarrow{\alpha} t'}$$

1.2 Late bisimulation

Definition: A *late bisimulation* is a binary relation R between closed process terms such that whenever $t_1 R t_2$

(i) $t_1 \xrightarrow{\tau} t_1' \Rightarrow \exists t_2'.\ t_2 \xrightarrow{\tau} t_2' \ \&\ t_1'\ R\ t_2'$ and $t_2 \xrightarrow{\tau} t_2' \Rightarrow \exists t_1'.\ t_1 \xrightarrow{\tau} t_1' \ \&\ t_1'\ R\ t_2'$

(ii) $t_1 \xrightarrow{a!v} t_1' \Rightarrow \exists t_2'.\ t_2 \xrightarrow{a!v} t_2' \ \&\ t_1'\ R\ t_2'$ and $t_2 \xrightarrow{a!v} t_2' \Rightarrow \exists t_1'.\ t_1 \xrightarrow{a!v} t_1' \ \&\ t_1'\ R\ t_2'$

(iii) $t_1 \xrightarrow{a?y} t_1' \Rightarrow \exists t_2', z.\ t_2 \xrightarrow{a?z} t_2' \ \&\ \forall v \in V.\ t_1'[v/y]\ R\ t_2'[v/z]$ and
$t_2 \xrightarrow{a?y} t_2' \Rightarrow \exists t_1', z.\ t_1 \xrightarrow{a?z} t_1' \ \&\ \forall v \in V.\ t_1'[v/y]\ R\ t_2'[v/z]$.

Say closed process terms t_1, t_2 are *late bisimilar* iff there is a late bisimulation R such that $t_1 R t_2$.

2 Open maps and bisimulation on presheaves

Let $\mathbf{P}$ be a small category. It is to be thought of as a category of path objects (or path shapes) in which morphisms stand for an extension of one path by another. Let $\widehat{\mathbf{P}} = [\mathbf{P}^{op}, \mathbf{Set}]$, the category of presheaves over $\mathbf{P}$. Recall, a morphism $h : X \to Y$, between presheaves X, Y, is *open* iff for all morphisms $m : P \to Q$ in $\mathbf{P}$, the square

$$\begin{array}{ccc} X(P) & \xrightarrow{Xm} & X(Q) \\ {\scriptstyle h_P}\big\downarrow & & \big\downarrow{\scriptstyle h_Q} \\ Y(P) & \xrightarrow{Ym} & Y(Q) \end{array}$$

is a quasi-pullback, i.e. whenever $p \in X(P)$ and $q \in Y(Q)$ satisfy $h_P(p) = (Ym)(q)$, then there exists $p' \in X(Q)$ such that $(Xm)(p') = p$ and $h_Q(p') = q$. (This definition of open map, translates via the Yoneda Lemma to an equivalent path-lifting property of h—see [12].)

Say presheaves X, Y are *bisimilar* iff there is a span of surjective open maps between them, equivalently, iff there is $R \hookrightarrow X \times Y$ such that the compositions with the projections $R \hookrightarrow X \times Y \xrightarrow{\pi_1} X$ and $R \hookrightarrow X \times Y \xrightarrow{\pi_2} Y$ are surjective open.

In [12, 4] we defined bisimulation between *rooted* presheaves, presheaves X, over a category assumed to have an initial object I, for which $X(I)$ is a singleton. For rooted presheaves bisimulation is defined merely through the presence of a open maps (not requiring surjectivity). This is because open maps between rooted presheaves are necessarily surjective.

We can cast further light on rooted presheaves with the help of a "lifting" construction which will be important later, as is to be expected from traditional domain theory. For $\mathbf{P}$, a small category, define its *lifting* $\mathbf{P}_\perp$ to consist of $\mathbf{P}$ with a new initial object (called $\perp$) adjoined freely. Given $X \in \widehat{\mathbf{P}}$, define $\mathrm{lift}(X) \in \widehat{\mathbf{P}}_\perp$ to be the rooted presheaf which acts as X on copies of $P \in \mathbf{P}$ and yields a singleton, $\{*\}$ say, on $\perp$. The lift operation extends in the obvious way to a functor which gives an equivalence between $\widehat{\mathbf{P}}$ and the subcategory of rooted presheaves over $\mathbf{P}_\perp$; on maps h in $\widehat{\mathbf{P}}$, $\mathrm{lift}(h)$ is open iff h is surjective open. These remarks are useful in another context, that of algebraic set theory—see [10], p. 72.

3 A domain-theoretic setting

In proposing categories of presheaves as our "domains" of processes we are leaving domain theory as traditionally understood; processes are denoted by presheaves, objects in a category rather than elements of a partial order. This is not new; several proposals have been made for generalisation of powerdomains that leave the category of partial orders, for instance [13, 1, 18], and presheaves, being a way to introduce nondeterministic branching to computation paths, have much in common with powerdomains.

We sketch a setting, generalising traditional domain theory, in which we can place the work on presheaf models. The category analogue of algebraic cpo's is finitely accessible categories [2] in which the role of the basis of finite/isolated/compact elements is replaced by that of a small subcategory of finitely presentable objects; every object of a finitely accessible category is a directed colimit of finitely presentable objects. This is analogous to the fact that an algebraic cpo is the ideal completion of its finite elements. Morphisms between finitely accessible categories are functors preserving directed colimits, the analogue of continuous functions.

A way to introduce nondeterminism to a finitely accessible category $\mathcal{C}$ is via a construction on the "basis" of finitely presentable objects $\mathcal{C}^0$: Freely close $\mathcal{C}^0$ under all finite colimits to get a new basis (in which nondeterministic branching has been introduced). The finitely accessible category with this new category as basis, got by closing under directed colimits, can be thought of as the nondeterministic computations of $\mathcal{C}$. This "ideal completion" is equivalent to the category of presheaves over $\mathcal{C}^0$ (by results of [9], ch.VI). So taking presheaves combines two operations, adding branching to a basis (the part that takes us outside partial orders), and then completing to a finitely accessible category. Viewed in this way, taking presheaves over the basis of a finitely accessible category yields a "monad" on finitely accessible categories, reminiscent of powerdomain monads.[1]

