Semantics of Name and Value Passing

Marcelo Fiore*
Computer Laboratory
University of Cambridge

Daniele Turi†
LFCS, Division of Informatics
University of Edinburgh

Abstract

We provide a semantic framework for (first order) message-passing process calculi by combining categorical theories of abstract syntax with binding and operational semantics. In particular, we obtain abstract rule formats for name and value passing with both late and early interpretations. These formats induce an initial-algebra/final-coalgebra semantics that is compositional, respects substitution, and is fully abstract for late and early congruence. We exemplify the theory with the π-calculus and value-passing CCS.

Introduction

A complete description of the semantics of a programming language requires both an operational semantics describing the behaviour of programs in terms of elementary steps and a more abstract denotational semantics describing the meaning of a program in terms of its components [32]. In the study of process calculi for concurrency (such as CCS [25], CSP [19], and ACP [4]) less emphasis is placed on denotational models and more on notions of behavioural equivalence, and on bisimulation equivalence [25] in particular. Still, for the operational semantics to be well-behaved, one requires that the chosen notion of behavioural equivalence be a congruence with respect to the constructs of the language.

To establish congruence results for behavioural equivalences it is convenient to define the operational semantics in terms of structural rules, i.e., Plotkin's SOS rules [29]. Correspondingly, much work has been done in order to identify SOS rule formats [10, 6, 17, 14] for which (strong) bisimulation is a congruence – the most well-known being GSOS [6]. However, such formats are hard to find and even harder to extend. Little or no success at all has been gained, e.g., in obtaining formats for more sophisticated process calculi than the above mentioned ones – process calculi with variable binding (like value-passing CCS [26] and the π-calculus [27]) in particular. The present paper addresses this very problem.

The solution we offer is based on understanding the mathematical structure underlying syntax and semantics of message passing processes. The formats we obtain are abstract and require a fair amount of category theory. However, concrete, syntactic formats can be distilled from them and this, indeed, will be the next step of our investigation.

The starting point for our work lies in [35], where a categorical rule format is defined in terms of functorial notions Σ and B of syntax and behaviour familiar from initial algebra [16] and final coalgebra [1, 36] semantics. This format is given by transformations

$$\Sigma(X \times BX) \longrightarrow BTX \qquad (1)$$

natural in the parameter X (to be thought of as a generic set of meta-variables used in the rules), where T is the term monad associated to the signature Σ, i.e., $TX = \mu Y. X + \Sigma Y$.

The type in (1) arises from giving to each operator of arity n of the signature a natural transformation

$$(X \times BX)^n \longrightarrow BTX \qquad (2)$$

describing the overall behaviour of the operator in terms of the behaviour of its arguments. This abstract format corresponds to GSOS when B is taken to be the functor on **Set** whose coalgebras are finitely branching labelled transition systems, i.e.,

$$BX = \wp_{\mathrm{f}}(L \times X) \qquad (3)$$

where L is a finite set of labels and $\wp_{\mathrm{f}}$ is the finite powerset functor. In this case, the domain $(X \times \wp_{\mathrm{f}}(L \times X))^n$ and the codomain $\wp_{\mathrm{f}}(L \times TX)$ of the map in (2) correspond, respectively, to the premises and the conclusions of GSOS rules for the operator. Interestingly, naturality accounts exactly for the GSOS restrictions on the occurrences of variables in the rules.

Any natural transformation of type (1) has the property that the coalgebraic behavioural equivalence associated to B (which in the above case coincides with bisimulation [2])

*Research supported by an EPSRC Advanced Research Fellowship.
†Research supported by EPSRC grant R34723.

0-7695-1281-X/01 $10.00 © 2001 IEEE

is a congruence with respect to the operators of the syntax Σ. This is a corollary of the more general fact that rules in the format (1) induce a denotational semantics which is adequate in the sense that it is fully abstract with respect to behavioural equivalence.

The above result is independent from the choice of category and functors, provided they have enough structure and properties. Here we exploit this generality in order to find formats for process calculi with variable binding. To this end, we first had to give a functorial notion of syntax with binding. This was one of the main motivations for the work in [13], where we moved from sets to variable sets. There, variable sets are taken to be functors (called covariant presheaves) from a category of contexts to **Set**; the category of contexts used is the category $\mathbb{F}$ of finite cardinals (i.e., sets of variables) and all functions (i.e., renamings). Most importantly, there exist a distinguished presheaf V of variables and a differentiation functor $\delta = (\_)^V$ on presheaves. The latter is used to model variable binding with arity V: for a presheaf X, the elements of δX in context n are simply the elements of X in the context $n + 1$ containing an extra variable – the variable to be bound.

We have now to find the right notions of behaviour B for name and value passing. Let us start from name passing, where the two most natural notions of behavioural equivalence are late and early bisimulation [27]. These are not congruences for the π-calculus though; one then considers the late and early congruences instead [27], obtained by closing bisimulation under renamings (i.e., the maps of $\mathbb{F}$).

Previous (implicitly) coalgebraic work on name passing [12, 33] was based on a functor B whose associated behavioural equivalence turns out to be late bisimulation. This functor B lives in the category of presheaves over the category $\mathbb{I}$ of name contexts allowing only injective renamings. Surprisingly, the natural extension of such B to the category of presheaves over $\mathbb{F}$ yields a new behaviour $\widetilde{B}$ whose associated equivalence is exactly late congruence.

We are also able to solve the problem left open in [12, 33] of giving a denotational semantics fully abstract with respect to early bisimulation by introducing a new behaviour whose associated equivalence is early bisimulation[1]. The extension of such behaviour to the presheaves over $\mathbb{F}$ has early congruence as associated equivalence. Therefore, the desired formats for early and late congruences live in the category of presheaves over $\mathbb{F}$ and, for instance, rules for unary binding will be of type

$$(X \times \widetilde{B}X)^V \longrightarrow \widetilde{B}TX \tag{4}$$

where $\widetilde{B}$ can be the extended behaviour for either late or early congruence.

[1] See also [28] for a different coalgebraic approach to early (and late) bisimulation and [8] for a domain equation for early bisimulation in the framework of presheaf models.

For value passing, we also give late and early behaviours, which are variations (cf. [20]) of the behaviour in (3). However, in order to model input rules we have to take into account the substitution structure present in value-passing calculi, i.e., the homogeneous substitution of messages in messages and the heterogeneous substitution of messages in processes. (For name passing this is not needed because substitution is just renaming, hence it is already, though implicitly, part of the category of presheaves over $\mathbb{F}$.)

The categorical framework for homogeneous substitution was developed in [13]. One considers a monoidal structure on presheaves '$\bullet$' with unit V. A presheaf $X \bullet Y$ can be thought of has having elements given by pairs of an element of X together with a substitution consisting of a tuple of elements of Y. One then takes the notion of homogeneous substitution on a presheaf M to be a monoid structure $V \longrightarrow M \longleftarrow M \bullet M$.

Here, in order to model the heterogeneous substitution of elements of a monoid M in elements of a presheaf X, we need to go one step further and consider monoid actions $X \bullet M \longrightarrow X$. Correspondingly, the modelling of rules takes place in the category of actions of the monoid of messages. Therefore, we need then to lift signatures with binding Σ and extend behaviours B to functors $\overline{\Sigma}$ and $\widetilde{B}$ on such category.

In general, we have primitive notions Σ and B living in different categories, of syntax $\mathcal{S}$ and behaviour $\mathcal{B}$ respectively, while the rules live in yet another category $\mathcal{A}$ of substitutions (e.g., monoid actions). These categories are related by adjunctions:

$$\begin{tikzcd} \mathcal{A} \arrow[loop above, "\widetilde{B}"] \arrow[loop below, "\overline{\Sigma}"'] \arrow[r, shift right=3] & \mathcal{S} \arrow[l, shift right=3, "\top"'] \arrow[l, shift left=3, "\top"] \arrow[loop below, "\Sigma"'] \arrow[r, shift right=2, "\top"'] & \mathcal{B} \arrow[l, shift right=2] \arrow[loop above, "B"] \end{tikzcd} \tag{5}$$

The lifting of the Σ on $\mathcal{S}$ to a $\overline{\Sigma}$ on $\mathcal{A}$ is done by means of a distributive law over the monad induced by the monadic adjunction $\mathcal{A} \underset{\longleftarrow}{\overset{\longrightarrow}{\top}} \mathcal{S}$, while the behaviour $\widetilde{B}$ on $\mathcal{A}$ is obtained by (right) extending B on $\mathcal{B}$ along the composite adjunction $\mathcal{A} \underset{\longrightarrow}{\overset{\longleftarrow}{\top}} \mathcal{B}$. These constructions yield liftings/extensions as follows:

$$\begin{tikzcd} \overline{\Sigma}\text{-Alg} \arrow[r] \arrow[d] & \Sigma\text{-Alg} \arrow[d] \\ \mathcal{A} \arrow[r] & \mathcal{S} \end{tikzcd} \qquad \begin{tikzcd} \widetilde{B}\text{-Coalg} \arrow[r] \arrow[d] & B\text{-Coalg} \arrow[d] \\ \mathcal{A} \arrow[r] & \mathcal{B} \end{tikzcd}$$

The abstract rule format ensuring that behavioural equivalence is a congruence consists then of natural transformations of type

$$\overline{\Sigma}(X \times \widetilde{B}X) \longrightarrow \widetilde{B}\overline{T}X \tag{6}$$

For name passing the actions of the monoid of variables are simply presheaves on $\mathbb{F}$, hence $\overline{\Sigma}$ is equal to Σ. For the original GSOS case of [35], with no variable binding, all three categories collapse to the category of sets, hence $\overline{\Sigma}$ and $\widetilde{B}$ are equal to Σ and B respectively and we recover (1).

