Thesis description:

Name-passing process calculi: operational models and structural operational semantics.

Sam Staton.

Technical Report UCAM-CL-TR-688. University of Cambridge Computer Laboratory, June 2007.

Document prepared March 2008. Contact: sam.staton@cl.cam.ac.uk.

Summary

My thesis is about foundations for formal semantics of name-passing process calculi. These calculi are languages for describing systems of agents that communicate channel names along named channels. This facility provides a natural way of describing the mobility of communication links. (The π-calculus of Milner et al. [1992] is a paradigmatic example of such a language.)

The thesis is split into two parts, reflecting the two aspects of the foundations of name-passing calculi that are addressed.

- Part I of the thesis is dedicated to *operational models for name-passing calculi*. Conventional operational models, such as labelled transition systems, are inappropriate for name-passing systems. For this reason I develop and relate two different models of name-passing from the literature: indexed labelled transition systems, based on work of Cattani and Sewell [2004], and a coalgebraic approach introduced by Fiore and Turi [2001]. Connections are made with the History Dependent Automata of Montanari and Pistore [2005], and I introduce a new operational model using the nominal logic of Pitts [2003].

- Part II of the thesis concerns *structural operational semantics for name-passing calculi*. Various work has been done on the meaning of rule-based transition system specifications, and on congruence rule formats for simple calculi. This work is not relevant to name-passing systems, due to (for instance) variable-binding and substitution in the syntax, and side conditions on rules. I investigate the application of Turi and Plotkin's [1997] 'mathematical structural operational semantics' to the name-passing case, by developing it for the operational models introduced in the first part. An important result is extracted from the analysis of the structural operational semantics. If a specification of a name-passing calculus is given in a certain format, then bisimilarity (a natural notion of equivalence) is guaranteed to be a congruence.

While the thesis draws important conclusions about the foundations for name-passing calculi, it must be emphasised that the ramifications of this work spread far beyond this. Many features of name-passing calculi, particularly the variable-binding and substitution, are relevant to modern programming languages, and also in languages for cryptographic protocols. By studying foundations for name-passing calculi, I provide a step towards the foundations of these more elaborate languages and idioms. Indeed, this the subject of my current research.

Notes for the reader. This report is structured as follows. After introducing some preliminary concepts (the π-calculus and nominal logic), I provide summaries of the two parts of the thesis. Following this, I provide more detailed, technical descriptions of the two parts, with more careful cross-references to the thesis body. I conclude with a brief summary of ongoing work related to the thesis.

In this report, I use alphabetic section numbering, to distinguish from the numeric numbering system of the thesis. All numeric references are references to sections or results of the thesis.

Contents

A. Preliminaries 2
A.a. Name-passing and the π-calculus 2
A.b. Nominal logic and nominal sets 2
B. Summary of Part I (Operational models) 3
C. Summary of Part II (Structural operational semantics) 6
D. Detailed overview of Part I (Operational models) 7
D.a. Presheaves of states 7
D.b. Labelled transition systems for name-passing 8
D.c. Coalgebras in general 10
D.d. Coalgebras for name-passing 11
D.e. Refining the state spaces: sheaves, nominal substitutions, and named-sets . . 12
E. Detailed overview of Part II (Structural operational semantics) 15
E.a. First-order framework 15
E.b. GSOS rules for name-passing 17
F. Beyond the thesis 20

A. Preliminaries

A.a. Name-passing and the π-calculus

The paradigmatic example of a name-passing calculus is the π-calculus [Milner et al., 1992]. I include here some rudiments of the π-calculus, to give an informal impression of what is involved in the syntax and semantics of name-passing calculi. More details are given in Sec. 3.1 of the thesis and the references therein.

The π-calculus describes the communication of channel names along (named) channels. For instance, consider the processes

$P = c(a).P'$ which receives a name on channel c and binds it to a in P'.
$Q = \bar{c}d.Q'$ which transmits name d on channel c, becoming Q'.

There is a transition $(P \mid Q) \xrightarrow{\tau} (([d/a]P') \mid Q')$, during which P and Q communicate. Notice that, after the transition, the transmitted name d is substituted for the bound name a in P'.

The restriction operator, ν, acts to hide names when they are used as channels. But when names are considered as data, the operator ν has a different behaviour. The process $\nu d.\bar{c}d$ can perform an output action; here it is helpful to think of the operator ν as describing the generation of a new name d. The output of newly generated names is called *bound output*. This behaviour gives rise to the phenomenon of *scope extrusion*. For instance, in the π-calculus, we have a silent transition

$$(\nu d.\bar{c}d.P') \mid c(d).Q' \xrightarrow{\tau} \nu d.(P' \mid Q')$$

during which the scope of d extrudes to include Q'. This phenomenon can be used to describe mobility of communication links.

A.b. Nominal logic and nominal sets

Nominal logic was introduced by Pitts [2003] as a first-order theory of variable binding. An overview is given in Sec. 7.2 of the thesis, and I now give a brief summary.

Nominal logic. A nominal logic theory is a theory of many-sorted first-order logic that has particular symbols and axioms. There must be a sort N of names, and, for each sort X there must be a sort $[\mathsf{N}]\mathsf{X}$. If a is a term of sort N, and t is a term of some other sort X, then we have a term $\langle a\rangle t$ of sort $[\mathsf{N}]\mathsf{X}$ and a formula $a \# t$. The term $\langle a\rangle t$ is to be thought of as "a bound in t, up-to α-equivalence". The formula $a \# t$ is to be thought of as "a is fresh for t". For example, a simple theorem of nominal logic is:

$$\forall a : \mathsf{N}.\forall x : \mathsf{X}.\ \ a\#(\langle a\rangle x) \quad .$$

Nominal sets. A nominal logic theory can be interpreted in set theory, in the category of sets, just as any other first-order theory. A more natural category for nominal logic, though, is the category of *nominal sets*, as introduced by Pitts [2006] and Gabbay. The definitions are recalled in Secs. 7.1.1–7.1.3 of the thesis, and I now give an overview.

To begin, we fix an infinite set $\mathcal{N}$, the set of 'names', and we write $\mathsf{Sym}(\mathcal{N})$ for the group of all permutations of $\mathcal{N}$. A $\mathsf{Sym}(\mathcal{N})$-set is a set X together with a function $\bullet_X : \mathsf{Sym}(\mathcal{N}) \times X \to X$. *Equivariant functions* between $\mathsf{Sym}(\mathcal{N})$-sets are functions that preserve the group-action structure. We can define a notion of *support* for the elements of any nominal $\mathsf{Sym}(\mathcal{N})$-set. Roughly, a support of an element is a set of names that determine the permutation action. A *nominal set* is a $\mathsf{Sym}(\mathcal{N})$-set X for which every element $x \in X$ has a (finite) support. We let **Nom** be the category of nominal sets and equivariant functions.

A basic example is the nominal set of π-calculus terms up-to α-equivalence. The permutation action is given by permuting the free names of terms. A set of names C supports a term in this set if all the free names of the term are in C.

Perhaps the best way to understand nominal sets, and the interpretation of nominal logic in nominal sets, is to investigate the constructions that are available in nominal sets. The most important construction is *name-abstraction*, which models binding and α-equivalence. For each nominal set X we have a set $[\mathcal{N}]X$ of elements of X with names abstracted from them. Indeed, $[\mathcal{N}]X = (\mathcal{N} \times X)/_\sim$, where $\sim$ is a notion of α-equivalence, defined in terms of supports and permutations.

There are other, more straightforward constructions. The set $\mathcal{N}$ of names is a nominal set. Given two nominal sets, it is straightforward to define a group action structure on the product and coproduct (disjoint union) of their underlying sets, and this defines the products and coproducts in **Nom**.

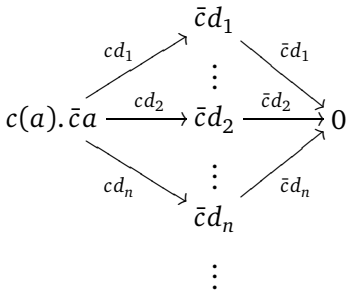
B. Summary of Part I (Operational models)

The standard operational model of a non-deterministic system is the labelled transition system (LTS). A labelled transition system is a set of states, X, together with a transition relation, $(\rightarrow_X) \subseteq X \times L \times X$, describing how the system may move from one state two another, with the labels in L describing what is observed during transitions. Bisimilarity is the finest reasonable equivalence relation on states.

Labelled transition systems provide an appropriate syntax-independent model for simple languages such as Milner's CCS [1989]. There is a labelled transition system whose states are the terms of CCS, and whose transitions describe how one CCS term can evolve into another. Bisimilarity in this LTS is a first notion of equality for CCS, Milner's 'strong equivalence'.

Labelled transition systems are not immediately relevant as models of name-passing process calculi. For a first attempt, one might follow the usual approach for CCS, by considering a labelled transition system whose states are the terms of the π-calculus, and with a transition from one term to another as permitted in the language. For instance, consider the π-calculus term $c(a).\bar{c}a$, which receives a name on channel c and then transmits it again. Taking a naive approach, the reachable LTS for this very simple term has an infinity of states, allowing for the input and retransmission of

any name.