The Kleisli category of the monad associated with taking presheaves is **Prof**, the bicategory of *profunctors* (see e.g. [3] where they are called *distributors*). Profunctors and their categorical constructions provide a convenient setting in which to provide semantics to process calculi with value and process passing. The bicategory **Prof** has small categories as objects and as morphisms $F : \mathbf{P} \nrightarrow \mathbf{Q}$, where $\mathbf{P}$ and $\mathbf{Q}$ are small categories, we take functors $F : \mathbf{P} \to \widehat{\mathbf{Q}}$. Composition in **Prof**, say of $F : \mathbf{P} \nrightarrow \mathbf{Q}$ and $G : \mathbf{Q} \nrightarrow \mathbf{R}$, is given to within isomorphism by $G^\dagger \circ F : \mathbf{P} \nrightarrow \mathbf{R}$—here $G^\dagger$ is the left Kan extension $\mathrm{Lan}_{y_Q} G$ of G with respect to the Yoneda embedding $y_Q : \mathbf{Q} \to \widehat{\mathbf{Q}}$. Left Kan extensions and so composition are only determined up to isomorphism; thus the fact that **Prof** is really a bicategory, and not a category. Note that profunctors, or more properly their left Kan extensions, preserve (surjective) open maps and so bisimulation by [4] Lemma 3—the extra preservation of surjectivity is easy to show. **Cat** the category of small categories embeds in **Prof**: A functor $F : \mathbf{P} \to \mathbf{Q}$ is sent to the composition $y_Q \circ F$ with the Yoneda embedding $y_Q : \mathbf{Q} \to \widehat{\mathbf{Q}}$. The embedding $\mathbf{Cat} \to \mathbf{Prof}$ preserves small colimits.

Prof forms a model of classical linear logic. To see its monoidal closed structure, for small categories $\mathbf{P}, \mathbf{Q}$, define

$$\mathbf{P} \multimap \mathbf{Q} = \mathbf{P}^{op} \times \mathbf{Q} \quad \text{and} \quad \mathbf{P} \otimes \mathbf{Q} = \mathbf{P} \times \mathbf{Q} \ ,$$

where product $\times$ on the right is the usual product of categories, and observe the natural bijection:

$$\mathbf{Prof}(\mathbf{P}, [\mathbf{Q} \multimap \mathbf{R}]) \cong \mathbf{Prof}(\mathbf{P} \otimes \mathbf{Q}, \mathbf{R})$$

The unit of $\otimes$ is $\mathbf{1}$, the category with a single object and morphism. **Prof** has products and coproducts which coincide on objects, where both are given by

[1] In this motivational section, we won't be distracted by the constructions more properly taking place in a 2-category/bicategory—thus the quotes around "monad".

coproduct in **Cat**. As a model of classical linear logic there is the same kind of degenerary familiar from the category of relations; par ($\wp$) coincides with tensor ($\otimes$), and $\perp$ with $\mathbf{1}$ (so **Prof** is compact-closed). Linear involution $\mathbf{P}^\perp$ is isomorphic to $\mathbf{P} \multimap \mathbf{1}$ and so to $\mathbf{P}^{op}$.

Morphisms $\mathbf{1} \nrightarrow (\mathbf{P} \multimap \mathbf{Q})$ correspond to presheaves over $\mathbf{P}^{op} \times \mathbf{Q}$ and so to profunctors $\mathbf{P} \nrightarrow \mathbf{Q}$. They correspond to colimit-preserving functors from $\widehat{\mathbf{P}}$ to $\widehat{\mathbf{Q}}$.

When we attend to presheaf semantics we are involved with various sorts of functors. Certainly we quickly encounter functors from $\mathbf{P}$ to $\widehat{\mathbf{Q}}$ corresponding, to within isomorphism, to colimit-preserving functors from $\widehat{\mathbf{P}}$ to $\widehat{\mathbf{Q}}$, between presheaves. We also meet more general "continuous" functors $\widehat{\mathbf{P}} \to \widehat{\mathbf{Q}}$, for example to cope with processes which can receive processes as values. As usual in linear logic we can recover these with the help of an exponential (!). Define $!\mathbf{P}$, to be a completion of $\mathbf{P}$ under finite colimits; more precisely we can take $!\mathbf{P}$ to be a skeletal subcategory of the subcategory of $\widehat{\mathbf{P}}$ consisting of finitely presentable objects. Then profunctors $!\mathbf{P} \nrightarrow \mathbf{Q}$ correspond, to within isomorphism, to functors $\widehat{\mathbf{P}} \to \widehat{\mathbf{Q}}$ which are continuous in the sense that they preserve directed colimits.

Prof provides us with a rich repertoire of constructions on categories of presheaves. We pause to ask how the constructions are reflected in notions of open maps and bisimulation.

It is clear when a map is (surjective) open in a coproduct $\mathbf{P} + \mathbf{Q}$ in **Prof**: $h : X \to Y$ is (surjective) open in $\widehat{\mathbf{P} + \mathbf{Q}}$ iff the two components $h_1 : X_1 \to Y_1$ and $h_2 : X_2 \to Y_2$ are (surjective) open in $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$ respectively. Because products in **Prof** are given by the same construction on objects, the same holds for products.

Let $h : X \to Y$ be a map in $\widehat{\mathbf{P} \otimes \mathbf{Q}}$. For $P \in \mathbf{P}$, define h^P to be the natural transformation $h^P : X(P, -) \to Y(P, -)$ with component $(h^P)_Q = h_{P,Q}$ at $Q \in \mathbf{Q}$. In a similar way, define $h^Q : X(-, Q) \to Y(-, Q)$ for any $Q \in \mathbf{Q}$. Now, we can observe: $h : X \to Y$ is (surjective) open in $\widehat{\mathbf{P} \otimes \mathbf{Q}}$ iff
$\forall P \in \mathbf{P}.\ h^P$ is (surjective) open in $\widehat{\mathbf{Q}}$ and $\forall Q \in \mathbf{Q}.\ h^Q$ is (surjective) open in $\widehat{\mathbf{P}}$.

There is a similar characterisation of open maps in $\widehat{\mathbf{P} \multimap \mathbf{Q}}$ because $\mathbf{P} \multimap \mathbf{Q} = \mathbf{P}^{op} \times \mathbf{Q}$. A map $h : X \to Y$ in $\widehat{\mathbf{P} \multimap \mathbf{Q}}$ is (surjective) open iff
$\forall P \in \mathbf{P}.\ h^P$ is (surjective) open in $\widehat{\mathbf{Q}}$ and $\forall Q \in \mathbf{Q}.\ h^Q$ is (surjective) open in $\widehat{\mathbf{P}^{op}}$.