The next obvious step for our work is to characterise the categorical rule formats for name and value passing proposed in this paper in elementary syntactic terms. The rule formats so obtained will certainly not be as in [5], where binding and substitution are defined within the rules rather than treated at the syntactic level. For value passing, our categorical rule format seems to be related to a syntactic format proposed in [30]. The relationship with the format of [15] for which a conservative extension property holds should also be investigated.

Another aspect we would like to consider is recursion. At present we would deal with guarded recursion following [34], but it would be interesting to deal with unguarded recursion along the lines of [31], hence working with variable cpos instead of variable sets.

Finally, there seems to be a tight correspondence between the coalgebras of our new behaviour for early bisimulation and the indexed labelled transition systems of [7]. We would like to investigate this for sheaves (in the Schanuel topos) rather than presheaves over $\mathbb{I}$.

1. Basic syntactic and semantic structures

1.1. Expressions

Syntax. Consider the following abstract grammar of expressions for integers

$$e ::= x \mid \underline{z} \mid e_1 \,\mathtt{plus}\, e_2 \mid e_1 \,\mathtt{minus}\, e_2 \tag{7}$$

where x ranges over a countable list of variables x_i ($i \in \mathbb{N}$) and $\underline{z}$ over the set of integers $\mathbb{Z}$.

Following [13], we consider terms in a context, so that we can stratify expressions into a family $\{ E_n \}_{n\in\mathbb{N}}$ of sets indexed by natural numbers (indicating the number of variables in the context). The set E_n consists of the expressions with at most n (*canonical*) free variables (typically denoted by $x_1, \ldots, x_n$). Thus, $\{ E_n \}_{n\in\mathbb{N}}$ is the least solution of the equations

$$\{ X_n = \{ x_1, \ldots, x_n \} + \mathbb{Z} + {X_n}^2 + {X_n}^2 \}_{n\in\mathbb{N}} \tag{8}$$

Semantics. We write $\mathcal{E}[\![e]\!]_n$ for the interpretation of an expression e in the context $x_1, \ldots, x_n$; that is, for the function $\mathbb{Z}^n \longrightarrow \mathbb{Z}$ defined compositionally as follows:

$$\begin{aligned}
&1.\ \mathcal{E}[\![x_i]\!]_n = \pi_i \quad (i^{\text{th}} \text{ projection}, 1 \leq i \leq n)\\
&2.\ \mathcal{E}[\![\underline{z}]\!]_n = \lambda\vec{x}.z \quad (\text{constant function z})\\
&3.\ \mathcal{E}[\![e_1 \,\mathtt{plus}\, e_2]\!]_n = \lambda\vec{x}.(\mathcal{E}[\![e_1]\!]_n(\vec{x}) + \mathcal{E}[\![e_2]\!]_n(\vec{x}))\\
&4.\ \mathcal{E}[\![e_1 \,\mathtt{minus}\, e_2]\!]_n = \lambda\vec{x}.(\mathcal{E}[\![e_1]\!]_n(\vec{x}) - \mathcal{E}[\![e_2]\!]_n(\vec{x}))
\end{aligned} \tag{9}$$

This interpretation is an initial algebra semantics. Indeed, the semantic domain given by

$$\{ \mathbf{Set}(\mathbb{Z}^n, \mathbb{Z}) \}_{n\in\mathbb{N}} \tag{10}$$

where $\mathbf{Set}(S, S')$ denotes the set of functions from a set S to a set S', has a (pointwise) algebra structure given by the evident maps

$$\begin{aligned}
&\{ x_1, \ldots, x_n \} \longrightarrow \mathbf{Set}(\mathbb{Z}^n, \mathbb{Z})\\
&\mathbb{Z} \longrightarrow \mathbf{Set}(\mathbb{Z}^n, \mathbb{Z})\\
&\mathbf{Set}(\mathbb{Z}^n, \mathbb{Z})^2 \cong \mathbf{Set}(\mathbb{Z}^n, \mathbb{Z}^2) \longrightarrow \mathbf{Set}(\mathbb{Z}^n, \mathbb{Z})\\
&\mathbf{Set}(\mathbb{Z}^n, \mathbb{Z})^2 \cong \mathbf{Set}(\mathbb{Z}^n, \mathbb{Z}^2) \longrightarrow \mathbf{Set}(\mathbb{Z}^n, \mathbb{Z})
\end{aligned} \tag{11}$$

and

$$\mathcal{E} = \{ \mathcal{E}[\![\_]\!]_n : E_n \longrightarrow \mathbf{Set}(\mathbb{Z}^n, \mathbb{Z}) \}_{n\in\mathbb{N}} \tag{12}$$

is the unique algebra homomorphism from $\{ E_n \}_{n\in\mathbb{N}}$ to $\{ \mathbf{Set}(\mathbb{Z}^n, \mathbb{Z}) \}_{n\in\mathbb{N}}$.

1.2. Presheaves

Categorically, families $\{ X_n \}_{n\in\mathbb{N}}$ of sets are functors

$$X : \mathbb{N} \longrightarrow \mathbf{Set}$$

where $\mathbb{N}$ is the discrete category of natural numbers or, equivalently, finite cardinals. Since we regard a finite cardinal n as a context of n variables, a function $\rho : n \longrightarrow m$ can be seen as a renaming of variables. In order to model weakening, contraction, and exchange rules for contexts we need to use, instead of the discrete category $\mathbb{N}$, the category $\mathbb{F}$ of finite cardinals and all functions (cf. [13]). Correspondingly, we consider functors

$$X : \mathbb{F} \longrightarrow \mathbf{Set}$$

i.e., (covariant) *presheaves* over $\mathbb{F}$. Thus, we will be working with families $\{ X_n \}_{n\in\mathbb{N}}$ of sets equipped with an *action* that associates every $x \in X_n$ (i.e., an element of X at stage n) and every renaming $\rho : n \longrightarrow m$ with

$$x[\rho] = X(\rho)(x) \in X_m$$

Presheaves over $\mathbb{F}$ form a category $\mathbf{Set}^{\mathbb{F}}$, with natural transformations as morphisms.

Syntax. The family $\{ E_n \}_{n\in\mathbb{N}}$ with action

$$e[\rho] = e[^{x_{\rho 1}}/_{x_1}, \ldots, ^{x_{\rho n}}/_{x_n}] \qquad (\rho : n \longrightarrow m)$$

given by variable renaming defines a presheaf $E : \mathbb{F} \to \mathbf{Set}$. This presheaf is the least solution of the equation

$$X = V + \mathcal{K}_{\mathbb{Z}} + X^2 + X^2$$

in $\mathbf{Set}^{\mathbb{F}}$ (cf. (8)), where the presheaf of *variables*

$$V : \mathbb{F} \longrightarrow \mathbf{Set}\ , \qquad V_n = n \cong \{ x_1, \ldots, x_n \}$$

is the inclusion of $\mathbb{F}$ into $\mathbf{Set}$ and $\mathcal{K}_{\mathbb{Z}}$ is the constantly $\mathbb{Z}$ presheaf. Hence E is the free Σ-algebra $\mu Y.\, V + \Sigma Y$ over the presheaf of variables V, where

$$\Sigma : \mathbf{Set}^{\mathbb{F}} \longrightarrow \mathbf{Set}^{\mathbb{F}}\ , \quad \Sigma X = \mathbb{Z} + X^2 + X^2$$

is the endofunctor on presheaves associated to the operators on expressions.

Semantics. Also the semantic domain for expressions (10) has a presheaf structure. Indeed, for any object C of a cartesian category $\mathcal{C}$, we have a functor

$$\langle C, \_\rangle : \mathcal{C} \longrightarrow \mathbf{Set}^{\mathbb{F}}\ , \qquad \langle C, D\rangle_n = \mathcal{C}(C^n, D) \qquad (13)$$

The presheaf $\langle C, D\rangle$ can be thought of as the presheaf of mappings from environments of type C to results of type D. Formally, at stage n, it consists of the set of morphisms in $\mathcal{C}$ from C^n to D with action

$$f[\rho] = f \circ \langle \pi_{\rho 1}, \ldots, \pi_{\rho n}\rangle \qquad (\rho : n \longrightarrow m)$$

In particular, taking $\mathcal{C} = \mathbf{Set}$ and $C = D = \mathbb{Z}$ we obtain the presheaf $\langle \mathbb{Z}, \mathbb{Z}\rangle$ with underlying family of sets as in (10).

