Università di Pisa

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea Specialistica in Informatica

Saturated Transition Systems for Presheaf Models

Relatore:
Chiar.mo Prof.
Ugo Montanari

Candidato:
Matteo Sammartino

Anno Accademico 2007/2008

"Space [...] is big. Really big. You just won't believe how vastly hugely mindbogglingly big it is. I mean you may think it's a long way down the road to the chemist, but that's just peanuts to space."

The Hitchhiker's Guide to the Galaxy, Douglas Adams

Alla mia famiglia

Table of Contents

1 Introduction

In the last thirty years, category theory established itself as a powerful framework for computer science, capable of representing the syntax and the semantics of programming languages. Quoting J.C.Reynolds: "substantial leverage can be gained in attacking this problem if these concepts can be defined concisely in a framework which has already proven its ability to impose uniformity and generality upon a wide variety of mathematics".

The most important foundational contribute to the theory was given by F. William Lawvere in his Ph.D. thesis [1], where an algebraic theory and its models are represented respectively as a cartesian category and as cartesian functors from this category to a semantic domain, each one providing a *denotational semantics*. In the 80s this approach has been applied to a variety of languages: the *typed λ-calculus* [2], with types and terms being objects and morphisms of a cartesian closed category; *Algol-like languages* [3], where objects and morphisms represent memory stacks and expansions of memory stacks. All these models share the idea of decoupling the building blocks of the language and its semantics.

More recently, in [4] the model-theoretic framework of *institutions* has been introduced: it provides a general way of embedding logical signatures and models into categories, and then expressing "truth-preserving" translations from one logical system to another. On the more concrete side of programming languages modelling, functorial models for process algebras such as CCS and π-calculus has been proposed [5,6,7]. Since such languages have a notion of *abstract semantics* strongly related to the behaviour, i.e. to operational semantics, these models consist of three levels, one more than the ones previously cited: resources (e.g. names in the case of π-calculus), syntax, defined through algebras, and behaviour, defined through the categorical equivalent of *labelled transition systems*: coalgebras. More specifically: a category $\mathbf{C}$ represents resources and resource changes as objects and morphisms; the category of functors $\mathbf{C} \to \mathbf{Set}$, called *presheaves*, collects all the associations between resources and elements using those resources, and between resource changes and functions between sets of elements, induced by these changes. Then categories of algebras and coalgebras based on presheaves give the abstract syntax and behaviour. One step further is obtaining a compositional behaviour: in [8] Turi introduces *bialgebras*, which combine algebras and coalgebras to obtain compositional models.

While the general trend is to remain completely category-theoretic, this thesis proposes a simpler and more circumscribed set-theoretic/category-theoretic approach: starting from a suitable class of transition systems, we want to investigate how, and under what conditions they can be represented as coalgebras over presheaves.

Let us briefly introduce coalgebras to explain the crucial points of the thesis. Given a endofunctor $\mathbf{B} : \mathbf{Set} \to \mathbf{Set}$, called *behavioural endofunctor*, the behaviour of a system can be categorically represented as a couple $\langle S, \alpha_S \rangle$, where S is a set of states and α_S is a function $S \to \mathbf{B}S$, called *structure map*, mapping each state to its continuation in $\mathbf{B}S$. For instance, if $\mathbf{B}S = \boldsymbol{\mathcal{P}}_f(L \times S)$, where $\boldsymbol{\mathcal{P}}_f$ is the functor sending a set to its finite powerset and L is a set of labels, one can obtain a representation of a labelled transition system $\langle S, \longrightarrow \rangle$ by letting

$$\alpha_S(s) = \{(l_1, s_1), \ldots, (l_n, s_n)\} \iff s \xrightarrow{l_1}_S s_1, \ldots, s \xrightarrow{l_n}_S s_n \ . \tag{1}$$

In this setting, a bisimulation on $\langle S, \alpha_S \rangle$ is a coalgebra $\langle \mathcal{R}, \alpha_{\mathcal{R}} \rangle$ where $\mathcal{R} \subseteq S \times S$ and $\alpha_{\mathcal{R}}$ is a function obtained (roughly speaking) "quotienting" α_S with respect to $\mathcal{R}$. This means that, if $\langle S, \alpha_S \rangle$ represents a labelled transition system, $\mathcal{R}$ contains pairs of states which can perform the same paths of computation, i.e. the ordinary notion of bisimulation on labelled transition systems.

If we move from coalgebras on $\mathbf{Set}$ to coalgebras on $\mathbf{Set}^{\mathbf{C}}$ we gain more structure: a coalgebra is $\langle \mathbf{S}, \alpha_{\mathbf{S}} \rangle$, where $\mathbf{S}$ is a presheaf and $\alpha_{\mathbf{S}}$ is a *natural transformation* (i.e. a family of functions with some additional conditions). Intuitively: systems have an indexed family of states which is closed under the functions generated by $\mathbf{S}$, because $\mathbf{S}\sigma : \mathbf{S}a \to \mathbf{S}b$; bisimulations are indexed families of relations and, as we will prove, they have the inherent property of being a congruence with respect to the functions $\mathbf{S}\sigma$, for each morphisms σ of $\mathbf{C}$. This property makes coalgebras over presheaves especially suitable for representing transition systems whose bisimilarity is a congruence.

After providing the needed background in **Chapter 2**, in **Chapter 3** the set-theoretic constructions are given: we define a class of labelled transition systems with typed states, and we call them *indexed labelled transition systems*; the associated notion of bisimulation is an indexed family of relations, one for each type. Then we show that, given a set of operation symbols and a family of functions interpreting them, one can get the coarsest bisimulation which is a congruence with respect to such functions by adding transitions of the form $s \xrightarrow{\sigma} f_\sigma(s)$, where f_σ is the function interpreting σ. We call *indexed saturated labelled transition systems* (ISLTS) the transition systems with the additional transitions.

In **Chapter 4** we discuss the category-theoretic aspects. Since ISLTSs are based on a family of states and a family of functions indexed respectively by types and operation symbols, moving to coalgebras over presheaves seems quite straightforward: types and operation symbols are embedded into a category $\mathbf{C}$, the preasheaf of states is a functor $\mathbf{C} \to \mathbf{Set}$ induced by the two families and the structure map is obtained from the transition relation. However, some difficulties are encountered. In particular we deal with two issues: 1) the definition of a suitable behavioural functor for representing ISLTSs and 2) the conditions a ISLTS must satisfy in order to be representable as a coalgebra. For issue 1), we show that the behavioural functor $\mathbf{B} : \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}^{\mathbf{C}}$ can be defined similarly to that for labelled transition systems on $\mathbf{Set}$, but it has to employ specific operators due to the indexed nature of presheaves. Issue 2) is related to the fact that the structure map must be a natural transformation $\mathbf{S} \to \mathbf{BS}$: we give sufficient and necessary conditions for the transition relation of a ISLTS to be representable as a natural transformation, and we provide an alternative behavioural functor $\widehat{\mathbf{B}}$ able to represent a transition system which is more complex than the original ISLTS, but has the same bisimilarity.

In **Chapter 5** we apply the theory: we give ISLTSs for early, late and open bisimilarity [9] for the π-calculus, for saturated bisimilarity [10] and we treat the case of ISLTs with polyadic operations. For each one we discuss the coalgebraic representation. In particular, for the case of open bisimilarity a behavioural endofunctor has been defined in [11]: we show that the ISLTS for open bisimilarity cannot be represented as a coalgebra of such functor, therefore the alternative behavioural functor must be employed.

2 Background

This chapter provides the categorical notions needed to understand the theory and an overview of π-calculus, upon which most examples will be based.

2.1 Categories, Algebras and Coalgebras

We assume that the reader is familiar with the notions of *category*, *functor*, *limit*, *colimit*, *initial object* and *final object* [12]. We recall the definition of *natural transformation*.

Definition 1. *Let* $\mathbf{F}, \mathbf{G} : \mathbf{C} \to \mathbf{G}$ *be functors between the categories* $\mathbf{C}$ *and* $\mathbf{G}$*. A* natural transformation η *from* $\mathbf{F}$ *to* $\mathbf{G}$ *associates to every object* $a \in |\mathbf{C}|$ *a morphism* $\eta_a : \mathbf{F}a \to \mathbf{G}a$ *in* $\mathbf{G}$*, such that for every morphism* $\sigma : a \to b$ *in* $\mathbf{C}$ *the following diagram commutes*

$$\begin{array}{ccc} \mathbf{F}a & \xrightarrow{\eta_a} & \mathbf{G}a \\ {\scriptstyle \mathbf{F}\sigma}\downarrow & & \downarrow{\scriptstyle \mathbf{G}\sigma} \\ \mathbf{F}b & \xrightarrow[\eta_b]{} & \mathbf{G}b \end{array}$$

Functors and natural transformations give rise to *functor categories.*

Definition 2. *Let* $\mathbf{C}$ *and* $\mathbf{G}$ *be categories. The functor category* $\mathbf{Func}(\mathbf{C}, \mathbf{G})$*, also denoted* $\mathbf{G}^{\mathbf{C}}$*, has as objects the functors from* $\mathbf{C}$ *to* $\mathbf{G}$ *and as morphisms the natural transformations between such functors.*

Limits and colimits in $\mathbf{Func}(\mathbf{C}, \mathbf{G})$ can be derived from those of category $\mathbf{G}$, as stated by the following proposition.

Proposition 3. *In a functor category limits and colimits are computed pointwise, i.e. the cone*

$$\begin{array}{ccccc} & & \mathbf{L} & & \\ & {\scriptstyle \eta_1}\nearrow & & \nwarrow{\scriptstyle \eta_n} & \\ \mathbf{F}_1 & & \cdots & & \mathbf{F}_n \end{array}$$

is a limit in $\mathbf{G}^{\mathbf{C}}$ *if and only if for all* $a \in |\mathbf{C}|$ *the cones*

$$\begin{array}{ccccc} & & \mathbf{L}a & & \\ & {\scriptstyle (\eta_1)_a}\nearrow & & \nwarrow{\scriptstyle (\eta_n)_a} & \\ \mathbf{F}_1 a & & \cdots & & \mathbf{F}_n a \end{array}$$

are limits in $\mathbf{G}$*. Analogously for colimits.*

Corollary 4. *The following statements hold:*

- *Let* $\mathbf{F}, \mathbf{H} \in |\mathbf{G}^{\mathbf{C}}|$ *and* $a \in |\mathbf{C}|$. *Then*

$$(\mathbf{F} \times \mathbf{H})a = \mathbf{F}a \times \mathbf{H}a \quad \textit{and} \quad (\mathbf{F} + \mathbf{H})a = \mathbf{F}a + \mathbf{H}a \ ;$$

- *The final object in* $\mathbf{G}^{\mathbf{C}}$ *is the functor* $\mathbf{1}$ *defined as follows:*

$$\mathbf{1}a = \mathbf{1}_{\mathbf{G}} \qquad \mathbf{1}\sigma = \mathrm{id}_{\mathbf{1}_{\mathbf{G}}} \ ,$$

for each object a *and each morphism* σ *in* $\mathbf{C}$.

Functor categories $\mathbf{Set}^{\mathbf{C}}$, for any category $\mathbf{C}$, are called *presheaf categories*.