Note openness and bisimilarity in $\widehat{\mathbf{P} \multimap \mathbf{Q}}$ involves openness and bisimilarity in $\widehat{\mathbf{P}^{op}}$! However, in the situation where $\mathbf{P}$ is a discrete category, h is open iff h^P is open for all $P \in \mathbf{P}$.

This proposal of a domain theoretic framework in which to understand presheaf models cannot be definitive at present. We would, for instance, expect to work within some cartesian-closed subcategory of finitely accessible categories. But, more importantly, until the aim of bringing independence models within a domain-theoretic framework is carried out fully we should remain open-minded.

4 A late path category

We seek a path category $\mathbf{P}$ with respect to which closed process terms of **VProc** denote presheaves. Its objects should reflect that a compution path of a process may begin with a τ-action, an output action $a!v$ or an input action $a?$, when it may either resume with a computation path, or, in the case where it has first performed an input action, input a value before resuming the computation path. This guides us to wishing to denote closed terms of **VProc** by presheaves over path category $\mathbf{P}$, which is an initial solution to

$$\mathbf{P} \cong \mathbf{P}_\perp + \sum_{(a,v)\in C\times V} \mathbf{P}_\perp + \sum_{a\in C} (V \multimap \mathbf{P})_\perp$$

in **Prof**—here we treat the set V as a discrete category. The solution is easy to construct, firstly because it is sufficient to find an initial solution to

$$\mathbf{P} \cong \mathbf{P}_\perp + \sum_{(a,v)\in C\times V} \mathbf{P}_\perp + \sum_{a\in C} (V^{op} \times \mathbf{P})_\perp$$

in **Cat** (where $V^{op} = V$ as V is discrete), and secondly because all the operations used preserve the property that the category is a partial order. This means an initial solution has the form of a partial order

$$\mathbf{P} = \mathbf{P}_\perp + \sum_{(a,v)\in C\times V} \mathbf{P}_\perp + \sum_{a\in c} (V^{op} \times \mathbf{P})_\perp$$

whose path objects are given inductively by:

- $\tau. \in \mathbf{P}$, and $\tau.P \in \mathbf{P}$ if $P \in \mathbf{P}$,

- $a!v. \in \mathbf{P}$, and $a!v.P \in \mathbf{P}$ if $P \in \mathbf{P}$,
- $a? \in \mathbf{P}$, and $a?(v \mapsto P) \in \mathbf{P}$ if $P \in \mathbf{P}$,

where $a \in C$ and $v \in V$, and whose morphisms (the partial order) are given inductively by the following clauses, where $P, P' \in \mathbf{P}, a \in C$ and $v \in V$:

- $P \leq P$,
- $\tau. \leq \tau.P$, and $\tau.P \leq \tau.P'$ if $P \leq P'$,
- $a!v. \leq a!v.P$, and $a!v.P \leq a!v.P'$ if $P \leq P'$,
- $a? \leq a?(v \mapsto P)$, and $a?(v \mapsto P) \leq a?(v \mapsto P')$ if $P \leq P'$.

Notation: We use (P, Q) to name the unique morphism from P to Q in $\mathbf{P}$ when $P \leq Q$.

We are using suggestive names for the objects of $\mathbf{P}$ to pick out to which component of a sum they belong:

- $\tau.$ is the least element of the leftmost summand of $\mathbf{P}$, other elements of this component being of the form $\tau.P$.
- $a!v.$ is the least element of the output summand associated with outputting value v on channel a; other elements of this component have the form $a!v.P$.
- $a?$ is the least element of the summand associated with a commitment to input on channel a; its other elements take the form $a?(v \mapsto P)$ and correspond to resuming a computation path after inputting value v.

We could have derived the above constructions on path objects systematically from operations associated with sums, lifting and product of categories.

5 Late presheaf semantics

We introduce operations on presheaves which capture the meaning of operations in **VProc**.

5.1 Prefixing

Let $X \in \widehat{\mathbf{P}}$. We define $\tau.X \in \widehat{\mathbf{P}}$ by taking $\tau.X = In_\tau \circ \mathrm{lift}(X)$. where $In_\tau : \widehat{\mathbf{P}}_\perp \to \widehat{\mathbf{P}}$ takes a presheaf over $\mathbf{P}_\perp$ to the corresponding presheaf over the left summand $\mathbf{P}_\perp$ in

$$\mathbf{P} = \mathbf{P}_\perp + \sum_{(a,v)\in C\times V} \mathbf{P}_\perp + \sum_{a\in C} (V^{op} \times \mathbf{P})_\perp \, . \qquad (\dagger)$$

Recalling our notation for path objects it follows that for $X \in \widehat{\mathbf{P}}$ and a path object $Q \in \mathbf{P}$

$$\tau.X(Q) \;=\; \begin{cases} X(P) & \text{if } Q = \tau.P, \\ \{*\} & \text{if } Q = \tau., \\ \emptyset & \text{otherwise.} \end{cases}$$

Similarly, for $X \in \widehat{\mathbf{P}}, a \in C$ and $v \in V$, we define $a!v.X \in \widehat{\mathbf{P}}$ so that on a path object $Q \in \mathbf{P}$

$$a!v.X(Q) \;=\; \begin{cases} X(P) & \text{if } Q = a!v.P, \\ \{*\} & \text{if } Q = a!v., \\ \emptyset & \text{otherwise.} \end{cases}$$

Let $F : V \to \widehat{\mathbf{P}}$ and $a \in C$. We define $a?F \in \widehat{\mathbf{P}}$ as follows. First notice that F corresponds to a presheaf X over $V^{op} \times \mathbf{P}$, and now define $a?F = In_{a?} \circ \mathrm{lift}(X)$ where $In_{a?} : \widehat{\mathbf{P}}_\perp \to \widehat{\mathbf{P}}$ takes a presheaf over $(V^{op} \times \mathbf{P})_\perp$ to the corresponding presheaf over the a-summand in $\mathbf{P}$ (see (†) above). Now, for $F : V \to \widehat{\mathbf{P}}, a \in C$ and a path object $Q \in \mathbf{P}$ we obtain

$$a?F(Q) \;=\; \begin{cases} (Fv)(P) & \text{if } Q = a?(v \mapsto P), \\ \{*\} & \text{if } Q = a?, \\ \emptyset & \text{otherwise.} \end{cases}$$

Notation: If $G(v) \in \widehat{\mathbf{P}}$, for any $v \in V$, we can as usual write $\lambda v.G(v)$ for the associated function $V \to \widehat{\mathbf{P}}$. We write $a?v.G(v)$ for $a?(\lambda v.G(v))$.