The copairing of the maps in (11) gives a Σ-algebra structure

$$\Sigma\langle \mathbb{Z}, \mathbb{Z}\rangle = \mathcal{K}_{\mathbb{Z}} + \langle \mathbb{Z}, \mathbb{Z}\rangle^2 + \langle \mathbb{Z}, \mathbb{Z}\rangle^2 \longrightarrow \langle \mathbb{Z}, \mathbb{Z}\rangle \qquad (14)$$

on $\langle \mathbb{Z}, \mathbb{Z}\rangle$ that induces the initial algebra semantics

$$\mathcal{E} : E \longrightarrow \langle \mathbb{Z}, \mathbb{Z}\rangle$$

of (12). Note that the naturality of $\mathcal{E}$ amounts to the identity

$$\begin{aligned} &\mathcal{E}[\![\, e[^{x_{\rho 1}}/_{x_1}, \ldots, ^{x_{\rho n}}/_{x_n}] \,]\!]_m(z_1, \ldots, z_m) \\ &\quad = \mathcal{E}[\![e]\!]_n(z_{\rho 1}, \ldots, z_{\rho n}) \end{aligned} \qquad (15)$$

for all $\rho : n \longrightarrow m$.

Syntax with binding. In the algebraic treatment of binding of [13], binding operators are modelled using the *differentiation* operator

$$\delta : \mathbf{Set}^{\mathbb{F}} \longrightarrow \mathbf{Set}^{\mathbb{F}}\ , \qquad (\delta X)_n = X_{n+1}$$

(For details, including initial algebra semantics, consult [13].)

Pi-calculus. The following grammar for (a fragment of) the π-calculus

$$t ::= \mathbf{0} \ \mid\ t_1 | t_2 \ \mid\ x(y).t \ \mid\ \bar{x}y.t \ \mid\ (x)t \ \mid\ [x = y]\,t$$

corresponds to the signature endofunctor

$$\begin{aligned} \Sigma X \ &= \ 1 + X \times X + V \times \delta X + V \times V \times X \\ &\quad + \delta X + V \times V \times X \end{aligned} \qquad (16)$$

on $\mathbf{Set}^{\mathbb{F}}$. Indeed, its initial algebra

$$\begin{aligned} T0 \ &\cong \ 1 + T0 \times T0 + V \times \delta T0 + V \times V \times T0 \\ &\quad + \delta T0 + V \times V \times T0 \end{aligned}$$

is the presheaf of π-calculus terms: at stage n it is the set of (α-equivalence classes of) terms with at most n (canonical) free variables, with action given by variable renaming.

Value-passing CCS. We will consider the following fragment of CCS passing expressions e as in (7) along a finite set of channels $c \in C$:

$$t ::= \mathbf{0} \ \mid\ t_1 | t_2 \ \mid\ c?(x).t \ \mid\ c!\langle e\rangle.t \ \mid\ [e_1 = e_2]\,t$$

This grammar has associated signature endofunctor

$$\begin{aligned} \Sigma_E X \ &= \ 1 + X \times X + \mathcal{K}_C \times \delta X \\ &\quad + \mathcal{K}_C \times E \times X + E \times E \times X \end{aligned}$$

on $\mathbf{Set}^{\mathbb{F}}$, where $\mathcal{K}_C$ is the constantly C presheaf.

More generally, we have a signature bifunctor $\Sigma : \mathbf{Set}^{\mathbb{F}} \times \mathbf{Set}^{\mathbb{F}} \longrightarrow \mathbf{Set}^{\mathbb{F}}$

$$\begin{aligned} \Sigma(M, X) \ &= \ 1 + X \times X + \mathcal{K}_C \times \delta X \\ &\quad + \mathcal{K}_C \times M \times X + M \times M \times X \end{aligned} \qquad (17)$$

parametric in the presheaf of messages being passed.

1.3. Substitution

Clones. We have seen that besides the operators, the semantics $\mathcal{E}$ also respects variable renaming (see (9) and (15)). However, $\mathcal{E}$ respects *substitution* in the stronger form of satisfying the *semantic substitution lemma*:

$$\begin{aligned} &\mathcal{E}[\![e\,[^{e_1}/_{x_1}, \ldots, ^{e_n}/_{x_n}]]\!]_m \\ &\quad = \mathcal{E}[\![e]\!]_n \circ \langle \mathcal{E}[\![e_1]\!]_m, \ldots, \mathcal{E}[\![e_n]\!]_m\rangle \end{aligned} \qquad (18)$$

In other words, $\mathcal{E}$ is not only an algebra homomorphism but also, as we explain below, a clone homomorphism.

Recall that an *(abstract) clone* [9, page 132] X, consists of a family $\{ X_n \}_{n \in \mathbb{N}}$ of sets, a family

$$\{ \nu_i^{(n)} \in X_n \mid 1 \leq i \leq n \}_{n \in \mathbb{N}}$$

of distinguished elements, and a family

$$\{ \mu_m^{(n)} : X_n \times (X_m)^n \longrightarrow X_m \}_{n,m \in \mathbb{N}}$$

of operations such that, for every element t of X_n, every n-tuple $\vec{u} = (u_1, \ldots, u_n)$ of elements of X_m, and every m-tuple $\vec{v}$ of elements of X_ℓ, the following three axioms hold:

$$\begin{array}{l} \mu_m(\nu_i; \vec{u}) = u_i \qquad \mu_n(t; \nu_1, \ldots, \nu_n) = t \\ \mu_\ell(\mu_m(t; \vec{u}); \vec{v}) = \mu_\ell(t; \mu_\ell(u_1; \vec{v}), \ldots, \mu_\ell(u_n; \vec{v})) \end{array} \tag{19}$$

An homomorphism $h : X \longrightarrow X'$ between clones is a family $\{ h_n : X_n \longrightarrow X'_n \}_{n \in \mathbb{N}}$ of functions that respects the clone structure.

The clone structure on the family $\{ E_n \}_{n \in \mathbb{N}}$ of expressions is given by the variables x_i $(1 \leq i \leq n)$ in E_n and by the simultaneous substitution of expressions for expressions

$$\begin{array}{ccc} E_n \times (E_m)^n & \longrightarrow & E_m \\ (e; e_1, \ldots, e_n) & \longmapsto & e\,[{}^{e_1}/_{x_1}, \ldots, {}^{e_n}/_{x_n}] \end{array}$$

(The three axioms in (19) amount to the familiar properties of substitution.) For the semantic domain $\langle \mathbb{Z}, \mathbb{Z} \rangle$, the clone structure is given by projections and function composition (together with pairing). In fact, for every object C of a cartesian category $\mathcal{C}$, one can form the *clone of operations* $\langle C, C \rangle$ on C, with $\nu_i^{(n)}$ given by the i^{th} projection $\pi_i : C^n \longrightarrow C$ and $\mu_m^{(n)}$ by the map

$$\begin{array}{ccc} \mathcal{C}(C^n, C) \times \mathcal{C}(C^m, C)^n & \longrightarrow & \mathcal{C}(C^m, C) \\ (f; f_1, \ldots, f_n) & \longmapsto & f \circ \langle f_1, \ldots, f_n \rangle \end{array}$$

Thus, with respect to the above clone structures, the requirement that the semantics $\mathcal{E}$ be a clone homomorphisms amounts to the identity (9.1) and the semantic substitution lemma (18).

Monoids. The clone structure has equivalent representations as either of the following: finitary monads on **Set**, Lawvere theories, substitution algebras [13, Theorem 3.3], or, most importantly for this work, monoids in the monoidal closed category $(\mathbf{Set}^{\mathbb{F}}, \bullet, V)$ [13, Proposition 3.4], where the monoidal product is defined by the following coend:

$$(X \bullet Y)_m = \int^{n \in \mathbb{F}} X_n \times (Y_m)^n \qquad (m \in \mathbb{F}) \tag{20}$$

This tensor product and variations thereof play a crucial role in this paper; they arise from the following general situation (see, e.g., [23, I.5]):

$$\begin{tikzcd} \mathbf{1} \arrow[r, "(1)"] \arrow[drr, "C"', "\cong"] & \mathbb{F}^{\mathrm{op}} \arrow[r, hook, "\mathbf{y}"] \arrow[dr, "C^{\#}"] \arrow[r, phantom, shift right=4ex, "\overset{\mathrm{Lan}}{\cong}"] & \mathbf{Set}^{\mathbb{F}} \arrow[d, bend right, "-\bullet C"'] \arrow[d, phantom, "\perp"] \\ & & \mathcal{C} \arrow[u, bend right, "\langle C, - \rangle"'] \end{tikzcd} \tag{21}$$

where $\mathcal{C}$ is cartesian and cocomplete and where $C^{\#}$ denotes the cartesian extension of C.

Proposition 1.1 1. For $\mathcal{C}$ and $\mathcal{D}$ cartesian and cocomplete categories, and $F : \mathcal{C} \longrightarrow \mathcal{D}$ a cartesian functor with a right adjoint, we have a canonical natural isomorphism

$$\_ \bullet FC \cong F(\_ \bullet C)$$

for all $C \in \mathcal{C}$.