Algebras and program syntax. The syntax of a programming language is usually defined by a *signature* Σ, which collects operators and constants; furthermore, Σ specifies a class of algebraic structures of the form $\langle A, \{\sigma^A \mid \sigma \in \Sigma\}\rangle$, where A is the *carrier* and each $\sigma^A : A^n \to A$ is the *interpretation* of the operator σ. The *term algebra* T_Σ, i.e. the algebra freely generated by the signature, is exactly the abstract syntax of the language.

We shall now describe two categorical approaches to algebras. The first approach, presented by F.W.Lawvere in [1], represents the term algebra and the specification as a category called *Lawvere theory*.

Definition 5. *A* Lawvere theory *is a cartesian category* $\mathcal{L}$ *with a distinguished object* $\mathbf{d}$ *such that every other object* a *in* $\mathcal{L}$ *is a finite power of* $\mathbf{d}$, *i.e. exists* n *such that* $a \cong \mathbf{d}^n$.

A morphism $\mathbf{d}^n \to \mathbf{d}$ in $\mathcal{L}$ corresponds to an n-ary operation, in particular morphisms of type $\mathbf{d}^0 = \mathbf{1} \to \mathbf{d}$ correspond to constants. Therefore, given $t_1, \ldots, t_n : \mathbf{1} \to \mathbf{d}$ and $\sigma : \mathbf{d}^n \to \mathbf{d}$, the morphism $\langle t_1, \ldots, t_n\rangle ; \sigma : \mathbf{1} \to \mathbf{d}$ corresponds to the term $\sigma(t_1, \ldots, t_n)$.

Definition 6. *A* model *of a Lawvere theory* $\mathcal{L}$ *is a cartesian functor* $\mathbf{M} : \mathcal{L} \to \mathbf{Set}$.

A model $\mathbf{M}$ can be regarded as an algebra: its carrier is $\mathbf{Md}$ and the interpretation of a n-ary operation σ of $\mathcal{L}$ is $\mathbf{M}\sigma : (\mathbf{Md})^n \to \mathbf{Md}$, since $\mathbf{M}$ preserves products; a natural transformation $\theta : \mathbf{M} \to \mathbf{N}$ between models acts like an algebra homomorphism, in fact we have

$$\begin{aligned}
\theta_{\mathbf{d}}(\mathbf{M}\sigma(v_1,\ldots,v_n)) &= && \text{(naturality of } \theta\text{)} \\
\mathbf{N}\sigma(\theta_{\mathbf{d}^n}(v_1,\ldots,v_n)) &= && \text{(cartesianity of } \mathbf{M} \text{ and } \mathbf{N}\text{)} \\
\mathbf{N}\sigma(\langle\theta_{\mathbf{d}};\pi_1,\ldots,\theta_{\mathbf{d}};\pi_n\rangle(v_1,\ldots,v_n)) &= && \\
\mathbf{N}\sigma(\theta_{\mathbf{d}}(v_1),\ldots,\theta_{\mathbf{d}}(v_n)) &\ . &&
\end{aligned}$$

The category of models and natural transformations between them is denoted by $\mathbf{Mod}(\mathcal{L})$.

Lawvere theories, as originally presented, are capable of representing only one-sorted specifications. However, definition 5 can be generalized to many-sorted specifications by allowing a set of distinguished objects, one for each sort. Equational specifications $\Gamma = \langle \Sigma, E\rangle$ are modelled by imposing $t_1 = t_2$ in $\mathcal{L}$ if and only if the same equation holds in E or, alternatively, by letting $\mathbf{M}t_1 = \mathbf{M}t_2$ in each model $\mathbf{M}$. In the following the Lawvere theory representing Γ will be denoted by $\mathcal{L}(\Gamma)$.

If Γ is a unary many-sorted specification, then a simpler representation of algebras is possibile: instead of considering the whole $\mathcal{L}(\Gamma)$ as the base category, we may consider the full subcategory $\mathbf{C}(\Gamma)$ whose objects are the distinguished objects of $\mathcal{L}(\Gamma)$, since products are not needed any more, and we can take the whole $\mathbf{Set}^{\mathbf{C}(\Gamma)}$ as $\mathbf{M}$. Notice that any category $\mathbf{C}$ may give rise to a specification with sorts $|\mathbf{C}|$ and operations $\|\mathbf{C}\|$, whose category of algebras is isomorphic to $\mathbf{Set}^{\mathbf{C}}$.

The other approch to categorical representation of algebras is described by the following definition.

Definition 7. *Let* $\boldsymbol{\Sigma} : \mathbf{Set} \to \mathbf{Set}$ *be any endofunctor on* $\mathbf{Set}$*. The category of* $\boldsymbol{\Sigma}$*-algebras, denoted by* $\mathbf{Alg}(\boldsymbol{\Sigma})$*, has as objects pairs* $\langle X, h\rangle$*, with* X *a set and* $h : \boldsymbol{\Sigma}X \to X$ *a function, called* interpretation map*. The morphisms of the category, called* $\boldsymbol{\Sigma}$-homomorphisms*, are functions between the underlying sets preserving the algebra structure, i.e. making the following diagram commute:*

$$\begin{array}{ccc}
\boldsymbol{\Sigma}X & \xrightarrow{h} & X \\
{\scriptstyle \Sigma f}\downarrow & & \downarrow{\scriptstyle f} \\
\boldsymbol{\Sigma}Y & \xrightarrow[g]{} & Y
\end{array}$$

The functor $\boldsymbol{\Sigma}$ represents the signature and it is usually defined as

$$\coprod_{\sigma\in\Sigma} X^{\mathrm{ar}(\sigma)} \ ,$$

namely the disjoint union of the sets obtained by applying (abstractly) each operator to the elements of the carrier; the disjoint union allows to define the interpretation map of a $\boldsymbol{\Sigma}$-algebra $A = \langle X, h \rangle$ as

$$h = [\sigma_1^A, \ldots, \sigma_n^A], \quad \sigma_1, \ldots, \sigma_n \in \Sigma \ ,$$

whose action is

$$h(\sigma(v_1, \ldots, v_n)) = \sigma^A(v_1, \ldots, v_n), \qquad v_1, \ldots, v_n \in X, \sigma \in \Sigma \ .$$

The action of Σ on functions is

$$\boldsymbol{\Sigma} f : \sigma(v_1, \ldots, v_n) \mapsto \sigma(f(v_1), \ldots, f(v_n)) \ ,$$

thus the requirement that $f : A = \langle X, h \rangle \to B = \langle Y, g \rangle$ has to satisfy to be a homomorphism can be expressed as

$$f(\sigma^A(v_1, \ldots, v_n)) = \sigma^B(f(v_1), \ldots, f(v_n)) \ .$$

Up until now we analyzed the case of algebras on **Set**, however definition 7 can be extended to any endofunctor $\boldsymbol{\Sigma} : \mathbf{C} \to \mathbf{C}$ on any category $\mathbf{C}$.

The term algebra is the initial object $\mathbf{0}_{\mathbf{Alg}(\boldsymbol{\Sigma})}$ in $\mathbf{Alg}(\boldsymbol{\Sigma})$. This is a consequence of the following property.

Proposition 8. *The initial object* $\mathbf{0}_{\mathbf{Alg}(\boldsymbol{\Sigma})} = \langle T, h_T \rangle$ *of* $\mathbf{Alg}(\boldsymbol{\Sigma})$ *is the least fix points of* $\boldsymbol{\Sigma}$*, i.e.* $\boldsymbol{\Sigma} T \cong T$*.*

As a consequence of this proposition, a tuple of terms $t_1, \ldots, t_n \in T$, representing the application of an operator σ, is sent by h_T to the term $\sigma(t_1, \ldots, t_n)$, again in T.

We now recall the notions of *context* and *congruence*. We introduce the following notation: $T_\Sigma(X)$ denotes the initial $(\Sigma + X)$-algebra and $ass : X \to A$ an assignment for the set of variables X into a Σ-algebra A. The free extension of ass is denoted by $\overline{ass} : T_\Sigma(X) \to A$.

Definition 9. *Given a specification* $\Gamma = \langle \Sigma, E \rangle$*, a* context *over* Γ *is an element of* $T_\Sigma(\{\bullet\})$ *with exactly one occurrence of the single variable* $\bullet$*. Given a* Γ*-algebra* A *and* $a \in A$*, we denote by* $C[a]$ *the element* $\overline{ass}(C)$ *of* A*, where* $ass(\bullet) = a$*. A relation* $\mathcal{R}$ *over* A *is a* congruence *if* $(a, b) \in \mathcal{R}$ *implies* $(C[a], C[b]) \in \mathcal{R}$ *for every context* C *over* Γ*.*

Coalgebras and program behaviour. The *operational semantics* of a language defines how programs are to be executed and what their observable effect is. More specifically, it aims at specifying the actions that programs can perform and their subsequent transitions, thus resulting in a suitable labelled transition system (LTS). In a categorical setting, LTSs are represented by *coalgebras* [13].

Definition 10. *Let* $\mathbf{B} : \mathbf{C} \to \mathbf{C}$ *be an endofunctor on a category* $\mathbf{C}$*. The category of* $\mathbf{B}$*-coalgebras, denoted by* $\mathbf{Coalg}(\mathbf{B})$*, has as objects pairs* $\langle X, h \rangle$*, with* $X \in |\mathbf{C}|$*, and* $h : X \to \mathbf{B}X$*, called* structure map*. The morphisms* $f : \langle X, h \rangle \to \langle Y, g \rangle$ *of the category, called* $\mathbf{B}$-cohomomorphisms*, are morphisms* $f \in \mathbf{C}(X, Y)$ *preserving the coalgebra structure:*

$$\begin{array}{ccc} X & \xrightarrow{h} & \mathbf{B}X \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle Bf} \\ Y & \xrightarrow[g]{} & \mathbf{B}Y \end{array}$$

A coalgebra $\langle X, h \rangle$ represents a system with states in X and whose behaviour is determined by h: if $\mathbf{B}$ is an endofunctor on $\mathbf{Set}$, which is the most common case, then X is a set and h is a function assigning to each state its continuation in $\mathbf{B}X$. For instance, the coalgebras of the endofunctor

$$\mathbf{B}X = X \times X + \mathbf{1}$$

are able to represent binary trees: the structure map sends a node to $\star$ if it does not have any children, i.e. the node is a leaf, otherwise sends it to its subtrees. The form of $\mathbf{B}$, in case $\mathbf{B}$-coalgebras are intended to represent non-deterministic LTSs, is:

$$\mathbf{B}X = \boldsymbol{\mathcal{P}}_f(L \times X) \ ,$$

where L is a set of labels and $\boldsymbol{\mathcal{P}}_f : \mathbf{Set} \to \mathbf{Set}$ is the (covariant) finite power-set functor:

$$\boldsymbol{\mathcal{P}}_f X = \{X' \mid X' \subseteq X\} \qquad \boldsymbol{\mathcal{P}}_f f : X \mapsto \{\ fx \mid x \in X\} \ ;$$

which models finite non-determinism.

Given a LTS $\langle S, \longrightarrow_S \rangle$ over a set of labels L, it can be represented as the coalgebra $\langle S, h \rangle$, where h satisfies:

$$s \xrightarrow{\alpha}_S s' \iff (\alpha, s') \in h(s) \ .$$

We now recall the notion of *bisimulation*, the standard way to represent behavioural equivalence between states of a LTS.