5.2 Sums

Coproducts of presheaves provide nondeterministic sums of processes.
If $X_1, X_2, \cdots, X_n \in \widehat{\mathbf{P}}$, we use $X_1 + \cdots + X_n$ to denote the presheaf which at a path object $P \in \mathbf{P}$ takes the set-value

$$(X_1 + \cdots + X_n)(P) = X_1(P) + \cdots + X_n(P),$$

the disjoint union of sets $X_1(P), \cdots, X_n(P)$. For a morphism (P, Q) of $\mathbf{P}$, where $P \leq Q$,

$$(X_1, + \cdots + X_n)(P, Q) = X_1(P, Q) + \cdots + X_n(P, Q),$$

making use of the functorial nature of disjoint union (= coproduct) of sets.

Similarly, if $X_i, i \in I$, is an indexed family of presheaves $X_i \in \widehat{\mathbf{P}}$, we use $\sum_{i \in I} X_i$ to denote their coproduct. If $I = \emptyset$ this is the empty presheaf $\emptyset$, with empty set as value at each path object.

5.3 A decomposition result

We will now observe that every presheaf $X \in \widehat{\mathbf{P}}$ decomposes into a sum of disjoint components rooted at one of the minimal path objects $\tau., a!v., a?$ where $a \in C, v \in V$. The notion of *rooted component* will play a key role. Let M be a minimal object in $\mathbf{P}$, Let $X \in \widehat{\mathbf{P}}$. Any $m \in X(M)$ determines a sub-presheaf C_m of X as follows. Letting $m \in X(M)$, define

$$C_m(P) \quad = \quad \begin{cases} \{p \in X(P) \mid X(M, P)(p) = m\} & \text{if } M \leq P, \\ \emptyset & \text{otherwise} \end{cases}$$

for $P \in \mathbf{P}$, and when $P \leq Q$ define the function $C_m(P, Q) : C_m(Q) \to C_m(P)$ by

$$C_m(P, Q)(q) = X(P, Q)(q) \text{ for } q \in C_m(Q)$$

— because X is a contravariant functor it follows that

$$X(M, P)(X(P, Q)(q) = X(M, Q)(q) = m$$

so that $X(P, Q)(q) \in C_m(P)$. It is easily checked that C_m is a presheaf and indeed a sub-presheaf of X because its action on morphisms (P, Q), when $P \leq Q$, restricts that of X.

Notation: In this situation, we shall say C_m is a *rooted component* of X at m.

Rooted components of X are pairwise disjoint in the sense that if M, M' are minimal objects of $\mathbf{P}$ and C_m is a rooted component at $m \in X(M)$ and

$C_{m'}$ is a rooted component at $m' \in X(M')$, then if at $P \in \mathbf{P}$, $C_m(P) \cap C_{m'}(P) \neq \emptyset$ then $M = M'$ and $m = m'$. Thus, for any path object $P \in \mathbf{P}$,

$$X(P) = \bigcup_{M} \bigcup_{m \in X(M)} C_m(P) \ , \tag{1}$$

a disjoint union, where M ranges over minimal objects of $\mathbf{P}$. Consequently, X is isomorphic to a sum of its rooted components:

$$X \cong \sum_{M} \sum_{m \in X(M)} C_m \tag{2}$$

where M ranges over minimal objects of $\mathbf{P}$ and C_m is the rooted component of X at m.

We analyse further the form of rooted components of $X \in \widehat{\mathbf{P}}$.

A rooted component C_i at $i \in X(\tau.)$ is isomorphic to $\tau.X_i$ where $X_i \in \widehat{\mathbf{P}}$ is given by

$$\begin{aligned} X_i(P) &= C_i(\tau.P), \text{ on objects } P \in \mathbf{P}, \text{ and} \\ X_i(P,Q) &= C_i(\tau.P, \tau.Q) : X_i(Q) \to X_i(P), \text{ on morphisms } P \leq Q \text{ of } \mathbf{P}. \end{aligned}$$

We write $X \xrightarrow{\tau} X'$ when there is $i \in X(\tau.)$ such that $X' = X_i$. The assignment $i \mapsto X_i$ is a bijection between the sets $X(\tau.)$ and $\{X' \mid X \xrightarrow{\tau} X'\}$.

A rooted component C_i at $i \in X(a!v)$, for $a \in C$ and $v \in V$, is isomorphic to $a!v.X_j$, where $X_j \in \widehat{\mathbf{P}}$ is given by

$$\begin{aligned} X_j(P) &= C_j(a!v.P), \text{ on objects } P \in \mathbf{P}, \text{ and} \\ X_j(P,Q) &= C_j(a!v.P, a!v.Q), \text{ on morphisms } P \leq Q \text{ of } \mathbf{P}. \end{aligned}$$

We write $X \xrightarrow{a!v} X'$ when there is $j \in X(a!v)$ such that $X' = X_j$. The assignment $j \mapsto X_j$ is a bijection between the sets $X(a!v.)$ and $\{X' \mid X \xrightarrow{a!v} X'\}$.

Let C_k be a rooted component at $K \in X(a?)$. Define

$$\begin{aligned} X_k(v)(P) &= C_k(a?(v \mapsto P)), \text{ and} \\ X_k(v)(P,Q) &= C_k(a?(v \mapsto P), a?(v \mapsto Q)) : X_k(v)(Q) \to X_k(v)(P). \end{aligned}$$

Then X_k is a function from values $v \in V$ to presheaves $X_k(v) \in \widehat{\mathbf{P}}$ such that C_k is isomorphic to $a?X_k$. We write $X \xrightarrow{a?} F$ when there is $k \in X(a?)$ such that F is isomorphic to X_k. The assignment $k \mapsto X_k$ is a bijection between the sets $X(a?)$ and $\{F \mid X \xrightarrow{a?} F\}$.

Recalling (1) above and the definition of X_j for $j \in X(a!v)$ and X_k for $k \in X(a?)$ we deduce:

$$X(\tau.P) = \bigcup_{i \in X(\tau.)} X_i(P)$$
$$X(a!v.P) = \bigcup_{j \in X(a!v.)} X_j(P)$$
$$X(a?(v \mapsto P)) = \bigcup_{k \in X(a?)} X_k(v)(P)$$

with unions which are disjoint, where $a \in C$ and $v \in V$.