2. For a cartesian and cocomplete category $\mathcal{C}$ such that, for all $C \in \mathcal{C}$, the functor $\_ \times C$ is cocontinuous, we have the following equivalence of categories

$$\begin{array}{ccc} \mathcal{C} & \simeq & \mathsf{CarCoc}(\mathbf{Set}^{\mathbb{F}}, \mathcal{C}) \\ C & \longmapsto & \_ \bullet C \\ FV & \longleftarrow & F \end{array}$$

where CarCoc is the category of cartesian and cocontinuous functors, and natural transformations. □

Corollary 1.2 For every $X \in \mathbf{Set}^{\mathbb{F}}$ and $C \in \mathbf{Set}^{\mathbb{C}}$, there are canonical natural isomorphisms as follows

$$\begin{array}{ccc} (\_ \bullet X) \bullet C & \cong & \_ \bullet (X \bullet C) \\ \langle X, \langle C, \_ \rangle \rangle & \cong & \langle X \bullet C, \_ \rangle \end{array}$$ □

In this paper we will exclusively consider the above tensor construction when $\mathcal{C} = \mathbf{Set}^{\mathbb{C}}$, for some small category $\mathbb{C}$ (see [23, VII.2 and VIII.4] for a general discussion in the context of topos theory). In this case, the tensor $X \bullet C$ (for $X \in \mathbf{Set}^{\mathbb{F}}$ and $C \in \mathbf{Set}^{\mathbb{C}}$) has the following elementary description

$$\begin{array}{rcl} (X \bullet C)_m & = & \int^{n \in \mathbb{F}} X_n \times (C_m)^n \\ & \cong & (\coprod_{n \in \mathbb{N}} X_n \times (C_m)^n)_{/\approx} \end{array} \qquad (m \in \mathbb{C})$$

where $\approx$ is the equivalence relation generated by

$$(x; c_{\rho 1}, \ldots, c_{\rho n}) \sim (x[\rho]; c_1, \ldots, c_{n'}) \quad (\rho : n \to n')$$

Note that in particular taking $\mathbb{C} = \mathbb{F}$ and $C = Y \in \mathbf{Set}^{\mathbb{F}}$ we obtain the tensor (20) on $\mathbf{Set}^{\mathbb{F}}$. We will also use the case where $\mathbb{C} = \mathbf{1}$ (the terminal category), hence $\mathcal{C} \cong \mathbf{Set}$ and C is a set S:

$$X \bullet S = \int^{n \in \mathbb{F}} X_n \times S^n$$

As mentioned above, the categories of clones and monoids in $(\mathbf{Set}^{\mathbb{F}}, \bullet, V)$ are equivalent, hence the semantics $\mathcal{E} : E \longrightarrow \langle \mathbb{Z}, \mathbb{Z} \rangle$ is both a Σ-algebra homomorphism and a monoid homomorphism. In fact, by Theorem 4.1 of [13], the presheaf of expressions E is the initial object in the category of Σ-*monoids* (consisting of compatible Σ-algebra and monoid structures with corresponding homomorphisms). And, as the Σ-algebra structure in (14) for the clone of operations $\langle \mathbb{Z}, \mathbb{Z} \rangle$ is compatible with the clone/monoid structure of $\langle \mathbb{Z}, \mathbb{Z} \rangle$, the semantics $\mathcal{E}$ is the unique Σ-monoid homomorphism from E to $\langle \mathbb{Z}, \mathbb{Z} \rangle$.

1.4. Categorical operational semantics

It is shown in [35] that operational rules of the form (1) for signature and behaviour endofunctors Σ and B on a bicartesian category $\mathcal{C}$ induce a compositional semantics having the (full abstraction) property that two terms have the same meaning if and only if they are bisimilar, provided that (*i*) the forgetful functor B-Coalg $\longrightarrow \mathcal{C}$ has a right adjoint (hence a final coalgebra exists), and (*ii*) the behaviour B preserves weak pullbacks. The main tool we use to establish (*i*) for the behaviours in the present paper is the following.

Proposition 1.3 (See [24, 3]) For a finitary (resp. accessible) endofunctor B on a locally finitely presentable (resp. accessible) category $\mathcal{B}$, the forgetful functor B-Coalg $\longrightarrow \mathcal{B}$ has a right adjoint. □

The above mentioned (coalgebraic) notion of *bisimulation* is due to [2]. In this paper, we will consider it in the following form: a B-*bisimulation* between two coalgebras $h : X \longrightarrow BX$ and $k : Y \longrightarrow BY$ is a relation (i.e., equivalence class of monos) $R \hookrightarrow X \times Y$ between the carriers X and Y which lifts to the coalgebras in the sense that the diagram

$$\begin{tikzcd} X \arrow[d] & R \arrow[l] \arrow[r] \arrow[d, dotted] & Y \arrow[d] \\ BX & BR \arrow[l] \arrow[r] & BY \end{tikzcd}$$

commutes for some coalgebra structure on R. For the behaviour in (3) B-bisimulation is (strong) bisimulation.

2. Message passing bisimulations

2.1. Value passing

Late bisimulation. To model value-passing CCS, with respect to a set of values $\mathbb{V}$ and a *finite* set of channels C, we consider the behaviour endofunctor

$$BS = \wp_{\mathrm{f}}(C \times S^{\mathbb{V}} + C \times \mathbb{V} \times S + S) \tag{22}$$

on **Set**, where the components of the sum respectively model input, output, and silent actions. (Cf. [20].)

With respect to this behaviour functor, coalgebraic bisimulation corresponds to late bisimilarity. Indeed, a coalgebra $h : S \longrightarrow BS$ induces the late transition relation

$$s \xrightarrow{c?()} f \text{ iff } (c, f) \in h(s) \quad (c \in C,\, s \in S,\, f \in S^{\mathbb{V}})$$
$$s \xrightarrow{c!\langle v \rangle} s' \text{ iff } (c, v, s') \in h(s) \quad (c \in C,\, v \in \mathbb{V},\, s, s' \in S)$$
$$s \xrightarrow{\tau} s' \text{ iff } s' \in h(s) \quad (s, s' \in A)$$

that provides a characterisation of coalgebraic bisimulation in familiar terms (see [21]) as follows.

Proposition 2.1 The following data are equivalent.

1. A coalgebraic bisimulation for a coalgebra on S.
2. A symmetric relation $R \subseteq S \times S$ such that $s_0 \; R \; s_0'$ implies
   - if $s_0 \xrightarrow{c?()} f$ then there exists f' such that $s_0' \xrightarrow{c?()} f'$ and $f(v) \; R \; f'(v)$ for all $v \in \mathbb{V}$;
   - if $s_0 \xrightarrow{c!\langle v \rangle} s$ then there exists s' such that $s_0' \xrightarrow{c!\langle v \rangle} s'$ and $s' \; R \; s_0'$;
   - if $s_0 \xrightarrow{\tau} s$ then there exists s' such that $s_0' \xrightarrow{\tau} s'$ and $s \; R \; s'$. □

To appreciate the way in which (22) models the late interpretation of input, it is instructive to use the isomorphism $\wp_{\mathrm{f}}(S + S') \cong \wp_{\mathrm{f}}(S) \times \wp_{\mathrm{f}}(S')$ and consider the behaviour in the following form

$$BS \cong \wp_{\mathrm{f}}(S^{\mathbb{V}})^C \times \wp_{\mathrm{f}}(\mathbb{V} \times S)^C \times \wp_{\mathrm{f}}(S)$$

from which, as observed by Gordon Plotkin, one can read the late interpretation off the first component of the product corresponding to "first choosing a derivative and then receiving a value". To model the early interpretation of input, corresponding to "first receiving a value and then choosing a derivative", one thus needs to reverse the role of the type constructors for non-determinism and inaction, and input.

Early bisimulation. Noticing the following decomposition of the finite powerset functor

$$\wp_{\mathrm{f}} \cong 1 + \wp_{\mathrm{f}}^{+}$$

where $\wp_{\mathrm{f}}^{+}$ is the *non-empty* finite powerset functor, a natural behaviour for the early interpretation is then the endofunctor

$$BS = (1 + \wp_{\mathrm{f}}^{+}(S)^{\mathbb{V}})^C \times \wp_{\mathrm{f}}(\mathbb{V} \times S)^C \times \wp_{\mathrm{f}}(S)$$

which we will consider below in the following uniform form

$$\begin{aligned} BS \;\cong\; & (C \rightharpoonup \wp_{\mathrm{f}}^{+}(S)^{\mathbb{V}}) \\ & \times (C \rightharpoonup \wp_{\mathrm{f}}^{+}(\mathbb{V} \times S)) \\ & \times (1 \rightharpoonup \wp_{\mathrm{f}}^{+}(S)) \end{aligned} \tag{23}$$

where $\_ \rightharpoonup \_ : p\mathbf{Set}^{\mathrm{op}} \times p\mathbf{Set} \longrightarrow \mathbf{Set}$ is the *partial-exponential* functor (see e.g. [11]).

In this setting, a coalgebra $h : S \longrightarrow BS$ induces the early transition relation

$s \xrightarrow{c?\langle v\rangle} s'$ iff $s' \in \pi_1(hx)(c)(v)$ $(c \in C, v \in \mathbb{V}, s, s' \in S)$

$s \xrightarrow{c!\langle v\rangle} s'$ iff $(v, s') \in \pi_2(hx)(c)$ $(c \in C, v \in \mathbb{V}, s, s' \in S)$

$s \xrightarrow{\tau} s'$ iff $s' \in \pi_3(hx)()$ $(s, s' \in S)$

that provides a characterisation of coalgebraic bisimulation in familiar terms as follows.

Proposition 2.2 The following data are equivalent.