Definition 11. *Let $\langle S, \longrightarrow \rangle$ be a LTS. A bisimulation is a symmetric binary relation $\mathcal{R}$ on S satisfying the following: $(s, r) \in \mathcal{R}$ and $s \xrightarrow{l} s'$ implies*

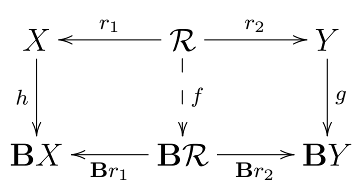
$$\exists r' : r \xrightarrow{l} r' \wedge (s', r') \in \mathcal{R} \, .$$

Given two states s and r, s is bisimilar *to r, written $s \sim r$, if there is a bisimulation $\mathcal{R}$ such that $(s, r) \in \mathcal{R}$.*

The equivalent notion in a coalgebraic setting is $\mathbf{B}$-*bisimulation*.

Definition 12. *Let $\langle X, h\rangle$ and $\langle Y, g\rangle$ be $\mathbf{B}$-coalgebras, for $\mathbf{B} : \mathbf{C} \to \mathbf{C}$. A $\mathbf{B}$-bisimulation between $\langle X, h\rangle$ and $\langle Y, g\rangle$ is an object $\mathcal{R} \in |\mathbf{C}|$ satisfying the conditions:*

- $X \xleftarrow{r_1} \mathcal{R} \xrightarrow{r_2} Y$ *is a* jointly-monic span*, i.e. for any object $A \in |\mathbf{C}|$ and any morphisms $f, g : A \to R$ such that $r_1; f = r_2; g$ and $r_2; f = r_2; g$, we must have $f = g$ (in* **Set** *this means $\mathcal{R} \subseteq X \times Y$);*
- *There is a structure map $f : \mathcal{R} \to B\mathcal{R}$ making the following diagram commute:*



If $\mathbf{C}$ is **Set**, bisimulation has an alternative characterization.

Proposition 13. *Let $\langle X, h\rangle$ and $\langle Y, g\rangle$ be $\mathbf{B}$-coalgebras, for $B : \mathbf{Set} \to \mathbf{Set}$. Then the kernel of each $f : \langle X, h\rangle \to \langle Y, g\rangle$ in $\mathbf{Coalg(B)}$ is a bisimulation. If $\langle Y, g\rangle$ is a final object, then the kernel of the unique morphism from $\langle X, h\rangle$ is the bisimilarity relation.*

The final object of $\mathbf{Coalg(B)}$ represents the *abstract semantics*, just like initial algebra represents abstract syntax. Moreover, the existence of a final coalgebra ensures that each class of bisimilar systems has a *minimal representative*. It is known that final coalgebras exist for any *accessible* endofunctor on *locally presentable* categories (see [14] for details).

2.2 The π-calculus

Let $\mathcal{N}$ be the set of names. We introduce the syntax of π-calculus agents.

Definition 14. *Let p, q range over agents and $x, y \in \mathcal{N}$. The set of agents is defined as follows:*

$$
\begin{aligned}
\alpha &:= \tau & p &:= \mathbf{0} \\
&\mid\ \overline{x}y & &\mid\ \alpha.p \\
&\mid\ x(y) & &\mid\ (\boldsymbol{\nu}x)p \\
& & &\mid\ [x = y]p \\
& & &\mid\ p \mid q \\
& & &\mid\ p + q
\end{aligned}
$$

The operators shown above have the following meaning:

1. $\mathbf{0}$ is a 0-ary operator representing a no-op agent.
2. *Prefixes*: $x(y).\_$ enables the input of an arbitrary name z at port x; $\overline{x}y.\_$ enables the output of the name y at port x; $\overline{x}(y).\_$ enables the output of the bound (see definition 15) name y at port x; $\tau.\_$, enables to perform the *silent action* τ.
3. *Restriction* $(\boldsymbol{\nu}x)\_$ makes the name x private, i.e. input/output actions at ports x and $\overline{x}$ are prohibited (communications between components of the process to which the restriction is applied are not prohibited).
4. *Match* $[x = y]\_$ allows the "execution" of its argument only if names x and y are identical.
5. *Composition* $\_ \mid \_$ enables its two arguments to act in parallel: they may act indipendently or *synchronize*, i.e. communicate with each other.
6. *Summation* $\_ + \_$ enables the whole agent to behave like one of the $+$ arguments.

Input and restriction are *binding operators*: the name y in $\overline{x}y.p$ and $(\boldsymbol{\nu}y).p$ is *bound*, and in both case the *scope* of the occurence is p. A name which is not bound is said to be *free*.

The set of all names occuring in p is denoted by $\mathrm{n}(p)$, the set of bound names is characterized as follows.

Definition 15. *Let p be an agent, the set* $\mathrm{bn}(p)$ *of its bound names is defined by structural induction as follows:*

$$\begin{aligned}
&\mathrm{bn}(\mathbf{0}) = \emptyset \\
&\mathrm{bn}(\overline{x}y.p) = \mathrm{bn}(\tau.p) = \mathrm{bn}([x = y]p) = \mathrm{bn}(p) \\
&\mathrm{bn}(x(y).p) = \{y\} \cup \mathrm{bn}(p) \\
&\mathrm{bn}((\boldsymbol{\nu} x)p) = \{x\} \cup \mathrm{bn}(p) \\
&\mathrm{bn}(p \mid q) = \mathrm{bn}(p + q) = \mathrm{bn}(p) \cup \mathrm{bn}(q)
\end{aligned}$$

The set of free names $\mathrm{fn}(p)$ clearly is $\mathrm{n}(p) \setminus \mathrm{bn}(p)$.

As well as for other calculi, an α-equivalence relation is defined: it partitions the set of agents in equivalence classes, each containing the agents with the same structure.

Definition 16. *The relation of α-equivalence for π-calculus agents, denoted by $\equiv_\alpha$, is defined as follows:*

$$\begin{aligned}
x(y).p &\equiv_\alpha x(y').p\{y'/y\} \\
(\boldsymbol{\nu} y).p &\equiv_\alpha (\boldsymbol{\nu} y').p\{y'/y\}
\end{aligned}$$

in both cases we must have $y' \notin \mathrm{fn}(p)$.

A function $\sigma : \mathcal{N} \to \mathcal{N}$ is called *renaming* and its action can be extended to agents.

Definition 17. *$p\sigma$ denotes the agent obtained from p by substituting $\sigma(z)$ for each occurrence of z in p for each z, with change of bound names to avoid captures.*

The expression "with change of bound names to avoid captures" means that, if σ maps a free name to a name in $\mathrm{bn}(p)$, the agent is first α-converted and then the substitution is applied.

After describing the syntax, we shall now describe the behaviour of π-calculus agents by means of a LTS. Four kinds of actions are available:

1. *Silent action* τ: the transition $p \xrightarrow{\tau} q$ means that p performs an action without interacting with the environment. Such actions arise from agents of the form $\tau.p'$ or $p' \mid p''$.
2. *Free output action* $\overline{x}y$: the transition $p \xrightarrow{\overline{x}y} q$ means that p emits the free name y on the port $\overline{x}$. Free output arise from the output prefix form $\overline{x}y.p$.

$$\text{PREF}\ \frac{-}{\alpha.p \xrightarrow{\alpha} p} \qquad \text{SUM}\ \frac{p \xrightarrow{\alpha} q}{p + q \xrightarrow{\alpha} q} \qquad \text{MATCH}\ \frac{p \xrightarrow{\alpha} q}{[x = x]p \xrightarrow{\alpha} q} \qquad \text{PAR}\ \frac{p \xrightarrow{\alpha} p'}{p \mid q \xrightarrow{\alpha} p' \mid q}\ \mathrm{bn}(\alpha) \cap \mathrm{fn}(q) = \emptyset$$

$$\text{COM}\ \frac{p \xrightarrow{u(x)} p' \qquad q \xrightarrow{\overline{u}y} q'}{p \mid q \xrightarrow{\tau} p' \mid q'\{y/x\}} \qquad \text{OPEN}\ \frac{p \xrightarrow{\overline{x}y} p'}{(\boldsymbol{\nu} y)p \xrightarrow{\overline{x}(y)} p'}\ y \neq x \qquad \text{CLOSE}\ \frac{p \xrightarrow{u(y)} p' \qquad q \xrightarrow{\overline{u}(y)} q'}{p \mid q \xrightarrow{\tau} (\boldsymbol{\nu} y)(p' \mid q')}$$

$$\text{RES}\ \frac{p \xrightarrow{\alpha} p'}{(\boldsymbol{\nu} y)p \xrightarrow{\alpha} (\boldsymbol{\nu} y)p'}\ y \notin \mathrm{n}(\alpha)$$

Fig. 1. Inference rules for the late π-calculus transition system. Rules involving operators $+$ and $\mid$ additionally have symmetric forms.

3. *Input action* $x(y)$: the transition $p \xrightarrow{x(y)} q$ means that p receives any name w on the port x and then evolves to $q\{w/y\}$. Input actions arise from the input prefix form $x(y).p$.
4. *Bound output action* $\overline{x}(y)$: the transition $p \xrightarrow{\overline{x}(y)} q$ means that p emits a bound name on the port $\overline{x}$ and (y) is the reference to where this name occurs. Bound output actions arise from free output actions which carry names out of their scope, as e.g. in the agent $(\boldsymbol{\nu} y)\overline{x}y.p$.

Definition 18. *The transition relation for π-calculus is the smallest relation satisfying the rules in Figure 1.*

We shall now look at some notions of behavioural equivalence for the π-calculus transition system. In the following we will use the phrase "a is fresh" to mean that the name a is different from any free name occurring in any of the agents in the definition.

Definition 19. *A (strong)* late *bisimulation is a symmetric binary relation $\mathcal{R}$ on agents satisfying the following: $(p, q) \in \mathcal{R}$ and $p \xrightarrow{\alpha} p'$ where* $\mathrm{bn}(\alpha)$ *is fresh implies that*

- *If $\alpha = a(x)$ then $\exists q' : q \xrightarrow{a(x)} q' \wedge \forall u : (p'\{u/x\}, q'\{u/x\}) \in \mathcal{R}$;*
- *Otherwise $\exists q' : q \xrightarrow{\alpha} q' \wedge (p', q') \in \mathcal{R}$.*

p and q are (strongly) late bisimilar, written $p \dot{\sim} q$, if they are related by a late bisimulation.

Definition 20. *A (strong)* early *bisimulation is a symmetric binary relation* $\mathcal{R}$ *on agents satisfying the following:* $(p,q) \in \mathcal{R}$ *and* $p \xrightarrow{\alpha} p'$ *where* $\mathrm{bn}(\alpha)$ *is fresh implies that*

- *If* $\alpha = a(x)$ *then* $\forall u\ \exists q' : q \xrightarrow{a(x)} q' \wedge (p'\{u/x\}, q'\{u/x\}) \in \mathcal{R}$*;*
- *Otherwise* $\exists q' : q \xrightarrow{\alpha} q' \wedge (p', q') \in \mathcal{R}$*.*

Early (strong) bisimilarity is denoted by $\dot{\sim}_E$*.*

Notice that the input case is treated in a special manner: the bound name is a placeholder for something to be received, therefore we must require that p' and q' are related for each value received. Early and late bisimilarities are not congruences, since they are not preserved by input prefix.