Many of the distinct states are syntactically closely related: the state at the top of the diagram is a renaming of the one directly below it. It is not appropriate, in the name-passing setting, to abstract away from this renaming structure. The information about renaming states is vital in giving a semantics to a language, and plays a crucial role in the definition of bisimilarity. For instance, in the π-calculus, a binary relation R on terms is a *bisimulation* if whenever $P\,R\,Q$ then

- for non-bound-output labels l, if $P \xrightarrow{l} P'$ then there is Q' with $P'\,R\,Q'$ and $Q \xrightarrow{l} Q'$, and
- if $P \xrightarrow{\bar{c}(a)} P'$ *with a not in the free variables of Q*, then there is Q' with $P'\,R\,Q'$ and $Q \xrightarrow{\bar{c}(a)} Q'$,

and similarly with the roles of P and Q interchanged. A bisimulation on π-calculus terms is *wide-open* if it is closed under all substitutions. The greatest wide-open bisimulation, wide-open bisimilarity, is arguably the finest reasonable equivalence for π-calculus terms. All these definitions make use of the free-variable and renaming structure of the set of π-calculus terms.

Questions. With these concerns noted, it is reasonable to ask: *what is an operational model of name-passing?* In the thesis I answer this by developing and relating various approaches.

Part I of the thesis is summarized as follows.

Key result. The following operational models of name-passing are essentially equivalent.

- A class of models involving indexed labelled transition systems and $\mathcal{N}$-LTSs, in the sense proposed by Cattani and Sewell [2004].
- A class of models built from structured coalgebras, following the proposals of Fiore and Turi [2001].
- A class of models of a theory of transition systems in Pitts's [2003] nominal logic.
- The history dependent automata of Montanari and Pistore [2005].

I sketch these different approaches briefly now.

Indexed labelled transition systems. Cattani and Sewell suggested that name-passing *can* be modelled in labelled transition systems, provided the states are indexed by sets of 'known names'. Given a function $C \to D$ between sets of names, there should be a function between sets of states, taking {*states that know C*} to {*states that know D*}. Mathematically speaking, the states of an *indexed labelled transition system* form the set of elements of a presheaf over the category of finite sets.

Once states are indexed in this way, it is possible to axiomatize apparent properties of a name-passing system. For instance, in the π-calculus, there is no bound output transition

$$(\bar{a}a \mid \nu a.\bar{c}a) \xrightarrow{\bar{c}(a)} \bar{a}a$$

intuitively because the name a is free in the left-hand side, and so cannot be a newly generated name. In the context of indexed transition systems, this requirement can be phrased model-theoretically, without reference to the particular syntax of the π-calculus. In the thesis, I investigate various axioms of this kind.

Structured coalgebras. The following observation motivates coalgebras as an abstraction of the notion of transition system: a labelled transition system $(\rightarrow) \subseteq X \times L \times X$ is the same thing as a coalgebra $X \rightarrow \mathscr{P}(L \times X)$ when it is seen as a non-deterministic 'next-state' function. Other model theoretic ideas, including bisimulation, can be defined in this setting. Fiore and Turi [2001] introduced a model for name-passing based on coalgebras for an endofunctor on the category of presheaves. In the thesis, I investigate and develop different endofunctors on different categories of presheaves. An important contribution of this thesis is the new notion of *structured coalgebra*. This is a fundamental general notion that is of particular import when modelling name-passing systems.

Transitions systems in nominal logic. Pitts's nominal logic [2003] provides facilities for expressing requirements about freshness of variables and binding structure. I introduce nominal logic theories of transition systems for name-passing. Nominal logic is a natural language for axiomatising the model-theoretic properties of name-passing systems. For instance, we can axiomatize the property of π-calculus terms mentioned above as follows. Here, the symbol $\#$ is pronounced 'fresh for'.

$$\forall P, P' : \mathsf{X},\ a, c : \mathsf{N}.\ P \xrightarrow{\tilde{c}(a)} P' \implies a \# P$$

The connection between these 'nominal transition systems' and the other approaches based on presheaves arises as follows. An important contribution of the thesis is the introduction of the theory of *nominal substitutions*, with the following result.

Theorem. The category of nominal substitutions is equivalent to a category of sheaves on finite sets.

Indeed, as well as connecting models of name-passing, this theorem provides an elegant connection between two competing approaches to abstract syntax with variable binding: the nominal sets approach of Pitts and Gabbay, and the approach of Fiore, Plotkin, and Turi [1999], which is based on presheaves.

History dependent automata. The final model of name-passing that I consider arises from the work of Montanari and Pistore [2005] on *history dependent automata*. The starting point of history dependent automata is a notion of *named-set*. A history dependent automaton is an automaton that is built of named-sets instead of sets — there is a named-set of states, and a named-set of labels, and so on. History dependent automata are connected with the other models by the following theorem, which makes essential use of the orbit-stabilizer theorem.

Theorem. The category of history dependent automata is equivalent to a category of sheaves on finite sets and injections.

It is pleasing, and perhaps surprising, that history dependent automata, invented as an efficient apparatus for model-checking name-passing systems, have such a tight connection with other, more mathematical structures.

C. Summary of Part II (Structural operational semantics)

The labelled transition system semantics of the π-calculus is given by a rule-based inductive definition. For example, communication in the π-calculus is specified by the following rule.

$$\frac{P \xrightarrow{c(z)} P' \quad Q \xrightarrow{\bar{c}d} Q'}{P \,|\, Q \xrightarrow{\tau} [d/z]P' \,|\, Q'} \qquad (*)$$

In the original work of Milner et al. [1992], the meaning of the rule-based inductive definition is not explained: this is left to the reader's intuitions. Through the 1980s and 1990s, various people initiated research into various kinds of rule-based inductive definitions. They asked: When, and how, do such definitions give rise to valid transition systems? What can be proved about systems defined in this way? Under what conditions is bisimilarity a congruence? Some of the results of these investigations are reported by Aceto et al. [2001]. An important analysis, from the point of view of this thesis, is that of Bloom et al. [1995], and their 'GSOS' format. A rule is in the *positive GSOS format* if it has the following shape,

$$\frac{\bigcup_{i=1}^{n} \left\{ X_i \xrightarrow{a_{ij}} Y_{ij} \;\middle|\; 1 \leq j \leq m_i \right\}}{\mathsf{op}(X_1, \ldots, X_n) \xrightarrow{c} C\left[\vec{X}, \vec{Y}\right]}$$

where all variables are distinct, $n \geq 0$ is the arity of op, $m_i \geq 0$, and $C\left[\vec{X}, \vec{Y}\right]$ is a context with free variables including at most the X's and Y's.

Every specification in the GSOS format defines a transition system, and, moreover, bisimilarity is a congruence. That is, for any n-ary operator op of the language, if $P_1 \sim Q_1, \ldots P_n \sim Q_n$, then $\mathsf{op}(P_1, \ldots, P_n) \sim \mathsf{op}(Q_1, \ldots Q_n)$ (writing $\sim$ for the bisimilarity relation).

Questions. The communication rule $(*)$ for the π-calculus does *not* fit into the GSOS format. There is a binding label in the premise, and a substitution in the conclusion. Other rules for the π-calculus involve side-conditions about the freshness of variables. Moreover, with name-passing calculi, the definition of bisimulation must be modified for it to be reasonable and for bisimilarity to be a congruence. This raises the question: *What is the meaning of a rule-based inductive definition of a name-passing process calculus?* More precisely: *What is a congruence rule format for name-passing process calculi?*

The GSOS format was put in a more categorical light in the 'mathematical structural operational semantics' (MSOS) of Turi and Plotkin [1997]. They explained how the rules in the GSOS format correspond to 'abstract rules' — natural transformations between endofunctors on **Set**. They explained how abstract rules provide recursion data which can be used to lift a monad-of-syntax to a category of coalgebras; this gives the congruence result.

Fiore and Turi [2001] considered name-passing calculi in the setting of mathematical structural operational semantics. They provided a specification of the π-calculus as an abstract rule, *viz*. natural transformation between endofunctors on a presheaf category, and thus obtained a congruence result. An immediate complaint is as follows: *natural transformations are not in the toolbox of a typical operational semanticist*.

The purpose of this Part II of the thesis is to address this complaint. I develop the theory of mathematical structural operational semantics in the setting of structured coalgebras. By careful inspection of the abstract rules in this MSOS, I extract a concrete rule format for rule-based inductive definitions. I call this format *the* Nominal GSOS$^+$ *format*[1]. The analysis is explained in the setting of nominal logic, which, after all, is intended as a reasonable formalization of everyday pen-on-paper proofs.

[1] Here and in the thesis I use the symbol N, introduced by Pitts, to prefix 'nominal' concepts.