1. A coalgebraic bisimulation for a coalgebra on S.
2. A symmetric relation $R \subseteq S \times S$ such that $s_0 \; R \; s_0'$ implies
   - if $s_0 \xrightarrow{c?\langle v\rangle} s$ then there exists s' such that $s_0' \xrightarrow{c?\langle v\rangle} s'$ and $s \; R \; s'$;
   - if $s_0 \xrightarrow{c!\langle v\rangle} s$ then there exists s' such that $s_0' \xrightarrow{c!\langle v\rangle} s'$ and $s \; R \; s'$;
   - if $s_0 \xrightarrow{\tau} s$ then there exists s' such that $s_0' \xrightarrow{\tau} s'$ and $s \; R \; s'$. □

2.2. Name passing

Following [12], we will consider notions of behaviour for the π-calculus in the category of (variable sets) $\mathbf{Set}^{\mathbb{I}}$, where $\mathbb{I}$ is the category of finite cardinals and injections. However, all the constructions involved are also meaningful for pullback-preserving presheaves in $\mathbf{Set}^{\mathbb{I}}$ and so, following [33], we also obtain notions of behaviour in the Schanuel topos (see e.g. [23, pages 155 and 158]).

Late bisimulation. The constructions needed to model late bisimulation [27] as in [12] are:

- The type of *names* $N \in \mathbf{Set}^{\mathbb{I}}$ with identity action $N_n = n$.
- The *power* type $\wp_{\mathrm{f}} : \mathbf{Set}^{\mathbb{I}} \longrightarrow \mathbf{Set}^{\mathbb{I}}$ with pointwise action $(\wp_{\mathrm{f}} P)_n = \wp_{\mathrm{f}}(P_n)$.
- *Products* $(\times)$ and *coproducts* $(+)$ given pointwise by $(P \times Q)_n = P_n \times Q_n$ and $(P + Q)_n = P_n + Q_n$.
- The *exponential* P^N with action given by $(P^N)_n = (P_n)^n \times P_{n+1}$ and $P(\iota)(f, p) = (f', p')$ where
$$f'(x) = \begin{cases} (fa)[\iota] & \text{if } x = \iota a \\ p[\iota, x] & \text{otherwise} \end{cases} \quad \text{and} \quad p' = p[\iota + 1]$$
- The *dynamic allocation* type $\delta : \mathbf{Set}^{\mathbb{I}} \longrightarrow \mathbf{Set}^{\mathbb{I}}$ with action given by $(\delta P)_n = P_{n+1}$ and $(\delta P)(\iota) = P(\iota + 1)$.

The behaviour functor for late bisimulation of [12, 33] is

$$BP = \wp_{\mathrm{f}}(N \times P^N + N \times N \times P + N \times \delta P + P) \tag{24}$$

on $\mathbf{Set}^{\mathbb{I}}$. Hence we have that

$$\begin{aligned} BP_n = \wp_{\mathrm{f}}(\; & n \times (P_n)^n \times P_{n+1} \\ & + n \times n \times P_n \; + \; n \times P_{n+1} \\ & + P_n \;) \end{aligned}$$

in $\mathbf{Set}$.

A coalgebra $h : P \longrightarrow BP$ induces the late transition relation

$p \xrightarrow{a?()} f, p'$ iff $(a, f, p') \in h_n(p)$
$(a \in n, p \in P_n, f \in (P_n)^n, p' \in P_{n+1})$

$p \xrightarrow{a!\langle b\rangle} p'$ iff $(a, b, p') \in h_n(p)$
$(a, b \in n, p, p' \in P_n)$

$p \xrightarrow{a!()} p'$ iff $(a, p') \in h_n(p)$
$(a \in n, p \in P_n, p' \in P_{n+1})$

$p \xrightarrow{\tau} p'$ iff $p' \in h_n(p)$
$(p, p' \in P_n)$

that provides a characterisation of coalgebraic bisimulation in familiar terms (see [27]) as follows.

Proposition 2.3 The following data are equivalent.

1. A coalgebraic bisimulation for a coalgebra on P.
2. A family of symmetric relations
$$\{ R_n \subseteq P_n \times P_n \}_{n \in \mathbb{N}}$$
   such that, for every $n \in \mathbb{N}$,
   (a) $p \; R_n \; q$ implies $p[\iota] \; R_m \; q[\iota]$, for all $\iota : n \rightarrowtail m$ in $\mathbb{I}$;
   (b) $p \; R_n \; q$ implies
   - if $p \xrightarrow{a?()} f, p'$ then there exist g, q' such that $q \xrightarrow{a?()} g, q'$, and $f(a) \; R_n \; g(a)$ (for all $a \in n$) and $p' \; R_{n+1} \; q'$;
   - if $p \xrightarrow{a!\langle b\rangle} p'$ then there exists q' such that $q \xrightarrow{a!\langle b\rangle} q'$ and $p' \; R_{n+1} \; q'$;

- if $p \xrightarrow{a!()} p'$ then there exists q' such that $q \xrightarrow{a!()} q'$ and $p' \; R_{n+1} \; q'$;
- if $p \xrightarrow{\tau} p'$ then there exists q' such that $p' \xrightarrow{\tau} q'$ and $p' \; R_n \; q'$. $\square$

Early bisimulation. The definition of a behaviour functor for early bisimulation (left open in [12, 33]) requires the introduction of a new type constructor.

- For a *mono-preserving* presheaf $P : \mathbb{I} \longrightarrow \mathbf{Set}$ we define $P \rightharpoonup \_ : \mathbf{Set}^{\mathbb{I}} \longrightarrow \mathbf{Set}^{\mathbb{I}}$ as the functor mapping a presheaf Q to the presheaf $P \rightharpoonup Q$ with action given by $(P \rightharpoonup Q)_n = P_n \rightharpoonup Q_n$ and

  $$(P \rightharpoonup Q)(\iota) = P(\iota) \rightharpoonup Q(\iota) : u \longmapsto Q(\iota) \circ u \circ P(\iota)^R$$

  where $P(\iota)^R(q) = p$ iff $P(\iota)(p) = q$ (see [11]).

  This construction extends that of products in that we have an injection $P \times Q \rightarrowtail P \rightharpoonup Q$ given by:

  $$\begin{array}{ccc} P_n \times Q_n & \rightarrowtail & P_n \rightharpoonup Q_n \\ p, q & \longmapsto & p^R \rightharpoonup q \end{array} \qquad (25)$$

  where $(p^R \rightharpoonup q)(x) = (\text{if } x = p \text{ then } q)$.

In the vein of the treatment of early bisimulation for value-passing CCS given in (23), we consider the following behaviour functor

$$\begin{aligned} BP = & \; (N \rightharpoonup \wp_f^+(P)^N) \\ & \times (N \rightharpoonup \wp_f^+(N \times P)) \times (N \rightharpoonup \wp_f^+(\delta P)) \\ & \times (1 \rightharpoonup \wp_f^+(P)) \end{aligned} \qquad (26)$$

in $\mathbf{Set}^{\mathbb{I}}$, where the components of the product respectively model input, free and bound output, and silent actions. (The role of the constructor $N \rightharpoonup \_$ in this behaviour functor is analogous to the one of the topped tensor product $N \otimes^\top \_$ in the model of [18].)

Note that because of the following isomorphisms

$$\begin{array}{rcl} \wp_f(P+Q) & \cong & \wp_f(P) \times \wp_f(Q) \\ \wp_f(N \times P) & \cong & N \rightharpoonup \wp_f^+(P) \\ \wp_f(P) & \cong & 1 \rightharpoonup \wp_f^+(P) \end{array}$$

the *late* behaviour functor (24) can be written in the following form

$$\begin{aligned} & (N \rightharpoonup \wp_f^+(P^N)) \\ & \times (N \rightharpoonup \wp_f^+(N \times P)) \times (N \rightharpoonup \wp_f^+(\delta P)) \\ & \times (1 \rightharpoonup \wp_f^+(P)) \end{aligned}$$

which makes clear that the late and early interpretations of free and bound output, and of silent actions are the same.

Considering the pointwise early behaviour

$$\begin{aligned} BP_n = & \; (n \rightharpoonup (\wp_f^+ P_n)^n \times \wp_f^+ P_{n+1}) \\ & \times (n \rightharpoonup \wp_f^+(n \times P_n)) \\ & \times (n \rightharpoonup \wp_f^+ P_{n+1}) \\ & \times (1 \rightharpoonup \wp_f^+ P_n) \end{aligned}$$

a coalgebra $h : P \longrightarrow BP$ induces the early transition relation

$p \xrightarrow{a?\langle b \rangle} p'$ iff $p' \in \pi_1(\pi_1(h_n p)(a))(b)$ $\quad (a, b \in n, p, p' \in P_n)$

$p \xrightarrow{a?()} p'$ iff $p' \in \pi_2(\pi_1(h_n p)(a))$ $\quad (a \in n, p \in P_n, p' \in P_{n+1})$

$p \xrightarrow{a!\langle b \rangle} p'$ iff $(b, p') \in \pi_2(h_n p)(a)$ $\quad (a, b \in n, p, p' \in P_n)$

$p \xrightarrow{a!()} p'$ iff $p' \in \pi_3(h_n p)(a)$ $\quad (a \in n, p \in P_n, p' \in P_{n+1})$

$p \xrightarrow{\tau} p'$ iff $p' \in \pi_4(h_n p)()$ $\quad (p, p' \in P_n)$

that provides a characterisation of coalgebraic bisimulation in familiar terms (see [27]) as follows.

Proposition 2.4 The following data are equivalent.