Sangiorgi in [9] introduces *open bisimulation*, which relates two agents if and only if, after being instantiated in (almost) all possible ways, they are able to perform the same action, ending up with agents which are related as well. Actually, not all instantiations are considered, but only those obtained by renamings σ such that, if a is a free name, formerly restricted and "freed" by a bound output, and b is any other free name, then $\sigma(a) \neq \sigma(b)$. Such restriction is motivated by the fact that a originally was a local name, thus intended to be fresh at any step of computation, for any instantiation of the agent.

The formal definition of open bisimulation employs relation on names, called *distinctions*, to "remember" names which must be kept distinct.

Definition 21. *A distinction is a finite, symmetric and irreflexive relation on names. A renaming* σ respects D *iff* $(a,b) \in D$ *implies* $\sigma(a) \neq \sigma(b)$*.*

Definition 22. *An* open bisimulation *is an indexed family* $\mathcal{R} = \{R_D\}$ *of symmetric, binary relations on agents satisfying the following: for all distinctions* D *and all renamings* σ *respecting* D*,* $(p,q) \in R_D$ *implies that*

- *If* $p\sigma \xrightarrow{\alpha} p'$*, with* $\mathrm{bn}(\alpha)$ *fresh, then* $\exists q : q\sigma \xrightarrow{\alpha} q' \wedge (p', q') \in R_{D\sigma}$*;*
- *If* $p\sigma \xrightarrow{\overline{a}(b)} p'$*, with* b *fresh, then* $\exists q : q\sigma \xrightarrow{\overline{a}(b)} q' \wedge (p', q') \in R_{D'}$*, where* $D' = D\sigma \cup \{\{b\} \times (\mathrm{fn}(p\sigma) \cup \mathrm{fn}(q\sigma))\}$*.*

We write $p \sim^O_D q$ *if there is an open bisimulation* $\mathcal{R}$ *such that* $(p,q) \in R_D$*.*

In [9] it is shown that open bisimilarity is a congruence with respect to all the operators.

3 Indexed Labelled Transition Systems

In this chapter we define a general notion of typed transition system, the associated typed bisimilarity and we show how such transition systems can be modified in order to make bisimilarity the largest bisimulation which is a congruence with respect to a given set of functions.

3.1 The basic transition system

Let $\mathcal{T}$ be a set of types and $\{S_a\}_{a\in\mathcal{T}}$ an indexed family of sets. This family can be seen as the family of states of an *indexed labelled transition system.*

Definition 23. *A* indexed labelled transition system *(ILTS) over* $\{S_a\}_{a\in\mathcal{T}}$ *and* L *is a couple* $\langle S, \longrightarrow\rangle$*, where* $S = \coprod_{a\in\mathcal{T}} S_a$ *and* $\longrightarrow \subseteq S \times L \times S$*.*

In the following we write s_a for s, if $s \in S_a$. Now we define the associated notion of bisimulation, which we call *indexed bisimulation.*

Definition 24. *An* indexed bisimulation *is a family of relations* $\mathcal{R} = \{R_a\}_{a\in\mathcal{T}}$ *such that:*

1. $(s_b, t_c) \in R_a \implies b = a \wedge c = a$;
2. $(s_a, t_a) \in R_a \wedge s_a \xrightarrow{l} s'_b \implies \exists t'_b : t_a \xrightarrow{l} t'_b \wedge (s'_b, t'_b) \in R_b$.

Two states s_a, t_a *are bisimilar, written* $s_a \sim^I_a t_a$*, if there is an indexed bisimulation* $\mathcal{R}$ *such that* $(s_a, t_a) \in R_a$*.*

This definition may sound artful, but the requirement that only two states with the same type can be bisimilar seems logical and in some interesting cases is not restrictive.

Example 25. Let $\mathcal{T}$ be the set of finite subsets of $\mathcal{N}$. Then the π-calculus late transition system can be represented as a ILTS on the family of sets of agents $\{\Pi_a\}_{a\in\mathcal{T}}$:

$$\Pi_a = \{p \mid \mathrm{fn}(p) \subseteq a\}$$

and labels $\{\tau, x(y), \overline{x}y, \overline{x}(y) \mid x, y \in \mathcal{N}\}$. The transition relation $\longrightarrow_\Pi$ is generated by the following rules:

$$\frac{p \xrightarrow{\tau} q \qquad p \in \Pi_a}{p_a \xrightarrow{\tau}_\Pi q_a} \qquad \frac{p \xrightarrow{x(y)} q \qquad p \in \Pi_a}{p_a \xrightarrow{x(y)}_\Pi q_{a\cup\{y\}}} \qquad \frac{p \xrightarrow{\overline{x}y} q \qquad p \in \Pi_a}{p_a \xrightarrow{\overline{x}y}_\Pi q_a} \qquad \frac{p \xrightarrow{\overline{x}(y)} q \qquad p \in \Pi_a}{p_a \xrightarrow{\overline{x}(y)}_\Pi q_{a\cup\{y\}}}$$

It easy to see that in this case $\sim^I$ coincides with *strong ground bisimilarity*, i.e. the ordinary bisimilarity for the late π-calculus transition system. Moreover, one can check if two agents p_a, q_b are bisimilar, with $a \neq b$: it is enough to check $s_{a\cup b}, t_{a\cup b}$ instead.

3.2 How to make bisimilarity a congruence

Let us consider a set $\mathcal{O}$ of operation symbols of the form $\sigma : a \to b$, where $a, b \in \mathcal{T}$, which is closed under composition, i.e. $\sigma \in \mathcal{O}$ and $\rho \in \mathcal{O}$ implies $\sigma \circ \rho \in \mathcal{O}$, associative and contains the special symbols id_a, which satisfy $\mathrm{id}_a \circ \sigma = \sigma \circ \mathrm{id}_a = \sigma$. We fix a well-behaved interpretation of such symbols, that is a family of functions $\{f_\sigma\}_{\sigma \in \mathcal{O}}$ satisfying the rules:

$$\frac{\sigma : a \to b}{f_\sigma : S_a \to S_b} \qquad \frac{s \in S_a}{f_{\mathrm{id}_a}(s) = s} \qquad \frac{\sigma_1 : a \to b \quad \sigma_2 : b \to c}{f_{\sigma_2} \circ f_{\sigma_1} = f_{\sigma_2 \circ \sigma_1}}$$

Then, given a ILTS $\langle S, \longrightarrow \rangle$, the indexed bisimulations which are congruences with respect to $\{f_\sigma\}_{\sigma \in \mathcal{O}}$ can be characterized explicitly: they are families $\mathcal{R}$ of relations on S such that, for each $(s_a, t_a) \in R_a$, if $f_\sigma(s_a) \xrightarrow{l} s'_c$ then there is t'_c such that $f_\sigma(t_a) \xrightarrow{l} t'_c$ and $(s'_c, t'_c) \in R_c$. This is an extension of *dynamic bisimulation* [15], defined for one-sorted monadic signatures and their contexts; in [15] it is also shown that dynamic bisimilarity is the coarsest ordinary bisimulation which is a congruence.

This notion of bisimulation is quite far from the ordinary one, but it suggests how $\langle S, \longrightarrow \rangle$ can be modified to make its ordinary bisimulations satisfy it: we add transitions of the form $s_a \xrightarrow{\sigma} f_\sigma(s_a)$. Formally, we define a *saturation function* Sat.

Definition 26. Sat *maps a ILTS* $\langle S, \longrightarrow \rangle$ *to* $\langle S, \longrightarrow_{SAT} \rangle$, *where:*

$$\longrightarrow_{SAT} = \longrightarrow \cup \bigcup_{a,b \in \mathcal{T}} \{(s_a, \sigma, f_\sigma(s_a)) \mid s_a \in S \wedge \sigma : a \to b \in \mathcal{O}\}$$

We call *indexed saturated labelled transition system* the ILTSs with the additional transitions.

Definition 27. *An* indexed saturated labelled transition system *(ISLTS) over* $\{S_a\}_{a \in \mathcal{T}}$, $\{f_\sigma\}_{\sigma \in \mathcal{O}}$ *and* L *is a ILTS* $\langle S, \longrightarrow \rangle$ *over* $\{S_a\}_{a \in \mathcal{T}}$ *and* $L \cup \mathcal{O}$ *such that, for each* s_a *and* $\sigma : a \to b \in \mathcal{O}$*:*

$$s_a \xrightarrow{\sigma} f_\sigma(s_a)\ .$$

We call action labels *the labels in* L *and* operation labels *those in* $\mathcal{O}$*.*

It's easy to see that, for a ISLTS, indexed bisimulations and ordinary bisimulations are indeed the same thing: condition 1 of definition 24 is automatically satisfied by ordinary bisimulations, because two bisimilar states must also have the same operation-labelled transitions, and this is true only if they have the same type. In the following, we choose to mantain the indexed version.

Now, let $ilts = \langle S, \longrightarrow \rangle$ be a ILTS and $islts = \mathrm{Sat}(ilts)$. Let us denote by $\mathcal{C}$ the set of congruences on S and by $\mathcal{B}, \mathcal{B}'$ the set of indexed bisimulations of the two systems, respectively. We have the following properties.

Lemma 28. $\mathcal{R} \in \mathcal{B} \cap \mathcal{C} \implies \mathcal{R} \in \mathcal{B}'$.

Proof. Given $(s_a, t_a) \in R_a$, the two states have the same action-labelled transitions and, for each $f_\sigma : P_a \to P_b$, we have that $(f_\sigma(s_a), f_\sigma(t_a)) \in R_b$, which implies $s_a \xrightarrow{\sigma} f_\sigma(s_a)$,$t_a \xrightarrow{\sigma} f_\sigma(t_a)$ in $islts$. □

Proposition 29. $\sim^I$ *on islts is the coarsest bisimulation of ilts which is a congruence.*

Proof. Since $\sim^I = \bigcup_{\mathcal{R} \in \mathcal{B}'} \mathcal{R}$ (here $\bigcup$ returns a family whose stages are the union of the arguments' stages with the same index), from lemma 28 we have that $\sim^I \supseteq \bigcup_{\mathcal{R} \in \mathcal{B} \cap \mathcal{C}} \mathcal{R}$. But $\sim^I \in \mathcal{B} \cap \mathcal{C}$, by definition of ISLTS and bisimulation, therefore $\sim^I = \bigcup_{\mathcal{R} \in \mathcal{B} \cap \mathcal{C}} \mathcal{R}$. □

Notice that, if the bisimilarity on $ilts$ is already a congruence with respect to $\{f_\sigma\}_{\sigma \in \mathcal{O}}$, then $islts$ has exactly the same bisimilarity.