Key result. A rule-based inductive definition in the И-GSOS$^+$ format induces an operational model for which bisimilarity is a congruence.

More precisely, I proceed as follows. I introduce a notion of *rule structure*, as the data (such as $(*)$ above) for a rule-based inductive definition for name-passing. A rule structure is used in two ways: (a) a rule structure gives rise to a natural transformation, an abstract rule for mathematical SOS, and (b) a rule structure gives rise to a theory of nominal logic. Thus a connection is made with MSOS on the one hand, and the usual workings of an operational semanticist on the other.

The conditions of the И-GSOS$^+$ format for name-passing are much more elaborate than the conditions for the GSOS format. One can argue that the conditions are necessary by finding examples of rule structures for which bisimilarity is not a congruence. For instance, suppose that we add an operator if-fresh to the π-calculus. It takes one name parameter and one term parameter with a binder, and its semantics are given by the following rule structure.

$$\frac{\mathsf{p} \xrightarrow{\tau} \mathsf{q}}{\mathsf{if\text{-}fresh}(\mathsf{a}, \langle \mathsf{a} \rangle \mathsf{p}) \xrightarrow{\tau} \mathsf{q}}$$

This rule structure is *not* in the И-GSOS$^+$ format, and should indeed be forbidden for the following reason. In this extension of the π-calculus, under the nominal logic semantics, we have the following behaviour. If a is fresh for $\langle b \rangle P$ then the term $\mathsf{if\text{-}fresh}(a, \langle b \rangle P)$ behaves exactly as P with regard to silent actions, because in that case the term $\mathsf{if\text{-}fresh}(a, \langle b \rangle P)$ is α-equivalent to $\mathsf{if\text{-}fresh}(a, \langle a \rangle P)$. But if a is not fresh, then $\mathsf{if\text{-}fresh}(a, \langle b \rangle P)$ cannot reduce. With this property, the operation if-fresh can be used to construct a context that distinguishes two very simple bisimilar terms. Thus the operation if-fresh, if allowed, would break the congruence of bisimilarity.

D. Detailed overview of Part I (Operational models)

I now provide a more detailed overview of Part I of the thesis. We begin with presheaves of states, and move through labelled transition system and coalgebraic models. The section concludes with the more refined kinds of state space: sheaves, nominal substitutions, and named-sets with symmetry.

The material of Part I is published as [Fiore and Staton, 2006].

D.a. Presheaves of states

The category **Set** has as objects, sets, and morphisms, functions between sets. As discussed above, it is excessively simplistic to model a name-passing system with merely a *set* of states. The states have additional structure, *viz*. free names and ways of renaming one state into another. We now develop some categories whose objects are sets with renaming structure, and whose morphisms preserve that structure.

For the remainder of this article we fix an infinite set $\mathcal{N}$ of names. Let the category **I** have as objects finite sets of names, and as morphisms injections between them. An important role is also played by the category **F** with the same objects, but whose morphisms are all functions between them.

A (covariant) presheaf over **I**, *i.e.* a functor $P : \mathbf{I} \to \mathbf{Set}$, is a first notion of set-with-renaming. Given a set C of names, the set $P(C)$ is thought of as the set of all states whose names are included in C. An injection $\iota : C \rightarrowtail D$ has an action, a function $P(\iota) : P(C) \to P(D)$, describing how states with names in C can be renamed to states with names in D. A natural transformation between such presheaves plays the role of a function that preserves the renaming structure. The category $\mathbf{Set}^{\mathbf{I}}$ is introduced in the thesis in Sec. 3.2.

In some situations it is necessary to substitute one name for another, in a non-injective way. A (covariant) presheaf over $\mathbf{F}$, *i.e.* a functor $X : \mathbf{F} \to \mathbf{Set}$, is a first notion of set-with-substitution. Any function $f : C \to D$ has an action, a function $P(f) : X(C) \to X(D)$, describing how to perform the substitution f on states with names in C. The category $\mathbf{Set}^{\mathbf{F}}$ is introduced in the thesis in Sec. 3.4.1.

D.b. Labelled transition systems for name-passing

Sets of states, elements of presheaves. Every presheaf P in $\mathbf{Set}^{\mathbf{I}}$ gives rise to a set $(\int P)$ of elements. An element of P is a pair (C, p), where $p \in P(C)$. In the thesis, I write $(C \vdash p)$ for an element of such an element. The idea is that the element $(C \vdash p)$ represents the state p in name-context C. Similar remarks are also valid for elements of presheaves in $\mathbf{Set}^{\mathbf{F}}$.

Indexed labelled transition systems

One appropriate model of name-passing is a labelled transition system whose states are elements of a presheaf. These I call indexed labelled transition systems (ILTSs).

Definitions 3.3.2 and 3.4.4 (in the thesis)**.** An $\mathbf{I}$-*indexed early labelled transition system* ($\mathbf{I}$-IL$_e$TS) is a presheaf $P \in \mathbf{Set}^{\mathbf{I}}$ together with a transition relation $\longrightarrow \subseteq \int P \times Lab_e \times \int P$. An $\mathbf{F}$-*indexed early labelled transition system* ($\mathbf{F}$-IL$_e$TS) is a presheaf $X \in \mathbf{Set}^{\mathbf{F}}$ together with a transition relation $\longrightarrow \subseteq \int X \times Lab_e \times \int X$.

In this definition, the set Lab_e is the set of labels for the early semantics. These labels are the input labels (written $c?d$), output labels (written $c!d$), and the silent labels τ. An important example of an indexed labelled transition is the semantics of the π-calculus.

Various conditions on $\mathbf{I}$-IL$_e$TSs and $\mathbf{F}$-IL$_e$TSs are appropriate. These are Axioms **I**1–**I**6 and **F**2', listed in Figures 3.4 and 3.5 in the thesis, reproduced here on page 9. The conditions were suggested by three things: (i) intuitions and existing lemmas about the transition systems for the π-calculus (as studied in Section 3.1 of the thesis); (ii) the aim of a bijective correspondence with the coalgebraic notion of model, discussed in the following section; (iii) the existing axiomatization of Cattani and Sewell [2004], who introduce $\mathcal{N}$-LTSs as $\mathbf{F}$-IL$_e$TSs that satisfy certain conditions.

Ground semantics. The input labels considered in the definition above are of the form $c?d$, to be read "input name d on the channel named c". These kinds of input label describe the *early* input semantics. An alternative semantics of input is what I call the *ground* semantics. (Other authors use the term *late*; the terminology 'ground' is motivated by the natural notion of bisimulation for this semantics.) A ground input label, $c?(d)$, is to be read "input a name on the channel named c and bind it to d", or alternatively, "input a fresh name d on the channel named c". I refer to those transition systems that use the ground, rather than early, semantics, as $\mathbf{I}$-IL$_g$TSs and $\mathbf{F}$-IL$_g$TSs. Ground transition systems are introduced in Section 3.3.4 of the thesis.

A crucial observation is that an $\mathbf{F}$-IL$_g$TS, with ground labels, induces an $\mathbf{F}$-IL$_e$TS, with early labels. The idea is, if you know how to input a fresh name, then you can derive how to input an arbitrary name, by substituting the arbitrary name for the fresh name. The construction is described in Equation 3.4.10 of the thesis.

Theorems 3.4.12 and 3.4.16. The following data are equivalent.

- An $\mathbf{F}$-IL$_e$TS satisfying Axioms **I**1–**I**6 and **F**2' (see the figure on page 9).
- An $\mathbf{F}$-IL$_e$TS satisfying Axioms **I**1–**I**6, that is induced by an $\mathbf{F}$-IL$_g$TS.
- An $\mathcal{N}$-LTS in the sense of Cattani and Sewell. □

Conditions for **I**-IL$_e$TS*s and* **F**-IL$_e$TS*s:*

I1. Channel is known and at most transmitted data is learnt:

$$C \vdash p \xrightarrow{\ell} C' \vdash p' \implies \mathsf{ch}(\ell) \subseteq C \ \wedge\ C' = C \cup \mathsf{dat}(\ell)$$

I2. If one name can be input, then so can any other: for all $d' \in \mathcal{N}$:

$$C \vdash p \xrightarrow{c?d} C \cup \{d\} \vdash p'$$
$$\implies \exists p'' \in P(C \cup \{d'\}).\quad C \vdash p \xrightarrow{c?d'} C \cup \{d'\} \vdash p''$$

I3. Bijective maps preserve transitions.

I4a. Knowing/forgetting input data preserves transitions.

I4b. Known output data must really be known.

I5. Inclusion maps preserve transitions.

I6. Inclusion maps reflect transitions.

Additional condition for **F**-IL$_e$TS*s:*

F2'. (For an **F**-IL$_e$TS.) Input is determined by the input of fresh names:

$$C \vdash x \xrightarrow{c?d} C \cup \{d\} \vdash x'$$
$$\iff \exists z \in (\mathcal{N} - C),\ x'' \in X(C \cup \{z\}).\ [d/z]x'' = x' \ \wedge\ C \vdash x \xrightarrow{c?z} C \cup \{z\} \vdash x''$$

Conditions on **I**-IL$_e$TSs and **F**-IL$_e$TSs. Reproduced from Figures 3.4 and 3.5 in the thesis.