Recalling the decomposition (2) above, we obtain the following decomposition result:

Proposition 1 *Let* $X \in \widehat{\mathbf{P}}$. *Then*

$$X \cong \sum_{i \in X(\tau.)} \tau.X_i + \sum_{(a,v) \in C \times V} \sum_{j \in X(a!v.)} a!v.X_j + \sum_{a \in C} \sum_{k \in X(a?.)} a?X_k \ .$$

5.4 Guarded recursive definitions

Presheaf categories possess all colimits and so in particular ω-colimits for building denotations of recursive definitions. In fact, because all our definitions have been given concretely as operations on sets, we are able to show that they are all continuous with respect to the sub-presheaf relation, and the solution of recursive definitions reduces to finding fixed points of a continuous function on cpo's; we obtain solutions up to equality and not just isomorphism.

There is clearly a well-founded relation $\prec$ on path objects $\mathbf{P}$ given by their inductive definition. If a presheaf X say is a solution to a guarded recursive definition then X will be equal to an expression in which each occurrence of X lies under a prefix operation. Hence by the results of Section 5.3, $X(P)$ is given in terms of $X(Q)$ where $Q \prec P$. Thus, by well-founded induction any solution is uniquely determined. A similar argument applies to an operation on presheaves, like parallel composition defined below, whose values on presheaves is defined recursively in terms of the operation under prefixes—it too is uniquely determined.

5.5 Parallel composition

Let $X, Y \in \widehat{\mathbf{P}}$ have the decompositions :

$$X \cong \sum_{i\in I} \tau.X_i + \sum_{(a,v)\in C\times V} \sum_{j\in J_{a,v}} a!v.X_j + \sum_{a\in C} \sum_{k\in K_a} a?X_k$$

$$Y \cong \sum_{l\in L} \tau.Y_l + \sum_{(a,v)\in C\times V} \sum_{m\in M_{a,v}} a!v.Y_m + \sum_{a\in C} \sum_{n\in N_a} a?Y_n$$

Their parallel composition $X \mid Y$ is defined recursively to be

$$\sum_{i\in I} \tau.(X_i \mid Y) + \sum_{(a,v)\in C\times V} \sum_{j\in J_{a,v}} a!v.(X_j \mid Y) + \sum_{a\in C} \sum_{k\in K_a} a?v.(X_k(v) \mid Y)$$

$$+ \sum_{l\in L} \tau.(X \mid Y_l) + \sum_{(a,v)\in C\times V} \sum_{m\in M_{a,v}} a!v.(X \mid Y_m) + \sum_{a\in C} \sum_{n\in N_a} a?v.(X \mid Y_n(v))$$

$$+ \sum_{(a,v)\in C\times V} \sum_{j\in J_{a,v}} \sum_{n\in N_a} \tau.(X_j \mid Y_n(v)) + \sum_{(a,v)\in C\times V} \sum_{m\in M_{a,v}} \sum_{k\in K_a} \tau.(X_k(v) \mid Y_m) \ .$$

5.6 Late denotational semantics

Suppose t is a process term with free process-variables within $U_1, \cdots, U_m$ and free value-variables within $x_1, \cdots, x_n$ (possibly empty lists). The denotation of t in this context, written $[\![t[U_1, \cdots, U_m; x_1, \cdots, x_n]]\!]$, is a function (extendable to a functor) $\widehat{\mathbf{P}}^m \times V^n \to \widehat{\mathbf{P}}$, given by structural induction on t in the usual fashion, matching syntactic constructs with the appropriate semantic operations:

$$[\![nil[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} = \emptyset, \quad \text{the empty presheaf.}$$

$$[\![\tau.t[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} = \tau.([\![t[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v})$$

$$[\![a!e.t[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} = a!w.([\![t[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v})$$

where e evaluates to w in environment $\vec{v}\ /\ \vec{x}$.

$$[\![a?y.t[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} = a?w.([\![t[\vec{U};\vec{x},y]]\!]\ \vec{X}\vec{v}\ w)$$

$$[\![t_1 \mid t_2[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} = [\![t_1[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} \mid [\![t_2[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v}$$

$$[\![t_1 + t_2[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} = [\![t_1[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} + [\![t_2[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v}$$

$$[\![[e_1 = e_2]t[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v}$$

$$= \begin{cases} [\![t[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} & \text{if } e_1, e_2 \text{ evaluate to a common value in } \vec{v}\ /\ \vec{x}. \\ \emptyset, & \text{the empty presheaf, otherwise.} \end{cases}$$

$$[\![U_i[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} = X_i$$

$$[\![recY.t[\vec{U};\vec{x}]]\!]\ \vec{X}\vec{v} = R, \quad \text{the unique solution of } R = [\![t[\vec{U},Y;\vec{x}]]\!]\ \vec{X}\ R\ \vec{v}\ .$$

Lemma 2 *Let t be a process term with free process-variables among $U_1, \cdots, U_m$ and free value-variables among $x_1, \cdots, x_n$. Suppose $s_1, \cdots, s_m$ are closed process-terms and that $v_1, \cdots, v_n$ are values in V. Then,*

$$[\![t[\vec{U};\vec{x}]]\!][\![\vec{s}]\!] = [\![t[\vec{s}\ /\ \vec{U}][\vec{v}\ /\ \vec{x}]]\!].$$

6 The late semantics related

The decomposition result and the preparatory discussion suggest that we view a presheaf over $\mathbf{P}$ as a transition system. In particular, it is sensible to view a relation $X \xrightarrow{\tau} X'$ holding between presheaves X, X' as meaning that the process represented by the presheaf X can make a τ-transition to a process represented by the presheaf X'. There is a similar reading of $X \xrightarrow{a!v} X'$, while $X \xrightarrow{a?} F$ means X can receive a value on channel a when, depending on the value v received, it will resume as process $F(v)$.

Thus, a closed process term is associated with two transition systems, one from the transition semantics and one from its denotation as a presheaf.

The next lemma asserts, essentially, that the relation

$$\{([\![t]\!], t) \mid t \text{ a closed process term}\}$$

is a late-bisimulation between the two transition systems.