4 Indexed Saturated LTSs as Coalgebras on Presheaves

In this chapter we will show how ISLTSs can be represented in a categorical setting. The basic idea is that the sets $\mathcal{T}$ and $\mathcal{O}$ can be seen as a category $\mathbf{C}$, with $|\mathbf{C}| = \mathcal{T}$ and $\|\mathbf{C}\| = \mathcal{O}$, and the association between types and states can be expressed through a presheaf $\mathbf{C} \to \mathbf{Set}$. More specifically, the families $\{S_a\}_{a\in\mathcal{T}}$ and $\{f_\sigma\}_{\sigma\in\mathcal{O}}$ at the base of each ISLTS define a presheaf by letting $\mathbf{P}a = P_a$ and $\mathbf{P}\sigma = f_\sigma$ (the conditions characterizing $\{f_\sigma\}_{\sigma\in\mathcal{O}}$ makes $\mathbf{P}$ a proper functor). Then $\mathbf{P}$ can be employed as the carrier of a coalgebra for a functor $\mathbf{B} : \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}^{\mathbf{C}}$.

We consider only ISLTSs, and not generic ILTSs, because coalgebraic bisimulations in a category of presheaf-based coalgebras are always congruences, as we will see in the first section. This restriction does not exclude ILTSs for which bisimilarity is already a congruence: they can be turned into ISLTSs through Sat without their bisimilarities being affected.

4.1 Some general properties of coalgebraic bisimulation

We first give an explicit characterization of $\mathbf{B}$-bisimulations, for any $\mathbf{B} : \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}^{\mathbf{C}}$.

Definition 30. *A $\mathbf{B}$-bisimulation between the $\mathbf{B}$-coalgebras $\langle \mathbf{P}, \gamma\rangle$ and $\langle \mathbf{Q}, \gamma'\rangle$ is a presheaf $\boldsymbol{\mathcal{R}}$ such that $\boldsymbol{\mathcal{R}}a \subseteq \mathbf{P}a \times \mathbf{Q}a$, for all $a \in |\mathbf{C}|$, and such that there is a natural transformation $\gamma_{\boldsymbol{\mathcal{R}}}$ which makes the following diagram commute:*

$$\begin{array}{ccccc} \mathbf{P} & \xleftarrow{\pi^1} & \boldsymbol{\mathcal{R}} & \xrightarrow{\pi^2} & \mathbf{Q} \\ \downarrow{\scriptstyle\gamma} & & \downarrow{\scriptstyle\gamma_{\boldsymbol{\mathcal{R}}}} & & \downarrow{\scriptstyle\gamma'} \\ \mathbf{BP} & \xleftarrow[\mathbf{B}\pi^1]{} & \mathbf{B}\boldsymbol{\mathcal{R}} & \xrightarrow[\mathbf{B}\pi^2]{} & \mathbf{BQ} \end{array}$$

Now we show that each $\mathbf{B}$-bisimulation is a congruence. As said in the background chapter, every presheaf can be seen as a unary many-sorted algebra and viceversa, therefore it makes sense to speak of congruence in this setting.

In the following the expression "$\mathbf{B}$-bisimulation on $\langle \mathbf{P}, \gamma\rangle$" stands for "$\mathbf{B}$-bisimulation between $\langle \mathbf{P}, \gamma\rangle$ and itself".

Proposition 31. *Let $\mathbf{B} : \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}^{\mathbf{C}}$ be an endofunctor on $\mathbf{Set}^{\mathbf{C}}$. Then each $\mathbf{B}$-bisimulation is a congruence with respect to all $\sigma \in \|\mathbf{C}\|$.*

Proof. Given any $\mathbf{B}$-bisimulation $\boldsymbol{\mathcal{R}}$ on a $\mathbf{B}$-coalgebra $\langle \mathbf{P}, \gamma \rangle$, by the naturality of π^1 and π^2 the diagram

$$\begin{array}{ccccc}
\mathbf{P}a & \xleftarrow{\pi^1_a} & \boldsymbol{\mathcal{R}}a & \xrightarrow{\pi^2_a} & \mathbf{P}a \\
\downarrow{\scriptstyle \mathbf{P}\sigma} & & \downarrow{\scriptstyle \boldsymbol{\mathcal{R}}\sigma} & & \downarrow{\scriptstyle \mathbf{P}\sigma} \\
\mathbf{P}b & \xleftarrow[\pi^1_b]{} & \boldsymbol{\mathcal{R}}b & \xrightarrow[\pi^2_b]{} & \mathbf{P}b
\end{array}$$

commutes, for all $a, b \in |\mathbf{C}|$ and $\sigma : a \to b$. Therefore $(\mathbf{P}\sigma(s_a), \mathbf{P}\sigma(t_a)) = \boldsymbol{\mathcal{R}}\sigma(s_a, t_a) \in \boldsymbol{\mathcal{R}}b$. □

Corollary 32. *Bisimilarity is a congruence with respect to all* $\sigma \in \|\mathbf{C}\|$.

In the background chapter a coalgebraic bisimulation on **Set** has been characterized as the kernel of a cohomomorphism. This definition can be extended to the case of coalgebras over $\mathbf{Set}^{\mathbf{C}}$, thanks to the following characterization of the *kernel of a natural transformation.*

Definition 33. *Let* $\mathbf{F}, \mathbf{G} : \mathbf{C} \to \mathbf{Set}$ *be functors and let* $\eta : \mathbf{F} \to \mathbf{G}$ *be a natural transformation between them. Then its* kernel *is a functor* $\mathbf{K}_\eta : \mathbf{C} \to \mathbf{Set}$ *defined by:*

$$\mathbf{K}_\eta a = \ker(\eta_a) \qquad \mathbf{K}_\eta \sigma(x, y) = (\mathbf{F}\sigma(x), \mathbf{F}\sigma(y)) \ .$$

Theorem 34. *Let* $\mathbf{B} : \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}^{\mathbf{C}}$ *be an endofunctor on* $\mathbf{Set}^{\mathbf{C}}$ *which preserves weak pullbacks (i.e. pullbacks for which the mediating morphism need not be unique). Then each* $\mathbf{B}$*-bisimulation is the kernel of a* $\mathbf{B}$*-cohomomorphism and viceversa.*

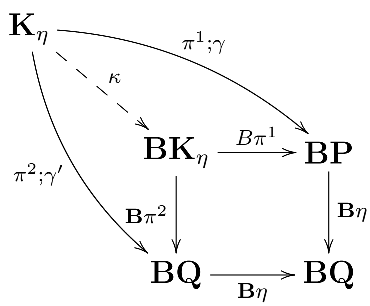
Proof. Given a coalgebra $\langle \mathbf{P}, \gamma \rangle$ and a $\mathbf{B}$-bisimulation $\boldsymbol{\mathcal{R}}$ on it, we consider the quotient presheaf $\mathbf{P}/\boldsymbol{\mathcal{R}}$:

$$(\mathbf{P}/\boldsymbol{\mathcal{R}})a = \mathbf{P}a/\boldsymbol{\mathcal{R}}a \qquad (\mathbf{P}/\boldsymbol{\mathcal{R}})\sigma : C \mapsto \{\mathbf{P}\sigma(c) \mid c \in C\} \ .$$

Then the family of functions $\epsilon_a = [-]_{\boldsymbol{\mathcal{R}}a}$ is a morphism $\epsilon : \mathbf{P} \to \mathbf{P}/\boldsymbol{\mathcal{R}}$ (naturality is straightforward). If we equip $\mathbf{P}/\boldsymbol{\mathcal{R}}$ with the structure map β defined by $\beta_a C = \gamma_a(s_a); (\mathbf{B}\epsilon)_a$, where $C \in (\mathbf{P}/\boldsymbol{\mathcal{R}})_a$ and s is any element of C, then ϵ becomes a $\mathbf{B}$-cohomorphism between $(\mathbf{P}, \gamma)$ and $(\mathbf{P}/\boldsymbol{\mathcal{R}}, \beta)$.

For the converse, let $\eta : \langle \mathbf{P}, \gamma \rangle \to \langle \mathbf{Q}, \gamma' \rangle$ be a $\mathbf{B}$-cohomomorphism. Since pullbacks in $\mathbf{Set}^{\mathbf{C}}$ are computed pointwise, we have that $\mathbf{P} \xleftarrow{\pi^1} \mathbf{K}_\eta \xrightarrow{\pi^2} \mathbf{Q}$ is a pullback of $\mathbf{P} \xrightarrow{\eta} \mathbf{Q} \xleftarrow{\eta} \mathbf{Q}$

and its image through $\mathbf{B}$ is a weak pullback. Then there is at least a $\kappa : \mathbf{K}_\eta \to \mathbf{B}\mathbf{K}_\eta$ which makes the following diagram commute.



□

4.2 From coalgebras to ISLTSs

Our first goal is to define a suitable behavioural endofunctor. If we consider the "naive" functor

$$\mathbf{BP} = \boldsymbol{\mathcal{P}}_f(\mathbf{L} \times \mathbf{P}) \ ,$$

where $\mathbf{L}$ is a chosen presheaf of labels, then $\mathbf{B}$-coalgebras model transition systems where the source and the target always have the same type, which is very restrictive. We need a functor which allows to select states from other stages of the carrier presheaf.

This can be achieved by employing endofunctors on $\mathbf{Set}^\mathbf{C}$ which, applied to a presheaf, modify the way it indexes the sets in its image. We derive such functors by "lifting" endofunctors on $\mathbf{C}$, which determine how indices are "renamed" in the resulting presheaf.

Definition 35. *Let $\mathbf{G} : \mathbf{C} \to \mathbf{C}$ be an endofunctor on $\mathbf{C}$. Then the functor $\boldsymbol{\delta}_\mathbf{G} : \mathbf{Set}^\mathbf{C} \to \mathbf{Set}^\mathbf{C}$ is defined on objects by*

$$\boldsymbol{\delta}_\mathbf{G}\mathbf{P}a = \mathbf{P}(\mathbf{G}a) \qquad \boldsymbol{\delta}_\mathbf{G}\mathbf{P}\sigma = \mathbf{P}(\mathbf{G}\sigma)$$

and on morphisms η by $(\boldsymbol{\delta}_\mathbf{G}\eta)_a = \eta_{\mathbf{G}a}$.

We are now able to define a more suitable behavioural functor: given a family $\mathscr{G} = \{\mathbf{G}_i\}_{i \in I}$, with $\mathbf{G}_i : \mathbf{C} \to \mathbf{C}$, the functor can be defined on presheaves as:

$$\mathbf{BP} = \boldsymbol{\mathcal{P}}_f(\mathbf{L} \times \coprod_{i \in I} \boldsymbol{\delta}_{\mathbf{G}_i}\mathbf{P}) \tag{2}$$

and on natural transformations $\eta : \mathbf{P} \to \mathbf{Q}$ as:

$$(\mathbf{B}\eta)_a = \boldsymbol{\mathcal{P}}_f \langle \mathbf{L}\mathrm{id}_a, \coprod_{i \in I} (\boldsymbol{\delta}_{\mathbf{G}_i} \eta)_a \rangle \ .$$

Theorem 36. *Each* **B***-coalgebra can be represented as a ISLTS.*

Proof. Let $\langle \mathbf{P}, \gamma \rangle$ be a **B**-coalgebra. Then we can define a ISLTS over $\{\mathbf{P}a\}_{a \in |\mathbf{C}|}$ and $\{\mathbf{P}\sigma\}_{\sigma \in \|\mathbf{C}\|}$ with set of labels $\widehat{L} = \bigcup_{a \in |\mathbf{C}|} \mathbf{L}a$, whose transition relation satisfies

$$s_a \xrightarrow{l} t_b \iff (l, t_b) \in \gamma_a(s_a)$$

and we denote it by $\langle S, \longrightarrow \rangle$.