Bisimulation for indexed labelled transition systems

Indexed labelled transition systems provide an appropriate setting for model-theoretic definitions of the kinds of bisimulation that are relevant for name-passing systems.

Definitions 3.3.4 and 3.4.4 (Paraphrased)**.** An **I**-*indexed binary relation* R between presheaves P and Q in $\mathbf{Set}^{\mathbf{I}}$ is a presheaf R in $\mathbf{Set}^{\mathbf{I}}$ that is a subobject of $P \times Q$. Similarly, an **F**-*indexed binary relation* R between presheaves X and Y in $\mathbf{Set}^{\mathbf{F}}$ is a presheaf R in $\mathbf{Set}^{\mathbf{F}}$ that is a subobject of $X \times Y$.

An **I**-*indexed early bisimulation* between two **I**-IL$_e$TSs, $(P, \xrightarrow[P]{})$ and $(Q, \xrightarrow[Q]{})$. is an **I**-indexed binary relation $R \subseteq P \times Q$ such that

$$\forall C \subseteq_{\mathrm{f}} \mathcal{N},\ (p,q) \in R(C),\ \ell \in Lab_{\mathrm{e}},\ (C' \vdash p') \in \textstyle\int P.$$
$$C \vdash p \xrightarrow[P]{\ell} C' \vdash p' \implies \exists q' \in Q(C').\ C \vdash q \xrightarrow[Q]{\ell} C' \vdash q' \text{and } (p', q') \in R(C')$$

and symmetrically (interchanging the roles of P and Q). An **F**-indexed early bisimulation is an **F**-indexed binary relation that is also an **I**-indexed early bisimulation.

For the **I**-IL$_e$TS for the π-calculus, an **I**-indexed early bisimulation is essentially an early bisimulation that is closed under injective substitutions, while an **F**-indexed early bisimulation is a wide-open bisimulation: it is an early bisimulation that is closed under all substitutions.

D.c. Coalgebras in general

Indexed labelled transition systems are one kind of operational model of name-passing. Another kind of model that is developed in the thesis is a coalgebraic model. Before discussing coalgebras for name-passing, there are some comments to be made about coalgebras in general.

Definition 2.1.2. Consider an endofunctor B on a category $\mathscr{C}$. A *B-coalgebra* is an object $X \in \mathscr{C}$ equipped with a morphism $X \to BX$.

Coalgebras have thus arisen as general notions of transition system in various areas of theoretical computer science [see *e.g.* Rutten, 2000]. In Sec 2.2 of the thesis, I provide a simple first example of how coalgebras represent a refined kind of transition system, by looking at the early semantics of value-passing systems. The reader may enjoy reading this as a prelude to the more elaborate refinements for name-passing systems.

I now summarize some important developments of the general theory of coalgebras that are introduced in the thesis.

Structured coalgebras

For various reasons studied in the thesis, coalgebras per se are too constraining. The thesis introduces the following new and more general notion of *structured coalgebra*.

Definition 2.4.2. Consider a functor $U : \mathscr{D} \to \mathscr{C}$, and let B be an endofunctor on $\mathscr{C}$. A *U-structured B-coalgebra* is an object X of $\mathscr{D}$ together with a morphism $UX \to BUX$ in $\mathscr{C}$ — *i.e.* a B-coalgebra structure for UX.

The intuition should be: "structured coalgebras are coalgebras whose state spaces have additional structure, which need not be preserved by the transition function."

When the structure functor $U : \mathscr{D} \to \mathscr{C}$ has a right adjoint $R : \mathscr{C} \to \mathscr{D}$, then U-structured B-coalgebras are the same things as (RBU)-coalgebras. For the case of name-passing systems, there typically *is* such a right adjoint. This right adjoint, however, is very clumsy to describe, and so the use of structured coalgebras adds an important degree of clarity to the development.

Many of the general results in the thesis are relevant to structured coalgebras — with or without a right adjoint. One important scenario, where there is no right adjoint, arises in the semantics systems involving term-for-variable substitution, such as process-passing calculi. This is mentioned briefly in Sec. 9.3.6 of the thesis and is the subject of a manuscript in preparation.

Coalgebraic bisimulation

The concept of *bisimulation* is relevant at the general level of structured coalgebras. We extend a notion due to Aczel and Mendler [1989], to the setting of structured coalgebras.

Definition 2.4.4 (Paraphrased)**.** Consider a functor $U : \mathscr{D} \to \mathscr{C}$, and let B be an endofunctor on $\mathscr{C}$. A *U-structured B-bisimulation* between U-structured B-coalgebras, (X, h) and (Y, k), is a span $X \leftarrow R \rightarrow Y$ in $\mathscr{D}$ for which there is a compatible U-structured B-coalgebra structure on R.

For labelled transition systems, bisimilarity, *viz.* the greatest bisimulation, is the finest reasonable equivalence. The existence of bisimilarity for coalgebras is studied in Section 5.2. The following result appears to be novel: the novelty is that it does not require the existence of a final B-coalgebra. The result is stated here for the non-structured case.

Corollary (of Props. 5.2.2 and 5.2.3)**.** Let $\mathscr{C}$ be a complete, well-powered category, and let B be a weak-pullback-preserving endofunctor on $\mathscr{C}$. There is a greatest bisimulation between every pair of B-coalgebras. It is the greatest post-fixed point for a monotone operator on the preorder of relations. □

For a given endofunctor B on a category $\mathscr{C}$, the *terminal sequence* is a well-known ordinal-indexed sequence of objects from $\mathscr{C}$ with the property that, if it converges, it converges at the final object in the category of B-coalgebras. In Sec. 5.2.2, a 'relation refinement sequence' is given, and, in Prop. 5.2.5, a tight connection is made between this relation refinement sequence and the terminal sequence.

Relating categories of coalgebras and notions of bisimulation

For every endofunctor B on a category $\mathscr{C}$, there is a category of B-coalgebras. This motivates the following idea, developed in Sec. 2.3 of the thesis. Consider a (2-)category Endo whose objects are pairs $(\mathscr{C}, B)$ of a category and an endofunctor on it. The construction of categories of coalgebras defines a (2-)functor Endo $\to$ **CAT** into the category of categories. Morphisms in Endo induce functors between categories of coalgebras. In Sec. 2.5.2 of the thesis I investigate how these functors be used to compare bisimilarity in the different categories of coalgebras. Theorem 2.5.7 is a little technical but very general; for instance, we have the following corollary.

Corollary (of Thm. 2.5.7)**.** Let B be an endofunctor on a category $\mathscr{C}$, and let B' be a split subfunctor of B. The greatest B'-bisimulation between two B'-coalgebras is also the greatest B-bisimulation between them, when they are considered as B-coalgebras. □

D.d. Coalgebras for name-passing

Early semantics. The labelled transition systems for name-passing have *presheaves* of states instead of sets of states. For a coalgebraic perspective, we consider the following endofunctor on the presheaf category $\mathbf{Set}^{\mathbf{I}}$. This endofunctor was first suggested by Fiore and Turi [2001].

$$\begin{array}{rlll} B_{\mathrm{e}}(-) = & \mathsf{inp}: & [N \rightleftharpoons [N \Rightarrow \mathscr{P}_{\mathrm{ne}}(-)]] & \textit{Input} \\ \times & \mathsf{out}: & [(N \times N) \rightleftharpoons \mathscr{P}_{\mathrm{ne}}(-)] & \textit{Free output} \\ \times & \mathsf{bout}: & [N \rightleftharpoons \delta(\mathscr{P}_{\mathrm{ne}}(-))] & \textit{Bound output} \\ \times & \mathsf{tau}: & [1 \rightleftharpoons \mathscr{P}_{\mathrm{ne}}(-)] & \textit{Silent}. \end{array} \qquad \text{(thesis 3.2.11)}$$

The constructions involved in this endofunctor are defined in Sec. 3.2.1 of the thesis, but in brief: $\mathscr{N}$ is a distinguished presheaf of names; $[(-) \rightleftharpoons (-)]$ is a partial function space; $[(-) \Rightarrow (-)]$ is a full function space; $\delta(-)$ is a name generation operator; and $\mathscr{P}_{\mathrm{ne}}(-)$ is a (pointwise) non-empty powerset operator.

A B_{e}-coalgebra is given by a presheaf $P \in \mathbf{Set}^{\mathbf{I}}$ together with a natural family of functions

$$\left\{ h_C : P(C) \to (B_{\mathrm{e}}P)(C) \right\}_{C \in \mathbf{I}}$$

To every finite set C of names and every state $p \in P(C)$, the coalgebra assigns a 4-tuple $h_C(p)$, the components of which describe the input, output, bound output and silent functionality of the state p. More intuition is given in Sec. 3.2.2 of the thesis.