1. A coalgebraic bisimulation for a coalgebra on P.
2. A family of symmetric relations

   $$\{ R_n \subseteq P_n \times P_n \}_{n \in \mathbb{N}}$$

   such that, for every $n \in \mathbb{N}$,

   (a) $p \; R_n \; q$ implies $p[\iota] \; R_m \; q[\iota]$, for all $\iota : n \rightarrowtail m$ in $\mathbb{I}$;

   (b) $p \; R_n \; q$ implies

   - if $p \xrightarrow{a?\langle b \rangle} p'$ then there exists q' such that $q \xrightarrow{a?\langle b \rangle} q'$ and $p' \; R_n \; q'$;
   - if $p \xrightarrow{a?()} p'$ then there exists q' such that $q \xrightarrow{a?()} q'$ and $p' \; R_{n+1} \; q'$;
   - if $p \xrightarrow{a!\langle b \rangle} p'$ then there exists q' such that $q \xrightarrow{a!\langle b \rangle} q'$ and $p' \; R_n \; q'$;
   - if $p \xrightarrow{a!()} p'$ then there exists q' such that $q \xrightarrow{a!()} q'$ and $p' \; R_{n+1} \; q'$;
   - if $p \xrightarrow{\tau} p'$ then there exists q' such that $q \xrightarrow{\tau} q'$ and $p' \; R_n \; q'$. $\square$

3 Semantics of name passing

To model the structural operational rules for the π-calculus using natural transformations of type (1), we are faced with the fact that the signature Σ is an endofunctor on $\mathbf{Set}^{\mathbb{F}}$ (see (16)) while the behaviour B (for both the late (24) and the and the early (26) interpretations) is an endofunctor on $\mathbf{Set}^{\mathbb{I}}$. Far from being a problem, this disparity allows for the desired compositionality result to hold. Indeed, both late and early bisimulations are *not* congruences. What we need are thus behaviour functors for late and early *congruences* instead. These behaviours can be obtained by (right) extending the B's on $\mathbf{Set}^{\mathbb{I}}$ along an adjunction $\mathbf{Set}^{\mathbb{F}} \underset{\longrightarrow}{\overset{\longleftarrow}{\top}} \mathbf{Set}^{\mathbb{I}}$ obtaining new endofunctors $\widetilde{B}$'s on $\mathbf{Set}^{\mathbb{F}}$. Moreover, a natural transformation of type

$$\Sigma(X \times \widetilde{B}X) \longrightarrow \widetilde{B}TX \tag{27}$$

in $\mathbf{Set}^{\mathbb{F}}$ will be suitable to model the desired structural operational rules for the π-calculus.

Late and early congruences. The adjunction we need between $\mathbf{Set}^{\mathbb{F}}$ and $\mathbf{Set}^{\mathbb{I}}$ is an instance of the adjunction in (21) taking $C = \mathbf{Set}^{\mathbb{I}}$ and $C = N$:

$$\mathbf{Set}^{\mathbb{F}} \underset{\_\bullet N}{\overset{\langle N, \_\rangle}{\underset{\longrightarrow}{\overset{\longleftarrow}{\top}}}} \mathbf{Set}^{\mathbb{I}} \tag{28}$$

Alternatively, one can describe this adjunction as the essential geometric morphism (see, e.g., [23, page 360]) associated to the inclusion $\mathbb{I} \longrightarrow \mathbb{F}$. Thus, we have a canonical natural isomorphism

$$X \bullet N \cong |X| \tag{29}$$

(essentially given by the action $X_n \times m^n \longrightarrow X_m$ of X) where $|\_| : \mathbf{Set}^{\mathbb{F}} \longrightarrow \mathbf{Set}^{\mathbb{I}}$ is the *forgetful* functor given by precomposing with the inclusion $\mathbb{I} \longrightarrow \mathbb{F}$.

We can now define, for every endofunctor B on $\mathbf{Set}^{\mathbb{I}}$, an endofunctor

$$\widetilde{B}X = \langle N, B|X|\rangle$$

i.e., the *right Kan extension* of $\langle N, B\_\rangle$ along $\langle N, \_\rangle$. Using the isomorphism (29) and the adjunction (28), the $\widetilde{B}$-coalgebras are in bijective correspondence with B-coalgebras $|X| \longrightarrow B|X|$. In other words, $\widetilde{B}$-coalgebras are B-coalgebras on presheaves with an action along *all* renamings (rather than only on injective ones). This makes a crucial difference in terms of coalgebraic bisimulation.

Proposition 3.1 For B as in (24) [resp. (26)], the following data are equivalent.

1. A coalgebraic $\widetilde{B}$-bisimulation for a coalgebra $X \longrightarrow \widetilde{B}X$.
2. A family of symmetric relations $\{R_n \subseteq X_n \times X_n\}_{n\in\mathbb{N}}$ as in Proposition 2.3 (2) [resp. Proposition 2.4 (2)] (with respect to the transposed B-coalgebra $|X| \longrightarrow B|X|$) where the closure condition (a) is generalised to
   $p\ R_n\ q$ implies $p[\rho]\ R_m\ q[\rho]$, for all $\rho : n \longrightarrow m$ in $\mathbb{F}$. □

Proposition 3.2 1. The functors $(\_)^N : \mathbf{Set}^{\mathbb{I}} \longrightarrow \mathbf{Set}^{\mathbb{I}}$ and $N^n \Rightarrow \_ : \mathbf{Set}^{\mathbb{I}} \longrightarrow \mathbf{Set}^{\mathbb{I}}$ ($n \in \mathbb{N}$) are finitary.

2. For B as in (24) and (26), the lifted functors $\widetilde{B}$ are finitary (hence the forgetful functor $\widetilde{B}$-Coalg $\longrightarrow \mathbf{Set}^{\mathbb{F}}$ has a right adjoint) and preserve weak pullbacks.

□

Therefore, every natural transformation of type (27), with B the late (early) behaviour functor, induces a compositional semantics fully abstract with respect to late (early) congruence.

Categorical rules. We sketch how the π-calculus operational rules [27] are modelled by a natural transformation of type (27). For brevity, we only consider the operational rules of the binding operators (input and restriction); the operational rules for the other operators are modelled along the lines of [34] using the isomorphisms

$$\langle C, D_1\rangle \times \langle C, D_2\rangle \cong \langle C, D_1 \times D_2\rangle$$
$$\delta\langle C, D\rangle \cong \langle C, D^C\rangle$$

satisfied by the functors in (13) with C cartesian closed, and the map

$$V \times X \longrightarrow \langle N, N \Rightarrow |X|\rangle \tag{30}$$

obtained by transposing $|V \times X| \cong N \times |X| \rightarrowtail N \Rightarrow |X|$, where the injection is given by (25).

Input. For input, the rule is modelled by a map of type

$$V \times \delta(X \times \widetilde{B}X) \longrightarrow \langle N, B|TX|\rangle$$

Using (30) and projecting out the components that do not contribute to the rule we can focus on defining a map of type

$$\delta X \longrightarrow \langle N, |X|^N\rangle \cong \delta\langle N, |X|\rangle$$

The required map is δ applied to the unit $X \rightarrowtail \langle N, |X|\rangle$ of the adjunction (28); that is,

$$\begin{array}{rcl} X_{n+1} & \longrightarrow & \mathbf{Set}^{\mathbb{I}}(N^{n+1}, |X|) \\ x & \longmapsto & \{\, \lambda\rho \in m^{n+1}.\, x[\rho]\, \}_{m\in\mathbb{I}} \end{array}$$

Note that this map can be used both for the late and early cases by precomposing it with suitable maps respectively arising from the injections $|X|^N \rightarrowtail \wp_{\mathrm{f}}^{+}(|X|^N)$ and $|X|^N \rightarrowtail (\wp_{\mathrm{f}}^{+}|X|)^N$.

Restriction. For restriction, the rule is modelled by a map of type $\delta(X \times \widetilde{B}X) \longrightarrow \langle N, B|TX|\rangle$ in $\mathbf{Set}^{\mathbb{F}}$ which, in fact, comes from a map of type

$$\delta B|X| \longrightarrow B|TX| \quad \text{in } \mathbf{Set}^{\mathbb{I}}$$

For instance, the core of this latter map corresponding to the following two rules

$$(\text{RES})\ \frac{\mathrm{P} \xrightarrow{\bar{a}b} \mathrm{Q}}{(x)\mathrm{P} \xrightarrow{\bar{a}b} (x)\mathrm{Q}}\ x \not\equiv a, b \qquad (\text{OPEN})\ \frac{\mathrm{P} \xrightarrow{\bar{a}x} \mathrm{Q}}{(x)\mathrm{P} \xrightarrow{\bar{a}(x)} \mathrm{Q}}\ x \not\equiv a$$

is the map

$$\delta N \times \delta N \times \delta|X| \xrightarrow{\mathrm{RO}} \wp_{\mathrm{f}}(N \times N \times |TX| + N \times \delta|TX|)$$

defined, using the internal language (see [12]), as follows:

$$\begin{array}{l} \mathsf{RO}(a,b,q) \\ \quad = \ \mathsf{case}\ a\ \mathsf{of} \\ \qquad \mathsf{old}(a') \ \Rightarrow \ \mathsf{let}\ q' = \delta\, \boldsymbol{\eta}\, q \\ \qquad\qquad\qquad\quad \mathsf{in}\ \ \mathsf{case}\ b\ \mathsf{of} \\ \qquad\qquad\qquad\qquad \mathsf{old}(b') \ \Rightarrow \ \{\, (a', b', \boldsymbol{\nu}\, q')\, \} \\ \qquad\qquad\qquad\qquad\quad \mathsf{new} \ \Rightarrow \ \{\, (a', q')\, \} \\ \qquad\quad \mathsf{new} \ \Rightarrow \ \emptyset \end{array}$$

where $\boldsymbol{\eta} : |X| \longrightarrow |TX|$ and $\boldsymbol{\nu} : \delta|TX| \longrightarrow |TX|$ (in $\mathbf{Set}^{\mathbb{I}}$) are respectively the (underlying maps of the) unit and the restriction operator (in $\mathbf{Set}^{\mathbb{F}}$) of the free Σ-algebra TX on X.