Now we show that **B**-bisimulations are isomorphic to the indexed bisimulations on $\langle S, \longrightarrow \rangle$. Given a **B**-bisimulation $\boldsymbol{\mathcal{R}}$, let us consider the indexed family $\mathcal{R} = \{\boldsymbol{\mathcal{R}}a\}_{a \in |\mathbf{C}|}$ and let $\widehat{\gamma} : \boldsymbol{\mathcal{R}} \to \mathbf{B}\boldsymbol{\mathcal{R}}$ be the natural transformation

$$\widehat{\gamma}_a(s_a, t_a) = \{(l, (s'_b, t'_b)) \mid (l, s'_b) \in \gamma_a(s_a) \wedge (l, t'_b) \in \gamma_a(t_a)\} \ , \tag{3}$$

which clearly makes the bisimulation diagram for $\boldsymbol{\mathcal{R}}$ commute. To prove that $\mathcal{R}$ is an indexed bisimulation on $\langle S, \longrightarrow \rangle$, let us consider $(s_a, t_a) \in R_a$: by definition of $\langle S, \longrightarrow \rangle$ and $\widehat{\gamma}$ we have that $(l, (s'_b, t'_b)) \in \widehat{\gamma}_a(s_a, t_a)$ if and only if $s_a \xrightarrow{l} s'_b$, $t_a \xrightarrow{l} t'_b$, and by the definition of **B** and $\mathcal{R}$ we have $(s'_b, t'_b) \in R_b$; furthermore, s_a, t_a can perform only transitions with the same label, due to the **B**-bisimulation diagram. This covers the case of labels in $\widehat{L}$. The case of operation-labelled transitions follows from proposition 31.

Conversely, given an indexed bisimulation $\mathcal{R}$, the presheaf $\boldsymbol{\mathcal{R}}$ defined as:

$$\boldsymbol{\mathcal{R}}a = R_a \qquad \boldsymbol{\mathcal{R}}\sigma(s_a, t_a) = (\mathbf{P}\sigma(s_a), \mathbf{P}\sigma(t_a))$$

can be equipped with the structure map (3) so to satisfy the definition of **B**-bisimulation. □

4.3 From ISLTSs to coalgebras

The translation from ISLTSs to coalgebras is more difficult because, if we mantain a functor of the form (2), we are able to represent only those ISLTSs whose transition relation is expressible as a natural transformation. This is stated formally in the following proposition.

Proposition 37. *Let $\langle S, \longrightarrow \rangle$ be a ISLTS on $\{S_a\}$,$\{f_\sigma\}$ and $\bigcup_{a \in |\mathbf{C}|} \mathbf{L}a$ such that for each transition $s_a \xrightarrow{l} t_b$ there is a $\mathbf{G}_i \in \mathscr{G}$ such that $\mathbf{G}_i a = b$, and we denote it by $\mathbf{G}_{a,b}$. Let $\mathbf{P}$ be the presheaf $\mathbf{P}a = S_a$,$\mathbf{P}\sigma = f_\sigma$. Then the family of functions $\gamma_a(s_a) = \{(l, t_b) \mid s_a \xrightarrow{l} t_b \wedge t_b \in \boldsymbol{\delta}_{\mathbf{G}_{a,b}} \mathbf{P}a\}$ is a natural transformation $\mathbf{P} \to \mathbf{BP}$ if and only if:*

$$(i) \;\; s_a \xrightarrow{l} t_b \implies f_\sigma(s_a) \xrightarrow{\mathbf{L}\sigma(l)} f_{\mathbf{G}_{a,b}\sigma}(t_b) \; ;$$
$$(ii) \;\; f_\sigma(s_a) \xrightarrow{l} t_b \implies \exists s_a \xrightarrow{l'} r_c : \mathbf{L}\sigma(l') = l \wedge f_{\mathbf{G}_{a,c}\sigma}(r_c) = t_b \; .$$

Proof. By the definition of γ, condition (i) is equivalent to $\langle \mathbf{L}\sigma, \boldsymbol{\delta}_{\mathbf{G}_{a,b}} \mathbf{P}\sigma \rangle (l, t_b) \in \gamma_b(\mathbf{P}\sigma(s_a))$, that is $\mathbf{BP}\sigma(\gamma_a s_a) \subseteq \gamma_b(\mathbf{P}\sigma(s_a))$. Condition (ii) is specular and is equivalent to $\mathbf{BP}\sigma(\gamma_a(s_a)) \supseteq \gamma_b(\mathbf{P}\sigma(s_a))$. Both conditions are then equivalent to $\gamma_a; \mathbf{BP}\sigma = \mathbf{P}\sigma; \gamma_b$, for all $a, b \in |\mathbf{C}|$, i.e. the naturality condition. ☐

Notice that the coalgebraic representation of a ISLTS is not unique: any behavioural endofunctor satisfying the conditions on $\mathscr{G}$ given in this proposition is suitable, since the particular choice of $\mathscr{G}$ does not affect the structure map's definition and thus the bisimulations. However, some interesting transition systems do not satisfy conditions (i) and (ii), and we will show an example in next chapter.

An alternative representation is possible through the following behavioural endofunctor.

Definition 38. *The behavioural endofunctor $\widehat{\mathbf{B}} : \mathbf{Set}^{\mathbf{C}} \to \mathbf{Set}^{\mathbf{C}}$ is defined on objects by:*

$$\widehat{\mathbf{B}}\mathbf{P}a = \boldsymbol{\mathcal{P}_c}\Big(\coprod_{b \in |\mathbf{C}|} (\mathbf{C}(a,b) \times \mathbf{L}b \times \coprod_{c \in |\mathbf{C}|} \mathbf{P}c)\Big)$$
$$\widehat{\mathbf{B}}\mathbf{P}\sigma : A \mapsto \{(\sigma_2, l, p_a) \mid (\sigma_1, l, p_a) \in A \wedge \sigma; \sigma_2 = \sigma_1\} \; ;$$

where $\boldsymbol{\mathcal{P}_c}$ is the functor sending a set to its countable subsets, and on morphisms $\mu : \mathbf{P} \to \mathbf{Q}$ by:

$$\widehat{\mathbf{B}}\mu_a : A \mapsto \bigcup_{\sigma \in \mathbf{C}(a,b)} \{(\sigma, l, \mu_b(p_b)) \mid (\sigma, l, p_b) \in A\} \; ,$$

where $A \in \widehat{\mathbf{B}}\mathbf{P}a$ in the whole definition.

This functor represents a transition as a triple (σ, l, p_c), which intuitively groups two actions: the embedding of the starting state into the context σ, whose interpretation is given by the carrier presheaf, and a transition from this modified state to p_c, labelled with l. In next proposition we show that any ISLTS gives rise to a $\widehat{\mathbf{B}}$-coalgebra which groups the ISLTS's transitions as just described and whose bisimulations are isomorphic to those of the ISLTS.

Theorem 39. *Each ISLTS is representable as a* $\widehat{\mathbf{B}}$*-coalgebra.*

Proof. Let $\langle S, \longrightarrow\rangle$ be a ISLTS over $\{S_a\}$,$\{f_\sigma\}$, and L. Then the carrier presheaf $\mathbf{P}$ is:

$$\mathbf{P}a = S_a \qquad \mathbf{P}\sigma = f_\sigma \ ,$$

the label presheaf is any $\mathbf{L} : \mathbf{C} \to \mathbf{Set}$ such that $\bigcup_{a\in|\mathbf{C}|} \mathbf{L}a = L$ (e.g. $\mathbf{L}a = L$ and $\mathbf{L}\sigma = \mathrm{id}_L$, for all $a \in |\mathbf{C}|$ and $\sigma \in \|\mathbf{C}\|$) and the structure map $\gamma : \mathbf{P} \to \widehat{\mathbf{B}}\mathbf{P}$ is defined by:

$$(\sigma, l, s'_b) \in \gamma_a(s_a) \iff s_a \xrightarrow{\sigma} f_\sigma(s_a) \xrightarrow{l} s'_b \ ,$$

where l is an action label. To prove that it defines a proper natural transformation, we have to check that the diagram

$$\begin{array}{ccc} \mathbf{P}a & \xrightarrow{\gamma_a} & \widehat{\mathbf{B}}\mathbf{P}a \\ {\scriptstyle \mathbf{P}\sigma}\downarrow & & \downarrow{\scriptstyle \widehat{\mathbf{B}}\mathbf{P}\sigma} \\ \mathbf{P}b & \xrightarrow[\gamma_b]{} & \widehat{\mathbf{B}}\mathbf{P}b \end{array}$$

commutes. Let $s_a \in \mathbf{P}a$, then the upper path leads to the result of $\widehat{\mathbf{B}}\mathbf{P}\sigma$ applied to the triplets in $\gamma_a(s_a)$, which represent transitions of the form $s_a \xrightarrow{\sigma'} f_{\sigma'}(s_a) \xrightarrow{l} t_b$: if σ' can be written as $\sigma; \sigma''$, then $\widehat{\mathbf{B}}\mathbf{P}\sigma$ "removes" σ and generates the new chain of transitions $\xrightarrow{\sigma''}\xrightarrow{l}$, which are exactly those of $f_\sigma(s_a)$. Naturality follows from the fact that the lower path leads to the set of triplets representing all the transitions

$$f_\sigma(s_a) \xrightarrow{\sigma'''} f_{\sigma;\sigma'''}(s_a) \xrightarrow{l} s'_d \ ,$$

for all the morphism σ''' with domain b.

We now prove that $\widehat{\mathbf{B}}$-bisimulations on $\langle \mathbf{P}, \gamma\rangle$ and indexed bisimulations on $\langle S, \longrightarrow\rangle$ are in one-to-one correspondence. Given an indexed bisimulation $\mathcal{R}$, we define the equivalent $\widehat{\mathbf{B}}$-bisimulation $\boldsymbol{\mathcal{R}}$ as follows:

$$\boldsymbol{\mathcal{R}}a = R_a \qquad \boldsymbol{\mathcal{R}}\sigma(s_a, t_a) = (\mathbf{P}\sigma(s_a), \mathbf{P}\sigma(t_a)) \ .$$

It actually is a $\widehat{\mathbf{B}}$-bisimulation, since it can be equipped with the structure map $\widehat{\gamma}$:

$$\widehat{\gamma}_a(s_a, t_a) = \{(\sigma, l, (s'_b, t'_b)) \mid (\sigma, l, s'_b) \in \gamma_a(s_a) \wedge (\sigma, l, t'_b) \in \gamma_a(t_a)\} \ ,$$

clearly making the bisimulation diagram commute.