Coalgebras with arbitrary substitutions. Recall that an **F**-IL$_{\mathrm{e}}$TS is an **I**-IL$_{\mathrm{e}}$TS whose set of states comes from a presheaf over **F**. With this, and the above corollary, in mind, we write $U^{\mathbf{I}}_{\mathbf{F}} : \mathbf{Set}^{\mathbf{F}} \to \mathbf{Set}^{\mathbf{I}}$ for the evident forgetful functor and consider a model based on $U^{\mathbf{I}}_{\mathbf{F}}$-structured B_{e}-coalgebras.

A *ground* semantics is particularly interesting for transition systems supporting arbitrary substitutions. An endofunctor on $\mathbf{Set}^{\mathbf{I}}$ for ground semantics is defined analogously to equation (3.2.11) above, by modifying the first line, so that only fresh input data is considered.

$$\begin{array}{rlll} B_{\mathrm{g}}(-) = & \mathsf{binp}: & [N \rightleftharpoons \delta(\mathscr{P}_{\mathrm{ne}}(-))] & \textit{Bound input} \\ \times & \ldots & & \end{array} \qquad \text{(thesis 3.2.13)}$$

There is an alternative definition of this endofunctor, through the isomorphism $B_g(-) \cong \mathscr{P}(L_g(-))$ where

$$\begin{array}{rlll} L_g(-) = & \mathsf{binp}: & N \times \delta(-) & \textit{Bound input} \\ + & \mathsf{out}: & N \times N \times (-) & \textit{Free output} \\ + & \mathsf{bout}: & N \times \delta(-) & \textit{Bound output} \\ + & \mathsf{tau}: & (-) & \textit{Silent}. \end{array} \qquad \text{(thesis 3.2.15)}$$

Corollary (of Thms. 3.3.7 and 3.3.8)**.** There is a bijective correspondence between B_e-coalgebras and **I**-IL$_e$TSs satisfying Axioms **I**1–**I**6 of Figure 3.4 (see page 9).

Corollaries 3.4.14 and 3.4.17. The following data are equivalent.

1. A $U^{\mathbf{I}}_{\mathbf{F}}$-structured B_g-coalgebra.
2. An **F**-IL$_e$TSs satisfying Axioms **I**1–**I**6 and **F**2'.
3. An $\mathcal{N}$-LTS in the sense of Cattani and Sewell. □

Coalgebraic bisimulation for name-passing

Since we have correspondences between coalgebras and indexed labelled transition systems, it would be reasonable to expect a correspondence between (structured) bisimulation and indexed bisimulation for ILTSs. In fact, this correspondence is more complicated than might be expected (and more complicated than suggested in [Fiore and Turi, 2001]). In Sec. 3.3.3. of the thesis, I investigate an anomaly that breaks the correspondence. The problem is resolved by introducing a closure operator on indexed bisimulations on ILTSs: the closure of an indexed bisimulation is a coalgebraic bisimulation. We thus have the following result.

Corollary 3.3.17 and Prop. 3.4.8. The final B_e-bisimulation between two B_e-coalgebras is the greatest **I**-indexed early bisimulation between the corresponding **I**-IL$_e$TSs. The final $U^{\mathbf{I}}_{\mathbf{F}}$-structured B_e-bisimulation between two $U^{\mathbf{I}}_{\mathbf{F}}$-structured B_e-coalgebras is the greatest **F**-indexed early bisimulation between the corresponding **F**-IL$_e$TSs. □

For wide-open relations, the requirements of ground and early bisimulation coincide. This is easily established at the coalgebraic level, by exhibiting one endofunctor as a split subfunctor of the other. This is Thm. 3.4.9 in the thesis.

D.e. Refining the state spaces: sheaves, nominal substitutions, and named-sets

To conclude our discussion of operational models of name-passing systems, I discuss some refinements that it is appropriate to make to the notion of state space.

Sheaves of states. There are some presheaves in $\mathbf{Set}^{\mathbf{I}}$ and $\mathbf{Set}^{\mathbf{F}}$ that have properties that would not be expected of a 'set-with-renaming'. An example is the state space considered in the 'anomaly' for the correspondence of bisimulations, mentioned above. Firstly, the action of a presheaf P in $\mathbf{Set}^{\mathbf{I}}$ need not be injective. From the intuition that we have given, this is obscure: the action of an injective renaming should surely be injective. Another concern is that, for $C \in \mathbf{I}$, and $D \subseteq C$, we can deduce that the 'free names' of an element p of $P(C)$ are contained in D, by looking at the action of injections on p. In this case, we say that D *supports* p. Our intuition would suggest that p ought to also reside in $P(D)$, but there are some presheaves in $\mathbf{Set}^{\mathbf{I}}$ for which this is not the case.

The condition about supports is exactly a *sheaf condition* for a particular Grothendieck coverage on **I**. The category of such sheaves, $\mathbf{Sh}(\mathbf{I})$, is known as the *Schanuel topos*. General information about sheaves is provided in Sec. 4.1.1 of the thesis, and the Schanuel topos is first discussed in Sec 4.2.1.

There are similar complaints to be made about presheaves in $\mathbf{Set}^{\mathbf{F}}$. Moreover, the forgetful functor $\mathbf{Set}^{\mathbf{F}} \to \mathbf{Set}^{\mathbf{I}}$ between presheaf categories, which is important at various stages in the development, does not factor through the sheaf category $\mathbf{Sh}(\mathbf{I})$. For these reasons we also consider a sheaf subcategory of $\mathbf{Set}^{\mathbf{F}}$. (see Sec. 4.3 of the thesis).

In Thm. 4.2.6 of the thesis, I show that all the constructions on $\mathbf{Set}^{\mathbf{I}}$, that are used to define endofunctors B_e and B_g, restrict to constructions on the sheaf category $\mathbf{Sh}(\mathbf{I})$.

Corollary (to Thm. 4.2.6)**.** The endofunctors B_e and B_g on $\mathbf{Set}^{\mathbf{I}}$ restrict to endofunctors B_e^{Sh} and B_g^{Sh} on $\mathbf{Sh}(\mathbf{I})$. □

It is important to note that the problems with bisimulation for coalgebras on presheaves, discussed in Sec. 3.3.3 of the thesis, do not arise for sheaves.

Theorem 4.2.5. For B_e-coalgebras with sheaf carriers, a B_e-bisimulation is the same thing as a $\mathbf{I}$-indexed early bisimulation on the induced $\mathbf{I}$-IL$_e$TSs. □

The Schanuel topos as a Kleisli category. An alternative description of the Schanuel topos is as the Kleisli category for a monad on the presheaf category $\mathbf{Set}^{\mathbf{B}}$ (see Sec. 4.4). Here, $\mathbf{B}$ is the category of finite sets of names and bijections between them. The intuition is as follows: a presheaf Q in $\mathbf{Set}^{\mathbf{B}}$ determines, for each set C of names, the set $Q(C)$ of all states involving precisely the names in C. (Recall that, by contrast, for a presheaf P in $\mathbf{Set}^{\mathbf{I}}$, the set $P(C)$ is thought of as all the states whose names are contained in C.)

The tighter, more explicit specification of names in $\mathbf{Set}^{\mathbf{B}}$ allows a simplified axiomatization of transition systems. I define an $\mathbf{B}$-IL$_e$TS to be a a labelled transition system whose states are the elements of a presheaf in $\mathbf{Set}^{\mathbf{B}}$. I introduce simplified axioms Axiom $\mathbf{B}1$– $\mathbf{B}3$ on $\mathbf{B}$-IL$_e$TSs in Sec. 4.4.2.

Nominal sets and nominal substitutions

The nominal sets of Pitts and Gabbay were recalled earlier in this report and are discussed in Sec. 7.1 of the thesis. It is well-known that the category of nominal sets is equivalent to the Schanuel topos.

In the thesis, in Sec. 3.3, I introduce a nominal logic theory of *nominal substitutions* as a 'nominal' formulation of sheaves in $\mathbf{Sh}(\mathbf{F})$. The theory has one sort, X, and a single function symbol $\mathsf{sub} : \mathsf{N}, [\mathsf{N}]\mathsf{X} \to \mathsf{X}$. There are four axioms. (Here, $[b/a]x$ is shorthand for $\mathsf{sub}(b, \langle a \rangle x)$.)

NOMSUB-1. $\forall a : \mathsf{N}.\ \forall x : \mathsf{X}.\ [a/a]x = x.$

NOMSUB-2. $\forall a, b : \mathsf{N}.\ \forall x : \mathsf{X}.\ a \# x \implies [b/a]x = x.$

NOMSUB-3. $\forall a, b, c : \mathsf{N}.\ \forall x : \mathsf{X}.\ [c/b][b/a]x = [c/b][c/a]x.$

NOMSUB-4. $\forall a, b, c, d : \mathsf{N}.\ \forall x : \mathsf{X}.\ c \neq b \neq a \neq d \implies [d/b][c/a]x = [c/a][d/b]x.$

Let **NomSub** be the category of models of this theory (in nominal sets). The following correspondence result is non-trivial.