Lemma 3 *Let t be a closed process term. Then,*

$$[\![t]\!] \xrightarrow{\tau} X \textit{ iff } \exists t'.\ t \xrightarrow{\tau} t' \ \&\ [\![t']\!] = X\ ,$$
$$[\![t]\!] \xrightarrow{a!v} X \textit{ iff } \exists t'.\ t \xrightarrow{a!v} t' \ \&\ [\![t']\!] = X\ ,$$
$$[\![t]\!] \xrightarrow{a?} F \textit{ iff } \exists t', y.\ t \xrightarrow{ay} t' \ \&\ [\![t'[y]]\!] = F\ .$$

Proof: For $W \in \widehat{\mathbf{P}}$ and t a closed process term define $W \approx t$ iff

$$\forall Z.\ W \xrightarrow{\tau} Z \Leftrightarrow \exists t'.\ t \xrightarrow{\tau} t' \ \&\ [\![t']\!] = Z,$$
$$\forall Z, a, v.\ W \xrightarrow{a!v} Z \Leftrightarrow \exists t'.\ t \xrightarrow{a!v} t' \ \&\ [\![t']\!] = Z, \text{ and}$$
$$\forall F, a.\ W \xrightarrow{a?} F \Leftrightarrow \exists t', y.\ t \xrightarrow{a?y} t' \ \&\ [\![t'[y]]\!] = F.$$

The proof proceeds by structural induction an process terms t with induction hypothesis:

> If t has free process-variables within $X_1, \cdots, X_n$, free value-variables within $x_1, \cdots, x_n$, and $S_1, \cdots, S_n$ are closed process-terms such that
>
> $$X_i \text{ is guarded in } t \text{ or } [\![S_i]\!] \approx S_i, \text{ whenever } 1 \leq i \leq m,$$
>
> then for all $v_1, \cdots, v_n \in V$,
>
> $$[\![t[\vec{X}; \vec{x}]]\!]\ [\![\vec{s}]\!]\vec{v} \approx t[\vec{s}\ /\ \vec{X}; \vec{v}\ /\ \vec{x}]$$
>
> —using an obvious vector notation.

Clearly, when t is closed the induction hypothesis amounts to $[\![t]\!] \approx t$, as required. □

As will be seen, the bisimilarity induced by spans of open maps in $\widehat{\mathbf{P}}$ coincides with the natural translation of late bisimulation to presheaves.

Definition: A *late bisimulation on presheaves* consists of a binary relation R on presheaves $\widehat{\mathbf{P}}$ such that whenever $X \; R \; Y$,

$$X \xrightarrow{\tau} X' \Rightarrow \exists Y'.\, Y \xrightarrow{\tau} Y' \;\&\; X' \; R \; Y',$$
$$Y \xrightarrow{\tau} Y' \Rightarrow \exists X'.\, X \xrightarrow{\tau} X' \;\&\; X' \; R \; Y',$$

$$X \xrightarrow{a!v} X' \Rightarrow \exists Y'.\, Y \xrightarrow{a!v} Y' \;\&\; X' \; R \; Y',$$
$$Y \xrightarrow{a!v} Y' \Rightarrow \exists X'.\, X \xrightarrow{a!v} Y \;\&\; X' \; R \; Y',$$

$$X \xrightarrow{a?} F \Rightarrow \exists G.\, Y \xrightarrow{a?} G \;\&\; \forall v \in V.\; F(v) \; R \; G(v),$$
$$Y \xrightarrow{a?} G \Rightarrow \exists F.\, X \xrightarrow{a?} F \;\&\; \forall v \in V.\; F(v) \; R \; G(v).$$

Say $X, Y \in \widehat{\mathbf{P}}$ are *late bisimilar* iff $X \; R \; Y$ for some late bisimulation on presheaves R.

That surjective open maps induce late bisimulations on presheaves follows directly from the next lemma.

Lemma 4 *Assume $f : X \to Y$ is an open map $\widehat{\mathbf{P}}$.*

Let M be a minimal object of $\widehat{\mathbf{P}}$. If C_m is a rooted component of X at $m \in X(M)$ then the image fC_m is a rooted component of Y at $f_M(m)$; the restriction f_{C_m} of f to C_m is an open map $f_{C_m} : C_m \to fC_m$.

Moreover, if f is surjective then any rooted component of Y is the image of a rooted component of X under f, and each restriction f_{C_m}, where C_m is a rooted component of X, is a surjective open map.

Proof: Direct consequence of the definition of open map. □

Corollary 5 *If $h : X \to Y$ is a surjective open map in $\widehat{\mathbf{P}}$, then X, Y are late bisimilar.*

Proof: Define R a relation on presheaves by:

$$W \; R \; Z \text{ iff } \exists f : W \to Z \text{ surjective and open in } \widehat{\mathbf{P}}.$$

Then R is a late bisimulation on presheaves by Lemma 4. □

Corollary 6 *If X, Y are bisimilar in $\widehat{\mathbf{P}}$, i.e. they are related by an span of surjective open maps, then X, Y are late bisimilar as presheaves.*

Proof: ¿From Corollary 5, as late bisimilarity on presheaves is easily seen to be an equivalence relation. □

Thus a span of surjective open maps yields a late bisimulation between presheaves. We now show the converse. For the presheaves X, Y and a late-bisimulation R which relates them we construct a sub-presheaf of $R_{XY} \subseteq X \times Y$ whose projections to X and Y are surjective open maps.

For $X \in \widehat{\mathbf{P}}$, recall from Section 5.3, the bijections between

- $i \in X(\tau.)$ and transitions $X \xrightarrow{\tau} X_i$,
- $j \in X(a!v)$ and transitions $X \xrightarrow{a!v} X_j$,
- $k \in X(a?)$ and transitions $X \xrightarrow{a?} X_k$.

They are used in the next definition.

Definition: Let R be a late bisimulation. Define, by induction on the structure of path objects $P \in \mathbf{P}$, sets $R_{XY}(P)$ whenever $X\ R\ Y$:

$$\begin{array}{lcl}
R_{XY}(\tau.) & = & \{(i,l) \in X(\tau.) \times Y(\tau.) \mid X_i\ R\ Y_l\} \\
R_{XY}(\tau.P) & = & \bigcup\{R_{X_i Y_l}(P) \mid (i,l) \in R_{XY}(\tau.)\} \\
 & & \\
R_{XY}(a!v.) & = & \{(j,m) \in X(a!v.) \times Y(a!v.) \mid X_j\ R\ Y_m\} \\
R_{XY}(a!v.P) & = & \bigcup\{R_{X_j Y_m}(P) \mid (j,m) \in R_{XY}(a!v.)\} \\
 & & \\
R_{XY}(a?) & = & \{(k,n) \in X(a?) \times Y(a?) \mid \forall v \in V.\ X_k(v)\ R\ Y_n(v)\} \\
R_{XY}(a?(v \mapsto P)) & = & \bigcup\{R_{X_{k(v)} Y_{n(v)}}(P) \mid (k,n) \in R_{XY}(a?)\}
\end{array}$$

Lemma 7 *Let R be a late bisimulation on presheaves. If $X\ R\ Y$, then*

(i) R_{XY} extends to a sub-presheaf of $X \times Y$.