4. Semantics of value passing

Actions. We have seen in §1.1 that the homogeneous substitution of expressions for variables in expressions can be modelled as monoids. For the heterogeneous substitution of expressions for variables in terms we can use *monoid actions* as follows. Every monoid $M = (M, \mu, \nu)$ in $\mathbf{Set}^{\mathbb{F}}$ defines a monad $\_ \bullet M$ on $\mathbf{Set}^{\mathbb{F}}$. The category of algebras of this monad M-**Act**, consists of (*right*) *actions* $A \bullet M \longrightarrow A$ [22, VII.4]. In elementary terms, this amounts to a family $\{\, \alpha_m^{(n)} : A_n \times (M_m)^n \longrightarrow A_m \,\}_{n,m \in \mathbb{N}}$ of operations such that

$$\alpha_m(a; \nu_1, \ldots, \nu_n) = a$$

$$\alpha_\ell(\alpha_m(a; \vec{u}); \vec{v}) = \alpha_\ell(a; \mu_\ell(u_1; \vec{v}), \ldots, \mu_\ell(u_n; \vec{v}))$$

for all a in A_n, $\vec{u}$ in $(M_m)^n$, and $\vec{v}$ in $(M_\ell)^m$. (Note the occurrence of μ in the second law.)

For examples of actions consider the following.

A V-action $A \bullet V \longrightarrow A$ is forced, by the unit law, to be the canonical isomorphism $A \bullet V \cong A$. Thus, the category V-**Act** is isomorphic to $\mathbf{Set}^{\mathbb{F}}$; which explains why, for name passing, we can do without extra substitution structure.

For objects C and D in a cartesian category $\mathcal{C}$, the monoid $\langle C, C\rangle$ has a canonical action on the presheaf $\langle C, D\rangle$ given by (pairing and) composition in $\mathcal{C}$.

As in any bicomplete monoidal closed category (cf. [23, VII.3]), a monoid homomorphism $M' \longrightarrow M$ induces a *reindexing* functor M-**Act** $\longrightarrow M'$-**Act** with both left and right adjoints. Thus, the semantics of expressions $E \longrightarrow \langle \mathbb{Z}, \mathbb{Z}\rangle$ and the unique homomorphism $V \longrightarrow M$ induce the following adjoint situations

$$\langle \mathbb{Z}, \mathbb{Z}\rangle\text{-}\mathbf{Act} \ \underset{\longleftarrow}{\overset{\longleftarrow}{\underset{\top}{\overset{\top}{\longrightarrow}}}} \ E\text{-}\mathbf{Act}\,, \qquad M\text{-}\mathbf{Act} \ \underset{\underset{\_\bullet M}{\longleftarrow}}{\overset{\overset{\langle M,\_\rangle}{\longleftarrow}}{\underset{\top}{\overset{\top}{\longrightarrow}}}} \ \mathbf{Set}^{\mathbb{F}}$$

where, on the right hand side, $X \bullet M$ has action given by multiplication and $\langle M, X\rangle$ has action given by multiplication and evaluation.

Syntax. The substitution of expressions in terms involves, in turn, a substitution of expressions in expressions. Thus, the signature bifunctor for value-passing CCS needs to be parametric in a *monoid* of messages. Accordingly, we let Σ be the bifunctor $\mathbf{Mon}(\mathbf{Set}^{\mathbb{F}}) \times \mathbf{Set}^{\mathbb{F}} \longrightarrow \mathbf{Set}^{\mathbb{F}}$ given by (17).

For a monoid M, we write Σ_M for the functor $\Sigma(M, \_) : \mathbf{Set}^{\mathbb{F}} \longrightarrow \mathbf{Set}^{\mathbb{F}}$. One can lift Σ_M to the category M-**Act** of M-actions by means of a distributive law

$$\lambda : \Sigma_M(\_) \bullet M \Longrightarrow \Sigma_M(\_ \bullet M)$$

of the endofunctor Σ_M over the monad induced by the monadic adjunction M-**Act** $\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}$ $\mathbf{Set}^{\mathbb{F}}$. This distributive law is essentially the strength described in [13, page 200], with the extra use of the multiplication of the monoid M in the fourth and fifth summand of Σ_M. The resulting endofunctor

$$\begin{array}{l} \overline{\Sigma}_M(A \bullet M \xrightarrow{\alpha} A) \\ \qquad = (\, \Sigma_M(A) \bullet M \xrightarrow{\lambda_A} \Sigma_M(A \bullet M) \xrightarrow{\Sigma\alpha} \Sigma A \,) \end{array}$$

on M-**Act** has as algebras presheaves A with both a Σ-algebra structure and an M-action compatible with each other in the sense that the evident diagram

$$\begin{tikzcd} \Sigma_M(A) \bullet M \arrow[r] \arrow[d] & \Sigma_M(A \bullet M) \arrow[r] & \Sigma_M(A) \arrow[d] \\ A \bullet M \arrow[rr] & & A \end{tikzcd}$$

commutes. We denote the corresponding category of $\overline{\Sigma}_M$-algebras by $\overline{\Sigma}_M$-Alg. The associated forgetful functor $\overline{\Sigma}_M$-Alg $\longrightarrow M$-**Act** has a left adjoint; and the induced monad is denoted by $\overline{T}_M$, as it is a lifting of the monad T_M induced by Σ_M.

Moreover, every monoid homomorphism $M' \longrightarrow M$ induces a reindexing functor $\overline{\Sigma}_M\text{-Alg} \longrightarrow \overline{\Sigma}_{M'}\text{-Alg}$, which is a lifting of the reindexing functor $M\text{-}\mathbf{Act} \longrightarrow M'\text{-}\mathbf{Act}$. In particular, the reindexing functor $\overline{\Sigma}_{\langle\mathbb{Z},\mathbb{Z}\rangle}\text{-Alg} \longrightarrow \overline{\Sigma}_E\text{-Alg}$ induced by the semantics of expressions $E \longrightarrow \langle\mathbb{Z},\mathbb{Z}\rangle$ allows us to turn every interpretation for $\overline{T}_{\langle\mathbb{Z},\mathbb{Z}\rangle}(0)$ into one for $\overline{T}_E(0)$.

Semantics. Let M be a monoid of messages in $\mathbf{Set}^{\mathbb{F}}$; a typical example being the clone of operations $\langle\mathbb{V},\mathbb{V}\rangle$ on a set of values $\mathbb{V}$.

We have the following situation (cf. (5))

$$\begin{tikzcd}
M\text{-}\mathbf{Act} \arrow[loop below, "\overline{\Sigma}_M"] \arrow[r, shift left=2, "\top"] & \mathbf{Set}^{\mathbb{F}} \arrow[l, shift right=4, "\langle M,_\rangle"'] \arrow[l, shift left=4, "_\bullet M"] \arrow[loop below, "\Sigma_M"] \arrow[r, shift right=2, "_\bullet 0"'] & \mathbf{Set} \arrow[l, shift right=2, "\langle 0,_\rangle"'] \arrow[loop above, "B"]
\end{tikzcd}$$

where the adjunction on the right can be alternatively described as the essential geometric morphism associated to the functor $(0) : \mathbf{1} \longrightarrow \mathbb{F}$; hence

$$X \bullet 0 \cong X_0$$

for all $X \in \mathbf{Set}^{\mathbb{F}}$.

To have both syntax and behaviour on the same category, we will proceed as in the previous section and (right) extend behaviour functors B on $\mathbf{Set}$ along the composite adjunction $M\text{-}\mathbf{Act} \underset{\longrightarrow}{\overset{\longleftarrow}{\top}} \mathbf{Set}$ to $\widetilde{B}$ on $M\text{-}\mathbf{Act}$. To do this easily, we need a lemma.