Theorem 7.3.2. The category **NomSub** is equivalent to $\mathbf{Sh}(\mathbf{F})$. □

Given this correspondence, it is reasonable to rework the coalgebraic models for name-passing in the context of nominal sets, by reinterpreting the endofunctor for ground behaviour (thesis eq. 3.2.15) in this setting. The only obstacle is an explicit description of the pointwise powerset on presheaves, which has not been considered in the nominal context before. This is treated in Sec. 7.1.5 of the thesis, where it is characterized as a set of 'support-bounded' subsets (Prop. 7.1.6).

It is also instructive to re-investigate the work on labelled transition systems in this setting. In Sec. 7.5 of the thesis I define nominal logic theories of labelled transition systems.

Definition 7.5.5. The nominal logic theory of *nominal ground labelled transition systems (*$\mathit{И}_{\mathrm{g}}$-LTS*s)* has one ground sort X and four relation symbols: a *bound input transition* relation symbol $(\overset{-?(-)}{\longrightarrow})$ with arity $\mathsf{X},\mathsf{N},\mathsf{N},\mathsf{X}$; an *output transition* relation symbol $(\overset{-!-}{\longrightarrow})$ with arity $\mathsf{X},\mathsf{N},\mathsf{N},\mathsf{X}$; a *bound output transition* relation symbol $(\overset{-!(-)}{\longrightarrow})$ with arity $\mathsf{X},\mathsf{N},\mathsf{N},\mathsf{X}$; and a *silent transition* relation symbol $(\overset{\tau}{\longrightarrow})$ with arity X,X; subject to Axioms $\mathit{И}_{\mathrm{g}}1$ and $\mathit{И}_{\mathrm{g}}2$ in Fig. 7.2.

An analogous definition is given for nominal early labelled transition systems in Defn. 7.5.1 of the thesis, with different axioms ($\mathit{И}_{\mathrm{e}}1$–$\mathit{И}_{\mathrm{e}}4$, in Fig. 1, omitted from this report).

$\mathit{И}_{\mathrm{g}}1$. The channel and free data are known, while binding data is fresh.

$$\forall x,y:\mathsf{X},\ c,d:\mathsf{N}.\qquad \left(x \overset{c?(d)}{\longrightarrow} y \implies \neg(c\#x)\wedge(d\#x)\right)$$
$$\wedge\ \left(x \overset{c!d}{\longrightarrow} y \implies \neg(c\#x)\wedge\neg(d\#x)\right)$$
$$\wedge\ \left(x \overset{c!(d)}{\longrightarrow} y \implies \neg(c\#x)\wedge(d\#x)\right)$$

$\mathit{И}_{\mathrm{g}}2$. Names in the derivative depend only on names in the source and communication data.

Figure 7.2: Axioms for the nominal logic theory of nominal ground labelled transition systems.

Corollary (of Thms 4.4.9 and Thm 7.5.3)**.** The following data are equivalent.

- A $B_{\mathrm{e}}^{\mathrm{Sh}}$-coalgebra.
- An **I**-IL$_{\mathrm{e}}$TS with sheaf carrier, satisfying Axioms **I**1–**I**6 (see page 9).
- An **B**-IL$_{\mathrm{e}}$TS satisfying Axioms **B**1–**B**3.
- A $\mathit{И}_{\mathrm{e}}$-LTS satisfying Axioms $\mathit{И}_{\mathrm{e}}1$–$\mathit{И}_{\mathrm{e}}3$. □

Corollary (of Prop. 7.5.4 and Thm. 7.5.6)**.** The following data are equivalent.

- A $U_{\mathbf{F}}^{\mathbf{I}}$-structured B_{g}^{Sh}-coalgebra.
- An **F**-IL$_{\mathrm{e}}$TS with sheaf carrier, satisfying Axioms **I**1–**I**6 and **F**2'.
- An $\mathcal{N}$-LTS, in the sense of Cattani and Sewell, with sheaf carrier.
- A $\mathit{И}_{\mathrm{e}}$-LTS over a nominal substitution satisfying Axioms $\mathit{И}_{\mathrm{e}}1$–$\mathit{И}_{\mathrm{e}}4$.
- A $\mathit{И}_{\mathrm{g}}$-LTS over a nominal substitution satisfying Axioms $\mathit{И}_{\mathrm{g}}1$–$\mathit{И}_{\mathrm{g}}2$. □

Efficient descriptions of sheaves: named sets with symmetries. Various authors, including Montanari and Pistore [2005], have proposed varieties of *named-sets* as models of name-passing calculi. So-called *history dependent automata*, *viz.* automata internal to categories of named-sets, have been used to provide efficient verification techniques for name-passing systems.

In the thesis, Sec. 5.1, I establish an equivalence between a category of *named-sets with symmetries*, and the Schanuel topos. For our purposes, a named-set-with-symmetries is a tuple

$$\left(I,\{m_i\}_{i\in I},\{H_i\}_{i\in I}\right)$$

where I is a set, and for all $i \in I$, m_i is a natural number and H_i is a group of permutations on m_i. The group H_i can be thought of as "the permutations of m_i that fix i". Morphisms between named-sets must be defined carefully, and the correct definition is quite elaborate.

Theorem 5.1.8. The category of named-sets with symmetries is equivalent to the Schanuel topos. □

The categories of presheaves, sheaves, and nominal sets and substitutions, considered in this thesis, are not amenable to machine implementation. For a presheaf P in $\mathbf{Set}^{\mathbf{I}}$, the set $\int P$ of elements is almost always infinite; nominal sets always have infinite carriers unless they have trivial action. The importance of named-sets-with-symmetry is that they allow finite presentations of interesting state spaces.

Corollary 5.1.12 (Simplified)**.** A named-set is finitely presentable (in the categorical sense) if and only if it has a finite carrier set. □

E. Detailed overview of Part II (Structural operational semantics)

In Part II of the thesis I study structural operational semantics of name-passing calculi. I now overview this part in two stages. First, I explain how the mathematical structural operational semantics of Turi and Plotkin [1997] can be used for structured coalgebras in general, with particular emphasis on the first-order setting that is relevant for simple languages such as Milner's CCS [1989]. Secondly, I explain how the theory can be applied to the name-passing case, and how a congruence rule format for name-passing can be extracted.

The material of this section is published as [Fiore and Staton, 2006, 2007].

E.a. First-order framework

My outline here follows the development of Chapter 6 of the thesis. Mathematical structural operational semantics can be explained by the following recipe.

MSOS-1. An algebraic signature $\mathbb{S}$ induces a monad $\mathbf{T}_{\mathbb{S},\mathbf{Set}}$ on the category of sets, whose algebras are the algebras for the signature. The elements of the set $T_{\mathbb{S},\mathbf{Set}}(\emptyset)$ are terms of the language specified by $\mathbb{S}$.

MSOS-2. Given a lifting $\tilde{\mathbf{T}}$ of the monad $T_{\mathbb{S},\mathbf{Set}}$ to a category of coalgebras, the initial $\tilde{\mathbf{T}}$-algebra is a coalgebra whose carrier is the set $T_{\mathbb{S},\mathbf{Set}}(\emptyset)$ of $\mathbb{S}$-terms. For this coalgebra, bisimilarity is a congruence.

MSOS-3. The universal property of a free monad on a signature can be used to define a monad lifting for it, through a (parameterized) recursion theorem.

MSOS-4. Recursion data of this kind can be seen as a rule-based inductive definition of the semantics of the language described by $\mathbb{S}$, in the GSOS format.

MSOS-1: Syntax through initial algebras and free monads

In Sec. 6.1 of the thesis, I consider algebras for signatures at three levels of abstraction. These are, beginning with the most abstract:

- Algebras for an arbitrary monad on an arbitrary category, in the sense of Eilenberg and Moore;

- Algebras for an endofunctor Σ on an arbitrary category, or equivalently algebras for the free monad $\mathbf{T}_\Sigma$ on Σ (provided it exists);
- Algebras for a signature $\mathbb{S}$, in a category $\mathscr{C}$ with finite limits, or equivalently algebras for the endofunctor $\Sigma_{\mathbb{S},\mathscr{C}}$ generated by the signature (provide $\mathscr{C}$ has enough sums), or equivalently algebras for the free monad $\mathbf{T}_{\mathbb{S},\mathscr{C}}$ generated by that endofunctor (when it exists).

The notion of *congruence* is traditionally defined only for the last, lowest level of abstraction, but can also be defined for the other levels.

MSOS-2: Monad liftings and congruence of bisimilarity

A *lifting* of a monad $\mathbf{T}$ on a category $\mathscr{C}$ along a functor $F : \mathscr{C}' \to \mathscr{C}$ is a monad $\mathbf{T}'$ on $\mathscr{C}'$ for which there is an isomorphism $FT' \cong TF$. The lifting is *strict* if this isomorphism is identity. A key idea is that *monad liftings describe good operational semantics*, as the following theorem corroborates.