(ii) The compositions $R_{XY} \hookrightarrow X \times Y \xrightarrow{\pi_1} X$ and $R_{XY} \hookrightarrow X \times Y \xrightarrow{\pi_2} Y$ are surjective open, where π_1, π_2 are the projections associated with the product $X \times Y$.

Proof: (i) It is first necessary to show that $R_{XY}(P) \subseteq X(P) \times Y(P)$. This follows by induction on the structure of $P \in \widehat{\mathbf{P}}$. For instance consider a path object of the form $a?(v \mapsto P)$. Suppose $X \xrightarrow{a?} X_k, k \in X(a?)$, and $Y \xrightarrow{a?} Y_n, n \in Y(a?)$, with $\forall v \in V.\ X_k(v)\ R\ Y_n(v)$. Now,

$$\begin{aligned} R_{X_k(v)Y_n(v)}(P) &\subseteq X_k(v)(P) \times Y_n(v)(P) \quad \text{by induction,} \\ &\subseteq X(a?(v \mapsto P)) \times Y(a?(v \mapsto P)) \quad \text{by Section 5.3.} \end{aligned}$$

Thus

$$R_{XY}(a?(v \mapsto P)) \subseteq X(a?(v \mapsto P)) \times Y(a?(v \mapsto P)).$$

An induction on the clauses for deriving morphisms $P \leq Q$ in $\widehat{\mathbf{P}}$ (see Section 4) shows $X(P,Q) \times Y(P,Q)$ restricts to a function $R_{XY}(Q) \to R_{XY}(P)$, making R_{XY} a sub-presheaf of $X \times Y$.

(ii) Write ρ_1, ρ_2 for the restriction of the projections $R_{XY} \hookrightarrow X \times Y \xrightarrow{\pi_1} X$ and $R_{XY} \hookrightarrow X \times Y \xrightarrow{\pi_2} Y$. That each component ρ_{1P}, ρ_{2P} is surjective is proved by induction on the structure of path objects P. The quasi-pullback conditions providing the openness of ρ_1 and ρ_2 are shown to hold by induction on the clauses for deriving morphisms $P \leq Q$ in $\widehat{\mathbf{P}}$. □

Hence:

Theorem 8 *Presheaves $X, Y \in \widehat{\mathbf{P}}$ are late-bisimilar iff they are related by a span of surjective open maps.*

The next lemma links late-bisimilation on presheaves and late-bisimulation on closed terms of **VProc**, and yields the main result of this section—the equivalence of the operational and denotational formulations of bisimilarity.

Lemma 9 *Let t_1, t_2 be closed process terms. The denotations $[\![t_1]\!], [\![t_2]\!]$ are late bisimilar as presheaves iff t_1, t_2 are late bisimilar.*

Proof: Assuming R is a late-bisimulation on presheaves, it is claimed we obtain a late-bisimulation S on (closed) process terms by defining

$$S = \{(t_1, t_2) \ \mid\ [\![t_1]\!]\ R\ [\![t_2]\!]\}.$$

Conversely, assuming S is a late-bisimulation on (closed) process terms, it is claimed we obtain a late bisimulation on presheaves by defining

$$R = \{([\![t_1]\!], [\![t_2]\!]) \ \mid\ t_1\ S\ t_2\}.$$

The proof of these two claims rests on Lemma 3, with recourse to the Substitution Lemma 2.

For example, suppose S is obtained from a late-bisimulation on presheaves as above. Suppose $s_1 S s_2$ and $s_1 \xrightarrow{a?y} s_1'$. For S to be a late bisimulation we are required to find a matching transition of s_2. However, by Lemmas 3, $[\![s_1]\!] \xrightarrow{a?} [\![s_1'[y]]\!]$, so because $[\![s_1]\!] R [\![S_2]\!]$ there is $G : V \to \widehat{\mathbf{P}}$ for which

$$[\![s_2]\!] \xrightarrow{a?} G \text{ and } \forall v \in V.\ [\![s_1'[y]]\!](v) R G(v).$$

By Lemma 3 again, there are s_2', z for which

$$s_2 \xrightarrow{a?z} s_2' \ \&\ [\![s_2'[z]]\!] = G.$$

By the Substitution Lemma 2,

$$\forall v \in V.\ [\![s_1'[v/y]]\!] R [\![s_2'[v/z]]\!],$$

i.e. $\forall v \in V.\ s'[v/y] S s_2'[v/z]$, as required. □

Theorem 10 *Closed process terms t_1, t_2 of* **VProc** *are late-bisimilar iff their denotations $[\![t_2]\!], [\![t_2]\!]$ are related by a span of surjective open maps.*

Proof: Directly from Theorem 8 and Lemma 9. □

7 Variations

A transition semantics and bisimulation for **VProc** with early value passing can be obtained easily on the lines of [14]. An appropriate presheaf semantics is obtained with a path category a partial order which is an initial solution to:

$$\mathbf{P} = \mathbf{P}_\perp + \sum_{(a,v) \in C \times V} \mathbf{P}_\perp + \sum_{(a,v) \in C \times V} \mathbf{P}_\perp$$

In fact $\widehat{\mathbf{P}}$ is isomorphic to rooted presheaves over $\mathbf{P}_\perp$ which is readily seen to be isomorphic to a category of synchronisation trees in which labels have the form τ, $a!v$ or $a?v$ where $a \in C$ and $v \in V$, a category $\mathbf{ST}_{C \times V}$ in the notation of [12, 4]. For such categories bisimulation obtained from open maps has been shown to coincide with Park and Milner's strong bisimulation [12].

Furthermore, denotational semantics is given in [19] in which denotations of terms as synchronisation trees are strong bisimilar to the transition systems from an operational semantics. Thus there is no difficulty in producing a denotational semantics so that the denotation of closed terms in $\widehat{\mathbf{P}}$ are connected by a span of open surjections iff the terms are strong bisimilar.