Lemma 4.1 For $\mathcal{C}$ cartesian and cocomplete, the composite adjunction $M\text{-}\mathbf{Act} \underset{|\_|}{\overset{\langle M,\_\rangle}{\rightleftarrows}} \mathbf{Set}^{\mathbb{F}} \underset{\_\bullet\mathcal{C}}{\overset{\langle \mathcal{C},\_\rangle}{\rightleftarrows}} \mathcal{C}$ is given by

$|\_| \bullet \mathcal{C} : M\text{-}\mathbf{Act} \underset{\longrightarrow}{\overset{\longleftarrow}{\top}} \mathcal{C} : \langle M \bullet \mathcal{C}, \_\rangle$. □

It follows that the extension of a behaviour functor B on $\mathbf{Set}$ is along the adjunction

$$|\_|_0 : M\text{-}\mathbf{Act} \underset{\longrightarrow}{\overset{\longleftarrow}{\top}} \mathbf{Set} : \langle M_0, \_\rangle \qquad (31)$$

where M_0 is the set of ground messages, yielding $\widetilde{B}$ on $M\text{-}\mathbf{Act}$ to be given by

$$\widetilde{B}A = \langle M_0, B(A_0)\rangle$$

Late and early congruences. As operational models for value passing we take $\widetilde{B}$-coalgebras

$$A \longrightarrow \langle M_0, B(A_0)\rangle$$

in $M\text{-}\mathbf{Act}$ where B is either of the two endofunctors on $\mathbf{Set}$ of (22) and (23). The adjunction (31) allows us to express these operational models in terms of coalgebras on $\mathbf{Set}$. Indeed, they are in bijective correspondence with functions

$$A_0 \longrightarrow B(A_0)$$

where A carries an M-action. Moreover, $\widetilde{B}$-coalgebra homomorphisms are action homomorphisms which at stage 0 are also B-coalgebra homomorphisms:

$$\begin{tikzcd}
A \bullet M \arrow[r] \arrow[d] & A \arrow[d] & A_0 \arrow[r] \arrow[d] & B(A_0) \arrow[d] \\
A' \bullet M \arrow[r] & A' & A'_0 \arrow[r] & B(A'_0)
\end{tikzcd}$$

Proposition 4.2 For B as in (22) [resp. (23)], the following data are equivalent.

1. A coalgebraic $\widetilde{B}$-bisimulation for a coalgebra $A \longrightarrow \widetilde{B}A$.
2. A family of symmetric relations $\{R_n \subseteq A_n \times A_n\}_{n\in\mathbb{N}}$ such that
   (a) R_0 is as in Proposition 2.1 (2) [resp. Proposition 2.2 (2)] (with respect to the transposed B-coalgebra $A_0 \longrightarrow B(A_0)$).
   (b) For every $n \in \mathbb{N}$, $s \; R_n \; s'$ implies
   $\alpha_m(s;\vec{v}) \; R_m \; \alpha_m(s';\vec{v})$, for all $\vec{v}$ in $(M_m)^n$. □

Proposition 4.3 1. The category of actions $M\text{-}\mathbf{Act}$ is locally finitely presentable.

2. For B as in (22) and (23), the extended functors $\widetilde{B}$ are accessible (hence the forgetful functor $\widetilde{B}\text{-Coalg} \longrightarrow M\text{-}\mathbf{Act}$ has a right adjoint) and preserve weak pullbacks. □

Categorical rules. Natural transformations in $M\text{-}\mathbf{Act}$ of type

$$\overline{\Sigma}(A \times \widetilde{B}A) \longrightarrow \widetilde{B}\overline{T}A \qquad (32)$$

with B the late (early) behaviour functor with set of values $\mathbb{V} = M_0$, are suitable to model structural operational rules for languages with value passing and give a categorical format inducing fully-abstract compositional semantics with respect to late (early) congruence.

Input. The most interesting rule to model is the axiom for input. As for the π-calculus, the core of this rule (both for the late and early behaviour) lies in the map

$$\overline{\delta}A \longrightarrow \langle\mathbb{V}, {A_0}^{\mathbb{V}}\rangle \cong \overline{\delta}\langle\mathbb{V}, A_0\rangle$$

obtained by applying $\overline{\delta}$ to the unit of the adjunction (31), namely:

$$\begin{array}{rcl} A_{n+1} & \longrightarrow & \mathbf{Set}(\mathbb{V}^{n+1}, A_0) \\ a & \longmapsto & \lambda\vec{v} \in \mathbb{V}^{n+1}.\, \alpha_0(a;\vec{v}) \end{array}$$

Acknowledgements. We gratefully acknowledge discussions with Gian-Luca Cattani, Gordon Plotkin and Davide Sangiorgi.

References

[1] P. Aczel. *Non-well-founded sets.* Number 14 in Lecture Notes. CSLI, 1988.

[2] P. Aczel and P. F. Mendler. A final coalgebra theorem. In D. H. Pitt, D. E. Rydeheard, P. Dybjer, A. M. Pitts, and A. Poigné, editors, *Proc. Category Theory and Computer Science*, volume 389 of *LNCS*, pages 357–365. Springer-Verlag, 1989.

[3] J. Adámek and J. Rosický. *Locally Presentable and Accessible Categories*, volume 189 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 1994.

[4] J. A. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60:109–137, 1984.

[5] K. Bernstein. A congruence theorem for structured operational semantics of higher-order languages. In *Proc. 13^{th} LICS Conf.*, pages 153–164. IEEE, Computer Society Press, 1998.

[6] B. Bloom, S. Istrail, and A. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, 1995.

[7] G.-L. Cattani and P. Sewell. Models for name-passing processes: Interleaving and causal. In *Proc. 15^{th} LICS Conf.*, pages 322–333. IEEE, Computer Society Press, 2000.

[8] G.-L. Cattani, I. Stark, and G. Winskel. Presheaf models for the π-calculus. In E. Moggi and G. Rosolini, editors, *Proc. Category Theory and Computer Science*, volume 1290 of *LNCS*, pages 106–126. Springer-Verlag, 1997.

[9] P. Cohn. *Universal Algebra.* Harper & Row, 1965.

[10] R. de Simone. Higher level synchronising devices in MEIJE-SCCS. *Theoretical Computer Science*, 37:245–267, 1985.

[11] M. Fiore. *Axiomatic Domain Theory in Categories of Partial Maps.* Distinguished Dissertations Series. Cambridge University Press, 1996.

[12] M. Fiore, E. Moggi, and D. Sangiorgi. A fully-abstract model for the π-calculus. In *Proc. 11^{th} LICS Conf.*, pages 43–54. IEEE, Computer Society Press, 1996. (Full version to appear in *Information and Computation*).

[13] M. Fiore, G. Plotkin, and D. Turi. Abstract syntax and variable binding. In *Proc. 14^{th} LICS Conf.*, pages 193–202. IEEE, Computer Society Press, 1999.

[14] W. Fokkink and R. van Glabbeek. Ntyft/ntyxt rules reduce to ntree rules. *Information and Computation*, 126(1):1–10, 1996.

[15] W. Fokkink and C. Verhoef. A conservative look at operational semantics with variable binding. *Information and Computation*, 146(1):24–54, 1998.

[16] J. Goguen, J. Thatcher, and E. Wagner. An initial algebra approach to the specification, correctness and implementation of abstract data types. In R. Yeh, editor, *Current Trends in Programming Methodology*, volume IV, pages 80–149. Prentice Hall, 1978.

[17] J. Groote and F. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100(2):202–260, 1992.

[18] M. Hennessy. A fully abstract denotational semantics for the π-calculus. Technical Report 4, COGS, University of Sussex, 1996. To appear in *Theoretical Computer Science.*

[19] C. A. R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.

[20] A. Ingólfsdóttir. A semantic theory for value-passing processes, Late approach, Part I: A denotational model and its complete axiomatization. Report Series RS-95-3, BRICS, Department of Computer Science, University of Aarhus, 1995.

[21] A. Ingólfsdóttir. A semantic theory for value-passing processes, Late approach, Part II: A behavioural semantics and full abstractness. Report Series RS-95-22, BRICS, Department of Computer Science, University of Aarhus, 1995.

[22] S. Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag, 1971.

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

[24] M. Makkai and R. Paré. *Accessible Categories: The Foundations of Categorical Model Theory*, volume 104 of *Contemporary Math.* Amer. Math. Soc., 1989.

[25] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS.* Springer-Verlag, 1980.

[26] R. Milner. *Communication and Concurrency.* International Series in Computer Science. Prentice Hall, 1989.

[27] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, I and II. *Information and Computation*, 100(1):1–77, Sept. 1992.

[28] U. Montanari and M. Pistore. π-calculus, structured coalgebras and minimal HD-automata. In *Mathematical Foundations of Computer Science 2000*, volume 1893 of *Lecture Notes in Computer Science.* Springer-Verlag, 2000.

[29] G. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Department of Computer Science, Aarhus University, 1981.

[30] G. Plotkin. Binding algebras: A step from universal algebra to type theory. Slides of a lecture at RTA'98, 1998.

[31] G. Plotkin. Bialgebraic semantics and recursion. Invited talk at *4^{th} Workshop on Coalgebraic Methods in Computer Science*, Genova, Italy, 2001.

[32] D. Scott. Outline of a mathematical theory of computation. In *Proc. 4^{th} Annual Princeton Conference on Inf. Sciences and Systems*, pages 169–176, 1970.

[33] I. Stark. A fully abstract domain model for the π-calculus. In *Proc. 11^{th} LICS Conf.*, pages 36–42. IEEE, Computer Society Press, 1996.

[34] D. Turi. Categorical modelling of structural operational rules: case studies. In E. Moggi and G. Rosolini, editors, *Proc. Category Theory and Computer Science*, volume 1290 of *LNCS*, pages 127–146. Springer-Verlag, 1997.

[35] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. 12^{th} LICS Conf.*, pages 280–291. IEEE, Computer Society Press, 1997.

[36] D. Turi and J. Rutten. On the foundations of final coalgebra semantics: non-well-founded sets, partial orders, metric spaces. *Mathematical Structures in Computer Science*, 8(5):481–540, 1998.