Corollary 6.2.3. Let $U : \mathscr{D} \to \mathscr{C}$ be a functor between categories, and let B be an endofunctor on $\mathscr{C}$. Let $\tilde{\mathbf{T}}$ be a monad on (U,B)-Coalg which is a strict lifting of a monad $\mathbf{T}$ on $\mathscr{D}$. Let (X,h), (Y,k) be U-structured B-coalgebras. Every final U-structured B-bisimulation between $\tilde{T}(X,h)$ and $\tilde{T}(Y,k)$ is a $\mathbf{T}$-congruence between the free $\mathbf{T}$-algebra on X and the free $\mathbf{T}$-algebra on Y. □

MSOS-3: Monad liftings from parameterized recursion

The result of Corollary 6.2.3 suggests that a monad lifting is a kind of "good operational semantics". We might ask how such a monad lifting should be defined.

When a monad is free for an endofunctor, the universality provides a recursion principle for it. A particular type of recursion data, called an "abstract rule", gives rise to a lifting of the monad to a category of structured coalgebras.

Definition 6.2.11. Let $U : \mathscr{D} \to \mathscr{C}$ be a functor between categories, and let B and Σ be endofunctors on $\mathscr{C}$. Suppose that the free monad $\mathbf{T}$ on Σ exists, and that $\mathbf{T}$ lifts along U to a monad $\tilde{\mathbf{T}}$ on $\mathscr{D}$. An *abstract rule for* $(\mathscr{C},\mathscr{D},U,B,\Sigma,\tilde{\mathbf{T}})$ is a natural transformation

$$\rho : \Sigma(U \times BU) \to BU\tilde{T} \quad .$$

An abstract rule gives rise, via a recursion theorem (introduced in Sec. 6.2.2), to an operator T_ρ on structured coalgebras.

Theorem 6.2.14. The operator T_ρ defines a strict lifting of the monad $\tilde{\mathbf{T}}$ along the forgetful functor from the category of U-structured B-coalgebras. □

Section 6.2.4 of the thesis is dedicated to the following aside, which, in some interesting cases, reduces Thm. 6.2.14 to the problem considered by Turi and Plotkin. Recall that in many cases, including the cases of interest in name-passing, the structure functor $U : \mathscr{D} \to \mathscr{C}$ has a right adjoint, $R : \mathscr{C} \to \mathscr{D}$. In this setting, there is a bijective correspondence between U-structured B-coalgebras and (RBU)-coalgebras.

Suppose, moreover, that Σ lifts along U to an endofunctor $\tilde{\Sigma}$ on $\mathscr{D}$, and that the monad $\tilde{\mathbf{T}}$ on $\mathscr{D}$ is free on $\tilde{\Sigma}$. In this setting, an abstract rule ρ for $(\mathscr{C},\mathscr{D},U,B,\Sigma,\tilde{\mathbf{T}})$ gives rise to an abstract rule $\bar{\rho}$ for $(\mathscr{D},\mathscr{D},\mathrm{id}_\mathscr{D},RBU,\tilde{\Sigma},\tilde{\mathbf{T}})$. We have the following correspondence result.

Theorem 6.2.20. The isomorphism of categories (U,B)-Coalg $\cong$ RBU-Coalg lifts to an isomorphism $((U,B)\text{-Coalg},\mathbf{T}_\rho) \cong (RBU\text{-Coalg},\mathbf{T}_{\bar{\rho}})$ of monads. □

MSOS-4: Positive GSOS

The GSOS format is a syntactic constraint on rule-based inductive definitions, introduced by Bloom et al. [1995]. For systems specified in the GSOS format, bisimilarity is a congruence. (The reader familiar with [Bloom et al., 1995] should be aware that in the thesis we only consider 'positive' GSOS rules, *i.e.* rules without negative premises.)

A crucial observation of Turi and Plotkin [1997] is that abstract rules for

$$(\mathbf{Set}, \mathbf{Set}, \mathsf{id}, \mathscr{P}(L \times -), \Sigma_{\mathbb{S},\mathbf{Set}}, \mathbf{T}_{\mathbb{S},\mathbf{Set}})$$

in the sense of Defn. 6.2.11 above, correspond to rule-based inductive definitions in the GSOS format. In Sec. 6.3 I derive the positive GSOS rule format from the shape of the abstract rules. This section serves as prelude to the more elaborate constructions needed in the name-passing case.

E.b. GSOS rules for name-passing

Chapter 8 of the thesis is dedicated to introducing a GSOS-like rule format for name-passing. To explain the name-passing case, I revise the process MSOS-1–4 explained on page 15.

И-MSOS-1. A nominal algebraic signature $\mathbb{S}$ induces a monad $\mathbf{T}_{\mathbb{S},\mathbf{NomSub}}$ on the category of nominal substitutions. The elements of the set $T_{\mathbb{S},\mathbf{NomSub}}(\emptyset)$ are terms of the language specified by $\mathbb{S}$, up-to α-equivalence, and with the natural substitution structure.

И-MSOS-2. Given a lifting $\tilde{\mathbf{T}}$ of the monad $\mathbf{T}_{\mathbb{S},\mathbf{Nom}}$ to the category of structured coalgebras, the initial $\tilde{\mathbf{T}}$-algebra is a structured coalgebra whose carrier is the set $T_{\mathbb{S},\mathbf{NomSub}}(\emptyset)$ of $\mathbb{S}$-terms up-to α-equivalence. For this structured coalgebra, wide-open bisimilarity is a congruence.

И-MSOS-3. The universal property of a free monad on a signature can be used to define a monad lifting for it, through a (parameterized) recursion theorem.

И-MSOS-4. Recursion data of this kind can be seen as a rule-based inductive definition of the semantics of the language described by $\mathbb{S}$, in a new format called the И-GSOS$^+$ format.

Of these steps, the congruence theorems (И-MSOS-2) and recursion principles (-3) are established in full generality in Ch. 6. Only И-MSOS-1 and -4 require significant further development in Ch. 8.

И-MSOS-1: Nominal algebraic signatures

In the π-calculus, we have operators such as parallel composition, which takes two processes as arguments. But there are also operators such as input prefix

$$\mathsf{inp}(c, \langle d \rangle P) \qquad \text{(usually written } c(d).P\text{, with 'syntactic sugar')}$$

meaning "input a name on channel c, and bind it to d in P". This operator takes a name parameter, c, and a process parameter, P, with one name, d, bound in it.

Definition 7.4.1. A *nominal algebraic signature* is given by a collection of operators, where each operator is associated with: an arity of names (a natural number); an arity of terms; and, for each term parameter, a number describing the binding depth.

For a classical algebraic signature, we can speak of models in any category with finite products. Models for nominal algebraic signatures can be considered in any category with finite products, a distinguished object of names, and an endofunctor describing binding. We call such categories *model categories*. Three such categories are particularly relevant:

- One model category is the category of sets $\mathbf{Set}_{\mathtt{N}}$ with a chosen set of name metavariables $\mathtt{N}$, and with binding given by $\mathtt{N} \times (-)$. In this setting, the free algebra $T_{\mathbb{S},\mathbf{Set}_{\mathtt{N}}}(\emptyset)$ for a nominal algebraic signature $\mathbb{S}$ is the set of *raw* terms, with names from $\mathtt{N}$, but without α-equivalence.

- Another model category is the category of nominal sets **Nom**, with the particular set $\mathscr{N}$ of names and the binding functor there. In this setting, the free algebra $T_{\mathbb{S},\mathbf{Nom}}(\emptyset)$ for a nominal algebraic signature $\mathbb{S}$ is the nominal set of terms up-to α-equivalence.

- Thirdly, there is the category of nominal substitutions **NomSub**. In this setting, the free algebra $T_{\mathbb{S},\mathbf{Nom}}(\emptyset)$ for a nominal algebraic signature $\mathbb{S}$ is the nominal set of terms up-to α-equivalence with the evident substitution structure.

For any set $\mathtt{N}$ of name metavariables, a function $\mathscr{V} : \mathtt{N} \to \mathscr{N}$, *i.e.* a *valuation* of the metavariables, induces a function $\mathbf{T}_{\mathbb{S},\mathbf{Set}_{\mathtt{N}}}(\emptyset) \to \mathbf{T}_{\mathbb{S},\mathbf{Nom}}(\emptyset)$, taking raw terms to terms up-to α-equivalence. Moreover, we can consider terms with explicit substitutions, by adding the substitution operation sub to a signature $\mathbb{S}$. A valuation function $\mathscr{V} : \mathtt{N} \to \mathscr{N}$ then also induces a function $\mathbf{T}_{\mathbb{S}+\mathsf{sub},\mathbf{Set}_{\mathtt{N}}}(\emptyset) \to \mathbf{T}_{\mathbb{S},\mathbf{NomSub}}(\emptyset)$. A more categorical view is that these valuation functions induce *monad morphisms* $(\mathbf{Nom}, \mathbf{T}_{\mathbb{S},\mathbf{Nom}}) \to (\mathbf{Set}, \mathbf{T}_{\mathbb{S},\mathbf{Set}_{\mathtt{N}}})$, and $(\mathbf{NomSub}, \mathbf{T}_{\mathbb{S},\mathbf{NomSub}}) \to (\mathbf{Set}, \mathbf{T}_{\mathbb{S}+\mathsf{sub},\mathbf{Set}_{\mathtt{N}}})$. This is explained in Secs 7.4 and 8.4.1 in the thesis.