A much greater challenge is provided by a process-passing language with a syntax similar to that of **VProc**

$$t ::= nil \mid \tau.t \mid a!t_1.t_2 \mid a?X.t \mid (t_1 \mid t_2) \mid t_1 + t_2 \mid X \mid recX.t$$

but where in contrast to **VProc** a process t_1 can be sent along a channel a by a process $a!t_1.t_2$ and an arbitrary process can be received on a and bound to process-variable X in a process $a?X.t$. A transition semantics can be found, for instance, in [16]. A path category for process-passing with late semantics is reasonably taken to be an initial solution to the following isomorphism in **Prof**

$$\mathbf{P} \cong \mathbf{P}_\perp + \sum_{a\in C}(\mathbf{P}\times\mathbf{P})_\perp + \sum_{a\in C}(!\mathbf{P} \multimap \mathbf{P})_\perp$$

or sufficiently an initial solution to

$$\mathbf{P} \cong \mathbf{P}_\perp + \sum_{a\in C}(\mathbf{P}+\mathbf{P})_\perp + \sum_{a\in C}((!\mathbf{P})^{op} \times \mathbf{P})_\perp$$

in **Cat**—the constructions one is led to by Section 3. There is little trouble in giving a denotational semantics to a term with n free variables as a functor $\widehat{\mathbf{P}}^n \to \widehat{\mathbf{P}}$. So closed terms, denoting presheaves in $\widehat{\mathbf{P}}$, inherit a notion of bisimulation from open maps in presheaf categories. But there is a problem in understanding the bisimulation that arises, for example as a coinductive definition based on a transition semantics, along the usual lines. The difficulties are due to the function space component $(!\mathbf{P} \multimap \mathbf{P})$.

On the other hand, there seem to be no fundamental difficulties in presenting a presheaf model of the Pi-calculus, where following the lead of [17, 5] we (Cattani,Stark,Winskel) move to $\mathbf{Prof}^{\mathbf{I}}$, indexed by a category of name-sets $\mathbf{I}$.

Acknowledgements

I especially thank Martin Hyland for very helpful discussions during my stay at the Isaac Newton Institute, Cambridge University. Thanks also to Jaap

van Oosten for useful pointers to the literature.

References

[1] Abramsky, S., On semantic foundations of applicative multiprogramming. Proc. ICALP'83, Barcelona, LNCS 154, 1983.

[2] Adamek, J., and Rosicky J., Locally presentable and accessible categories. LMS Lecture Notes Series 189, 1994.

[3] Borceux, F., Handbook of categorical logic, 1. Cambridge University Press, 1994.

[4] Cattani, G.L., and Winskel, G., Presheaf models for concurrency. Manuscript, 1996.

[5] Fiore, M., Moggi, E., and Sangiori, D., A fully abstract model for the Pi-calculus. Proc. of LICS'96.

[6] Hennessy, M., and Ingolfsdottir, A., A theory of communicating processes with value passing. Information and Computation, 107(2), 1993.

[7] Hennessy, M., and Plotkin, G., A term model for CCS. Proc. 9th MFCS, Poland, LNCS 88, 1980.

[8] Ingolfsdottir, A., A semantic theory for value-passing processes, late appraoch—Parts I and II. BRICS reports RS–95–3/22, 1995.

[9] Johnstone, P.T., Stone Spaces. Cambridge University Press, 1982.

[10] Joyal, A., and Moerdijk, I., Algebraic set theory. LMS Lecture Notes Series 220, Cambridge University Press, 1995.

[11] Joyal, A., and Moerdijk, I., A completeness theorem for open maps. In Annals of Pure and Applied Logic 70, 51-86, 1994.

[12] Joyal, A., Nielsen, M., and Winskel, G., Bisimulation from open maps. Report Series RS-94-7, BRICS, University of Aarhus, Denmark, May 1994. Accepted for a LICS 93 special issue of Information and Computation.

[13] Lehman, D., Categories for fixed point semantics. FOCS 17, 1976.

[14] Milner, A.R.G., Communication and concurrency, Prentice Hall, 1989.

[15] Milner, R., Parrow, J., and Walker, D., A calculus of mobile processes, Parts I and II. Information and Computation, 100:1-77, 1992.

[16] Sangiori, D., Bisimulation for higher-order process calculi. INRIA Report, Sophia-Antipolis, RR-2508, 1995.

[17] Stark, I.,A fully abstract domain model for the Pi-calculus. Proc. of LICS'96, 1996.

[18] Vickers, S., Geometric theories and databases. In LMS Lecture Notes Series 177, 1992.

[19] Winskel, G., and Nielsen, M., Models for concurrency. In the Handbook of Logic in Computer Science, vol.IV, ed. Abramsky, Gabbay and Maibaum, Oxford University Press, 1995.

Recent Publications in the BRICS Report Series

RS-96-44 **Glynn Winskel.** ***A Presheaf Semantics of Value-Passing Processes*****. November 1996. 23 pp. Extended and revised version of paper appearing in Montanari and Sassone, editors,** ***Concurrency Theory: 7th International Conference*****, CONCUR '96 Proceedings, LNCS 1119, 1996, pages 98–114.**

RS-96-43 **Anna Ingólfsdóttir.** ***Weak Semantics Based on Lighted Button Pressing Experiments, An Alternative Characterization of the Readiness Semantics*****. November 1996. An extended abstract to appear in the proceedings of the** ***10th Annual International Conference of the European Association for Computer Science Logic*****, CSL '96.**

RS-96-42 **Gerth Stølting Brodal and Sven Skyum.** ***The Complexity of Computing the k-ary Composition of a Binary Associative Operator*****. November 1996. 15 pp.**

RS-96-41 **Stefan Dziembowski.** ***The Fixpoint Bounded-Variable Queries are PSPACE-Complete*****. November 1996. 16 pp. Presented at the** ***10th Annual International Conference of the European Association for Computer Science Logic*****, CSL '96.**

RS-96-40 **Gerth Stølting Brodal, Shiva Chaudhuri, and Jaikumar Radhakrishnan.** ***The Randomized Complexity of Maintaining the Minimum*****. November 1996. 20 pp. To appear in a special issue of** ***Nordic Journal of Computing*** **devoted to the proceedings of SWAT '96. Appears in Karlson and Lingas, editors,** ***Algorithm Theory: 5th Scandinavian Workshop*****, SWAT '96 Proceedings, LNCS 1097, 1996, pages 4–15.**

RS-96-39 **Hans Hüttel and Sandeep Shukla.** ***On the Complexity of Deciding Behavioural Equivalences and Preorders – A Survey*****. October 1996. 36 pp.**

RS-96-38 **Hans Hüttel and Josva Kleist.** ***Objects as Mobile Processes*****. October 1996. 23 pp.**