Conversely, given a $\widehat{\mathbf{B}}$-bisimulation $\boldsymbol{\mathcal{R}}$ equipped with the structure map $\widehat{\gamma}$ defined above, the family of relations $R_a = \boldsymbol{\mathcal{R}}a$ is an indexed bisimulation on $\langle S, \longrightarrow \rangle$. In fact, let us consider $(s_a, t_a) \in R_a$: by the definition of $\widehat{\gamma}$ and by the bisimulation diagram we have that $s_a \xrightarrow{\sigma} f_\sigma(s_a) \xrightarrow{l} s'_b$ if and only if $t_a \xrightarrow{\sigma} f_\sigma(t_a) \xrightarrow{l} t'_b$, which implies, for $\sigma = \mathrm{id}_a$, the two states can perform the same action $l \in L$, becoming then elements related in R_b, since $(s''_b, t''_b) \in \boldsymbol{\mathcal{R}}b$ by the definition of $\widehat{\mathbf{B}}$. From proposition 31 we get that s_a and t_a can also perform the same operation-labelled transitions and result in two related states. $\square$

5 Examples

In this chapter we apply the theory to some concrete examples: early, late and open bisimilarity for the π-calculus and saturated bisimilarity. In the end we analyze the case of ISTLSs based on a set of polyadic operations.

5.1 Early bisimilarity

To construct a ISLTS for early bisimilarity we represent names as natural numbers and we define the set of types $\mathcal{T}_\pi$ as the set of finite cardinals (we denote the set $\{1, \ldots, n\}$ by n) and the set of operations $\mathcal{O}_\pi$ as the set of injective functions between them; they represent respectively sets of names and injective renamings. We use injective renamings because it is a known fact that early bisimilarity is closed under these operations (see e.g. proposition 2 in [16]).

Agents are typed with the set of their free names, but we do not consider the whole set of agents, instead we construct an abstract syntax containing only *canonical agents*, i.e. agents which are representatives of α-equivalence classes. The family of agents $\{\Pi_n\}_{n \in \mathcal{T}_\pi}$ is defined by the method of De Bruijn levels [17], according to the rules in figure 2. As we can see from the rules for input prefix and restriction, the name bound by a binding prefix is always $n+1$: this is how canonical agents are generated.

The transition system for early bisimilarity $satls_E = \langle \Pi^E, \longrightarrow_E \rangle$ is the ISLTS on $\{\Pi^E_n\}_{n \in \mathcal{T}_\pi}$, $\{f^E_\sigma : p \mapsto p\sigma\}_{\sigma \in \mathcal{O}_\pi}$ and $L_E = \{\tau, xy, \overline{x}y, \overline{x}(), x() \mid x, y \in \mathcal{N}\}$ whose action-labelled transitions are generated by the rules in figure 3, where the transitions in the premises belong to the early π-calculus transition relation, i.e. the transition relation obtained replacing the

$$\mathbf{0} \in \Pi_\emptyset \qquad \frac{p \in \Pi_n \quad m \in \mathcal{T}_\pi \quad m \supseteq n}{p \in \Pi_m} \qquad \frac{p \in \Pi_n \quad q \in \Pi_n}{p \mid q \in \Pi_n} \qquad \frac{p \in \Pi_n \quad x, y \in n}{[x = y]p \in \Pi_n}$$

$$\frac{p \in \Pi_{n+1} \quad x \in n}{x(n+1).p \in \Pi_n} \qquad \frac{p \in \Pi_n \quad x, y \in n}{\overline{x}y.p \in \Pi_n} \qquad \frac{p \in \Pi_{n+1}}{(\boldsymbol{\nu}\, n+1)p \in \Pi_n}$$

Fig. 2. Rules generating the family of agents.

$$\frac{p \xrightarrow{\tau} q \quad p \in \Pi_n}{p_n \xrightarrow{\tau}_E q_n} \qquad \text{E-INPUT}_1 \; \frac{p \xrightarrow{xy} q \quad p \in \Pi_n \quad y \in n}{p_n \xrightarrow{xy}_E q_n} \qquad \text{E-INPUT}_2 \; \frac{p \xrightarrow{x\, n+1} q \quad p \in \Pi_n}{p_n \xrightarrow{x()}_E q_{n+1}}$$

$$\frac{p \xrightarrow{\overline{x}y} q \quad p \in \Pi_n}{p_n \xrightarrow{\overline{x}y}_E q_n} \qquad \frac{p \xrightarrow{\overline{x}(n+1)} q \quad p \in \Pi_n}{p_n \xrightarrow{\overline{x}()}_E q_{n+1}}$$

Fig. 3. Inference rules for the ISLTS for early bisimilarity.

rules for input and parallel composition in figure 1 with the following:

$$\frac{-}{x(y).p \xrightarrow{x(u)} p\{u/y\}} \qquad \frac{p \xrightarrow{x(u)} p' \quad q \xrightarrow{\overline{x}u} q'}{p \mid q \xrightarrow{\tau} p' \mid q'}$$

Notice that an agent $x(n+1).p$ can input either a known name (rule E-INPUT$_1$) or a fresh name, represented by $n+1$ (rule E-INPUT$_2$). This is consistent, since receiving a fresh name is the same thing as first α-converting and then removing the prefix (recall that agents are representatives of α-equivalence classes, then α-converting is meaningless). Besides, notice that bound input/output labels do not carry the bound name in order to avoid redundancy: it is know to be $n + 1$.

The construction explained so far leads to the following result.

Proposition 40. *$p_n \dot{\sim}_E q_n$ if and only if $p_n \sim^I_n q_n$ with respect to $islts_E$.*

If we let $\mathcal{O}_\pi$ contain all renamings, and we modify $islts_E$ by adding the transitions for non-injective renamings, then $\sim^I$ coincide with *early congruence*, i.e. the coarsest early bisimulation which is preserved by all renamings.

Now we consider coalgebras. We denote by $\mathbf{I}$ the category induced by $\mathcal{T}_\pi$ and $\mathcal{O}_\pi$. Since it is a known property that, for any injective renaming σ, if $p \xrightarrow{l} q$ then $p\sigma \xrightarrow{l\sigma} q\sigma$ and viceversa (see e.g. [18] lemma 3 and 4), if we define a suitable endofunctor of the form (2) we can apply proposition 37 to $islts_E$.

The needed constructions are:

- The *presheaf of names* $\mathcal{N} : \mathbf{I} \to \mathbf{Set}$, which is just the embedding in $\mathbf{Set}$.

- The *presheaf of labels*

$$\begin{aligned}\mathbf{L}_\pi = \mathbf{1} & \qquad \text{(Silent Action)}\\ + \mathcal{N} \times \mathcal{N} & \qquad \text{(Free Input/Output)}\\ + \mathcal{N} & \qquad \text{(Bound Input/Output)}\end{aligned}$$

- The *dynamic allocation operator* $\boldsymbol{\delta} : \mathbf{Set}^{\mathbf{I}} \to \mathbf{Set}^{\mathbf{I}}$, generated, according to definition 35, by the endofunctor on $\mathbf{I}$ which sends n to $n+1$ and σ to $[\sigma, \mathrm{id}_1]$.

Finally, the behavioural functor is

$$\mathbf{BP} = \boldsymbol{\mathcal{P}}_f(\mathbf{L}_\pi \times (\mathbf{P} + \boldsymbol{\delta}\mathbf{P})) \ . \tag{4}$$

Remark 41. The abstract syntax defined in figure 2 can be obtained, more elegantly, as the initial $\boldsymbol{\Sigma}$-algebra $\mathbf{T}_{\boldsymbol{\Sigma}}$ for a functor $\boldsymbol{\Sigma} : \mathbf{Set}^{\mathbf{I}} \to \mathbf{Set}^{\mathbf{I}}$. For instance, let us consider the endofunctor given in [6]:

$$\boldsymbol{\Sigma}\mathbf{P} = \mathbf{1} + \mathbf{P} \times \mathbf{P} + \mathcal{N} \times \boldsymbol{\delta}\mathbf{P} + \mathcal{N} \times \mathcal{N} \times \mathbf{P} + \boldsymbol{\delta}\mathbf{P} + \mathcal{N} \times \mathcal{N} \times \mathbf{P} \ ,$$

each addend modeling respectively: the inert process, parallel composition, input prefix, output prefix, restriction and match.

Notice the usage of $\boldsymbol{\delta}$: it selects subagents with one additional free name, which will be bound in the resulting agent; e.g. an agent in $\mathbf{T}_{\boldsymbol{\Sigma}}n$ with an input prefix is represented by the couple (i, p), where $i \in n$ and $p \in \mathbf{T}_{\boldsymbol{\Sigma}}(n+1)$: the additional name in $n+1$ is considered as the bound name of the prefix. In other words: $\mathbf{T}_{\boldsymbol{\Sigma}}$ is the presheaf of agents quotiented by α-equivalence.

5.2 Late bisimilarity

Unlike early bisimilarity for the early transition system, late bisimilarity is not the standard bisimilarity of the late transition system, because it requires a quantification over names for input transitions. Then we must consider an ad-hoc transition system, in which this quantification is operationally represented.

This can be done by defining an additional syntactic construct $\lambda x.p$, called *abstraction prefix*, which intuitively means that p has performed an input transition but it is still waiting to receive the actual data; as soon as the value u is received, the agent becomes $p\{u/x\}$. We let $\mathrm{fn}(\lambda x.p) = \mathrm{fn}(p) \setminus \{x\}$.

We employ again $\mathcal{T}_\pi$ and $\mathcal{O}_\pi$, because late bisimilarity is closed under injective renaming. The family of agents $\{\Pi^L_n\}_{n\in\mathcal{T}_\pi}$ is $\{\Pi^E_n\}_{n\in\mathcal{T}_\pi}$ plus the abstract agents constructed by the rule

$$\frac{p \in \Pi^L_{N+1}}{\lambda\, n+1.p \in \Pi^L_N}$$

The transition system for late bisimilarity is the ISLTS $islts_L = \langle \Pi^L, \longrightarrow_L\rangle$ on $\{\Pi^L_n\}_{n\in\mathcal{T}_\pi}$, $\{f^L_\sigma : p \mapsto p\sigma\}_{\sigma\in\mathcal{O}_\pi}$ and $L_L = \{\tau, x, \overline{x}y, \overline{x}(), x() \mid x, y \in \mathcal{N}\}$, whose action-labelled transitions are derived by the late π-calculus transition system and the inference rules in figure 3, with the exception that E-INPUT$_1$ and E-INPUT$_2$ are replaced by the rules:

$$\text{L-INPUT}_1\ \frac{p \xrightarrow{x(n+1)} q \quad p \in \Pi^L_n}{p_n \xrightarrow{x()} q_{n+1}} \qquad \text{L-INPUT}_2\ \frac{p \xrightarrow{x(n+1)} q \quad p \in \Pi^L_n, y \in n}{p_n \xrightarrow{x}_L (\lambda\, n+1.p)_n \xrightarrow{y}_L q\{y/n+1\}_n}$$

The rule L-INPUT$_1$ adds a transition "declaring" that p_n can perform an input on channel x, and another one which performs the actual reception; the rules L-INPUT$_2$ treats the case of a fresh name input. The following result is an immediate consequence.

Proposition 42. *$p_n \dot{\sim} q_n$ if and only if $p_n \sim^I_n q_n$ with respect to $islts_L$.*

Employing all renamings instead of just injective ones, with a suitable modification of $islts_L$, makes $\sim^I$ equivalent to *late congruence*, i.e. the coarsest late bisimulation which is preserved by all renamings.

The behavioural functor is again (4), the only different thing is how labels are modeled: the component of $\mathbf{L}_\pi$ for bound input/output is also used for the new input actions.