И-MSOS-4: Rule structures for name-passing

The connection between abstract rules and a concrete rule format is broken down into five steps.

4a. A *rule structure* is defined as a formal syntactic structure, comprising 'premises' and a 'conclusion'. (See Defn. 8.1.1 in the thesis.)

4b. Every rule structure gives rise to an axiom of a nominal logic theory. Models of this nominal logic theory are transition systems that satisfy the rule. One of these models is identified as the *intended model*. (See Sec. 8.1.2 in the thesis.)

4c. Conditions are considered on rule structures, extending the first-order GSOS conditions (Conditions И-GSOS$^+$-1–12, Figs. 8.2 and 8.3, reproduced on page 19).

4d. Every rule structure that satisfies conditions И-GSOS$^+$-1–12 gives rise to an abstract rule (Sec. 8.4; Prop. 8.4.8, Thm. 8.4.10). A *family* of rule structures that all satisfy Conditions И-GSOS$^+$-1–12, also gives rise to an abstract rule, using the natural join-structure of the powerset (Sec. 8.4.4).

4e. The intended model of a family of rule structures is the initial algebra of the induced lifted monad (Sec. 8.4.4; Thm. 8.4.15). We conclude that wide-open bisimilarity is a congruence in the intended model (Thm. 8.4.16).

To explicate 4a, I reproduce the following definition.

Definition 8.1.1. Let $(\mathtt{X},\mathtt{N})$ be a pair of sets, with $\mathtt{N}$ finite. A *premise structure* over $(\mathtt{X},\mathtt{N})$ is an element of the set $\mathtt{X} \times \mathit{Lab}_g(\mathtt{N}) \times \mathtt{X}$. A *conclusion structure* over $(\mathtt{X},\mathtt{N})$ is a tuple in the set

$$\Sigma_{\mathbb{S},\mathbf{Set}_{\mathtt{N}}}(\mathtt{X}) \times \mathit{Lab}_g(\mathtt{N}) \times T_{\mathbb{S}+\mathsf{sub},\mathbf{Set}_{\mathtt{N}}}(\mathtt{X}).$$

(Here, $\mathit{Lab}_g(\mathtt{N})$ is a set of formal ground labels with names in $\mathtt{N}$.) A *rule structure for name-passing* over $(\mathtt{X},\mathtt{N})$ is a pair of a finite set of premise structures over $(\mathtt{X},\mathtt{N})$ and a conclusion structure over $(\mathtt{X},\mathtt{N})$.

И-GSOS$^+$-1.	Every term variable appears in the conclusion source or as a premise target.
И-GSOS$^+$-2.	The source of each premise appears in the conclusion source.
И-GSOS$^+$-3.	The target of any premise does not appear in any other premise.
И-GSOS$^+$-4.	The target of any premise does not appear in the conclusion source.
И-GSOS$^+$-5.	Each variable in the conclusion source is distinct.
	———
И-GSOS$^+$-6.	The binding variables in the conclusion source are not also free.
И-GSOS$^+$-7.	For each term parameter in the conclusion source, the binding variables are all distinct.
И-GSOS$^+$-8.	Bound names in premise labels are fresh for the premise sources.
И-GSOS$^+$-9.	Free names of the conclusion label appear in the conclusion source or in the premises.
И-GSOS$^+$-10.	Bound names of the conclusion label are fresh for the conclusion source.
И-GSOS$^+$-11.	Renamings in the conclusion target only affect relevant names.
И-GSOS$^+$-12.	No names become unbound in the induced transition.

The И-GSOS$^+$ format: conditions on a rule structure for name-passing (abbreviated from Figures 8.2 and 8.3 in the thesis). Conditions И-GSOS$^+$-1–5 are essentially the conditions of Positive GSOS. The remaining conditions are specific for name-passing.

For instance, the π-calculus rule for communication can be written as a rule structure:

$$\frac{\mathtt{x} \xrightarrow{\mathtt{c!d}} \mathtt{y} \qquad \mathtt{x}' \xrightarrow{\mathtt{c?(a)}} \mathtt{y}'}{\mathsf{par}(\mathtt{x},\mathtt{x}') \xrightarrow{\tau} \mathsf{par}(\mathtt{y},\mathtt{sub}(\mathtt{d},\langle \mathtt{a}\rangle \mathtt{y}'))}$$

This rule structure has term variables $\mathtt{X} = \{\mathtt{x},\mathtt{x}',\mathtt{y},\mathtt{y}'\}$, and name variables $\mathtt{N} = \{\mathtt{a},\mathtt{c},\mathtt{d}\}$. It consists of two premise structures, and one conclusion structure. Notice the explicit substitution that appears on the right hand side of the conclusion.

The rule structure can be considered as an axiom of nominal logic, in the nominal logic signature for nominal ground labelled transition system combined with the nominal algebraic signature for a parallel composition operator and a substitution operator. On the other hand, the rule structure satisfies Conditions И-GSOS$^+$-1–12, and hence describes abstract rule, and hence a monad lifting.

Other π-calculus rules are written as rule structures in Fig. 8.1 of the thesis (not reproduced here). They all satisfy Conditions И-GSOS$^+$-1–12 above.

We conclude with the principle theorem of the second part of the thesis.

Theorem 8.4.16. The intended model of the nominal logic theory arising from a class of rule structures in the И-GSOS$^+$ format has the following properties:

1. Axioms $И_g1$ and $И_g2$ (see page 14) are satisfied.
2. Wide open bisimilarity is a congruence. □

F. Beyond the thesis

The thesis provides a thorough study of operational models and structural operational semantics of name-passing calculi. This is an important contribution in itself, but it also provides a stepping stone towards the study of the foundations of more elaborate languages.

I am currently pursuing work in this direction, and I hold a Research Fellowship from the UK EPSRC with project title *Mathematical Operational Semantics for Data-Passing Processes*. Roughly speaking, the idea is to investigate the extent to which 'mathematical operational semantics' is relevant to increasingly sophisticated languages. One fruitful direction of research involves applying the theory of structured coalgebras to higher-order process calculi. This is mentioned briefly in Ch. 9 of the thesis, and research is now taking shape. Another exciting direction that I have been investigating recently is the possibility of using dependent type theory to formalize techniques for operational semantics. When the type theory is interpreted in the category of sets, we get first-order results; in the category of nominal substitutions, we get results that are relevant for name-passing calculi, and so on. This gives a treatment that is very general, but at the same time, the rule-formats that arise are quite concrete. I will present this work at LICS, later this year [Staton, 2008].

References

Note: Further references and a proper survey of the literature are provided in the thesis.

L. Aceto, W. Fokkink, and F. Vaandrager. Structural operational semantics. In *Handbook of Process Algebra*. Elsevier, 2001.

P. Aczel and N. P. Mendler. A final coalgebra theorem. In *Proc. CTCS'89*, pages 357–365, 1989.

B. Bloom, S. Istrail, and A. R. Meyer. Bisimulation can't be traced. *J. ACM*, 42(1):232–268, 1995.

G. L. Cattani and P. Sewell. Models for name-passing processes: interleaving and causal. *Inform. and Comput.*, 190:136–178, 2004.

M. P. Fiore and S. Staton. Comparing operational models of name-passing process calculi. *Inform. and Comput.*, 204(4):435–678, 2006. Extended abstract in Proc. CMCS'04.

M. P. Fiore and S. Staton. A congruence rule format for name-passing process calculi. *Inform. and Comput.*, 2007. Accepted for publication in special issue on Structural Operational Semantics.

M. P. Fiore and D. Turi. Semantics of name and value passing. In *Proc. LICS'01*, 2001.

M. P. Fiore, G. D. Plotkin, and D. Turi. Abstract syntax and variable binding. In *Proc. LICS'99*, 1999.

R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, I and II. *Inform. and Comput.*, 100(1):1–77, 1992.

U. Montanari and M. Pistore. History-dependent automata: An introduction. In *School on Formal Methods for the Design of Computer, Communication and Software Systems*, LNCS 3465, 2005.

A. M. Pitts. Nominal logic, a first order theory of names and binding. *Inform. and Comput.*, 186: 165–193, 2003.

A. M. Pitts. Alpha-structural recursion and induction. *J. ACM*, 53(3):459–506, 2006.

J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoret. Comput. Sci.*, 249(1):3–80, 2000.

S. Staton. General structural operational semantics through categorical logic. Accepted for *LICS'08*. 2008.

D. Turi and G. D. Plotkin. Towards a mathematical operational semantics. In *Proc. LICS'97*, 1997.