5.3 Open bisimilarity

As seen in the background, distinctions and distinction-preserving renamings play a central role in defining open bisimilarity, therefore we let the set of types $\mathcal{T}_O$ be the set of distinctions, which we denote by (n, d_n), where n is the carrier of the relation d_n, and we let the set of operations $\mathcal{O}_O$ be the set of all the distinction-respecting functions, represented as functions between distinctions, i.e. if $\sigma : n \to m$ is such that $(a, b) \in (n, d_n)$ implies $(\sigma(a), \sigma(b)) \in (m, d_m)$, then $\sigma : (n, d_n) \to (m, d_m) \in \mathcal{O}_O$. Furthermore, we define two operators δ^-, δ^+ on distinctions:

$$\delta^-(n, d_n) = (n+1, d_n) \qquad \delta^+(n, d_n) = (n+1, d_{n+1}) ,$$

where d_{n+1} is the symmetric closure of $d_n \cup \{(n+1, i) \mid i \in n\}$.

The family of agents $\{\Pi^O_d\}_{d \in \mathcal{T}_O}$ is defined as $\Pi^O_{(n,d_n)} = \Pi^E_n$, and the interpretation of morphisms is given by the family $\{f^O_\sigma : p \mapsto p\sigma\}_{\sigma \in \mathcal{O}_O}$. Then the transition system for open bisimilarity $islts_O = \langle \Pi^O, \longrightarrow_O \rangle$ is the ISLTS on $\{\Pi^O_d\}_{d \in \mathcal{T}_O}$, $\{f^O_\sigma\}_{\sigma \in \mathcal{O}_O}$ and the set of labels $\{\tau, x(), \overline{x}y, \overline{x}() \mid x, y \in \mathcal{N}\}$, whose action-labelled transitions are induced by the late π-calculus transition relation as follows:

$$\frac{p \xrightarrow{\tau} q \quad p \in \Pi^O_d}{p_d \xrightarrow{\tau}_O q_d} \qquad \frac{p \xrightarrow{x(n+1)} q \quad p \in \Pi^O_d}{p_d \xrightarrow{x()}_O q_{\delta^- d}} \qquad \frac{p \xrightarrow{\overline{x}y} q \quad p \in \Pi^O_d}{p_d \xrightarrow{\overline{x}y}_O q_d} \qquad \frac{p \xrightarrow{\overline{x}(n+1)} q \quad p \in \Pi^O_d}{p_d \xrightarrow{\overline{x}()}_O q_{\delta^+ d}}$$

The key points is the way types are changed by a bound output transition: the resulting agent has an enriched distinction, ensuring that the extruded name will be kept distinct from all the other free names by any later operation-labelled transition. This represents exactly the generation of a new distinction for the bound output transition case in the open bisimulation step. All the other transitions do not alter the distinction (input just changes the carrier). Therefore we have the following fact.

Proposition 43. *$p_d \sim^O_d q_d$ if and only if $p_d \sim^I_d q_d$ with respect to $islts_O$.*

To build a coalgebra representing $islts_O$, we first check if it is possible to build a coalgebra over a functor of the form (2). Let $\mathbf{D}$ be the category induced by $\mathcal{T}_O$ and $\mathcal{O}_O$, we need the following constructions:

- *Presheaf of names* $\mathcal{N} : \mathbf{D} \to \mathbf{Set}$, which sends (n, d_n) to the set n and $\sigma : (n, d_n) \to (m, d_m)$ to the function $\sigma : n \to m$.
- *Dynamic allocation operators* $\boldsymbol{\delta}^+, \boldsymbol{\delta}^- : \mathbf{Set}^\mathbf{D} \to \mathbf{Set}^\mathbf{D}$ induced, according to definition 35, by the endofunctors on $\mathbf{D}$ having δ^+, δ^- as actions on objects and sending a morphism $\sigma : d \to d'$ to $[\sigma, \mathrm{id}_\mathbf{1}]$, seen respectively as a morphism $\delta^+ d \to \delta^+ d'$ and as a morphism $\delta^- d \to \delta^- d'$.

In [11] an endofunctor $\mathbf{B} : \mathbf{Set}^\mathbf{D} \to \mathbf{Set}^\mathbf{D}$ is given:

$$\begin{aligned} \mathbf{BP} = \boldsymbol{\mathcal{P}}_f(\mathbf{P} & & \text{(Silent Action)} \\ + \mathcal{N} \times \mathcal{N} \times \mathbf{P} & & \text{(Output)} \\ + \mathcal{N} \times \boldsymbol{\delta}^+\mathbf{P} & & \text{(Bound Output)} \\ + \mathcal{N} \times \boldsymbol{\delta}^-\mathbf{P}) & & \text{(Input)} \end{aligned}$$

which, setting $\mathbf{L}_O = \mathbf{1} + \mathcal{N} + \mathcal{N} \times \mathcal{N}$, can be rewritten in the form (2) as

$$\mathbf{BP} = \boldsymbol{\mathcal{P}}_f(\mathbf{L}_O \times (\mathbf{P} + \boldsymbol{\delta}^+\mathbf{P} + \boldsymbol{\delta}^-\mathbf{P}))$$

(the two functors are naturally isomorphic). However, a **B**-coalgebra representing $islts_O$ does not exists.

To see this, let us consider the agent $p = [1 = 2]\overline{3}4.\mathbf{0} \in \Pi^O_{(4,\emptyset)}$: it has no outgoing transitions. Let $\sigma : (4, \emptyset) \to (4, \emptyset)$ be the function in $\mathcal{O}_O$ defined by

$$\sigma(1) = 1, \sigma(2) = 1, \sigma(3) = 3, \sigma(4) = 4,$$

then $f^O_\sigma(p) \xrightarrow{\overline{3}4}_O \mathbf{0}$, but this violates condition (ii) in proposition 37. Hence we have to employ the endofunctor of definition 38, whose action on objects in this case can be refined as follows:

$$\widehat{\mathbf{B}}\mathbf{P}d = \boldsymbol{\mathcal{P}}_{\boldsymbol{c}}(\coprod_{d' \in |\mathbf{D}|} (\mathbf{D}(d, d') \times \mathbf{L}d' \times (\mathbf{P}d + \boldsymbol{\delta}^+\mathbf{P}d + \boldsymbol{\delta}^-\mathbf{P}d))) \; .$$

5.4 Saturated bisimilarity

In [10] Bonchi and Montanari present a general theory of bisimilarity. They define an interactive system as a state-machine which can interact with the environment through an evolving interface.

Definition 44. *A* context interactive system (CIS) $\mathcal{I}$ *is a quadruple* $\langle\langle S, \Sigma\rangle, \mathbb{A}, O, tr\rangle$ *where*

- $\langle S, \Sigma\rangle$ *is a unary many-sorted signature closed under composition;*
- $\mathbb{A}$ *is a* $\langle S, \Sigma\rangle$*-algebra, with carriers* $\{A_s\}_{s\in S}$*;*
- O *is a set of observations;*
- $tr \subseteq |\mathbb{A}| \times O \times |\mathbb{A}|$ *is a labelled transition relation.*

Intuitively, sorts represent interfaces and the operations in Σ, interpreted in $\mathbb{A}$, if applied to a state represent an interaction of that state with the environment.

The associated notion of bisimilarity is *saturated bisimilarity*, which is the coarsest bisimulation closed under the operations in Σ.

Definition 45. *Let* $\langle\langle S, \Sigma\rangle, \mathbb{A}, O, tr\rangle$ *be a context interactive system. Then an indexed family of relations* $\mathcal{R} = \{R_s\}_{s\in S}$*, with* $R_s \subseteq A_s \times A_s$*, is a* saturated bisimulation *if and only if* $\forall s, s', t \in S, \forall\sigma \in \Sigma_{s,s'}$*, whenever* $(p_s, q_s) \in R_s$*:*

$$\sigma^{\mathbb{A}}(p_s) \xrightarrow{o} p'_t \implies \sigma^{\mathbb{A}}(q_s) \xrightarrow{o} q'_t \wedge (p'_t, q'_t) \in R_t \; .$$

We write $p_s \sim^S_s q_s$ *if and only if there is a saturated bisimulation* R *such that* $(p_s, q_s) \in R_s$*.*

Deriving a ISLTS $islts_{\mathcal{I}}$ for saturated bisimilarity from a CIS $\mathcal{I}$ is straightforward: if we let the set of types $\mathcal{T}_{\mathcal{I}}$ be S and that of operations $\mathcal{O}_{\mathcal{I}}$ be Σ, then $islts_{\mathcal{I}}$ is the ISLTS on $\{A_s\}_{s \in \mathcal{T}_{\mathcal{I}}}$, $\{\sigma^{\mathbb{A}}\}_{\sigma \in \mathcal{O}_{\mathcal{I}}}$, and O whose action-labelled transitions are those of tr. The equivalence of bisimilarities is obvious.

Proposition 46. $p_s \sim^S_s q_s$ *if and only if* $p_s \sim^I_s q_s$ *with respect to* $islts_{\mathcal{I}}$*.*

Let us consider the CIS for π-calculus $\mathcal{I}_\pi = \langle\langle S_\pi, \Sigma_\pi\rangle, \Pi, O_\pi, tr_\pi\rangle$, where $\langle S_\pi, \Sigma_\pi\rangle$ is the one-sorted signature defined as follows:

sort Agent
operations

$$\begin{aligned} \mathbf{0} &: \to Agent & p \mid \_ &: Agent \to Agent \\ \alpha.\_ &: Agent \to Agent & (\boldsymbol{\nu} x)\_ &: Agent \to Agent \\ p + \_ &: Agent \to Agent & [x = y]\_ &: Agent \to Agent \end{aligned}$$

where α is a prefix, $x, y \in \mathcal{N}$ and $p : Agent$, Π is the initial $\langle S_\pi, \Sigma_\pi\rangle$-algebra, O_π is the set $\{\tau, xy, \overline{x}y, \overline{x}(y) \mid x, y \in \mathcal{N}\}$ and tr_π is the early π-calculus transition relation. Then the relation $\sim^I$ on the resulting ISLTS is the largest congruence in early bisimilarity (analogously for late bisimilarity).

Finally, let us consider the coalgebraic point of view. Since we cannot make assumptions about the representability of tr as a natural transformation, we have to employ the behavioural functor of definition 38, with $\mathbf{L}$ the constant presheaf sending each $s \in |\mathbf{C}|$ to O. Notice that, in this case, the carrier presheaf is just the algebra $\mathbb{A}$ in functorial form.

5.5 The polyadic case

In the examples seen so far the family $\{f_\sigma\}$ contains only monadic functions, but this is not a requirement. For instance, we might consider a signature for π-calculus with $\_ \mid \_ : Agent \times Agent \to Agent$ as parallel composition, which certainly is more natural than considering a distinct parallel composition operator $\_ \mid p$ for each agent p. Now we show how to treat the polyadic case, slightly modifying the general theory if needed.

Let us consider a ILTS $\langle S, \longrightarrow\rangle$ on $\{S_a\}_{a\in\mathcal{T}}$ and L, and a set of basic operation symbols $\mathcal{O}$ of the form $\sigma : a_1 \times \cdots \times a_n \to a_{n+1}$, where $a_1, \ldots, a_{n+1} \in \mathcal{T}$, interpreted by the family $\{op_\sigma\}$.

To construct the ISLTS we consider the set of types $\mathcal{T}_\times = \bigcup_{n\in\{1,2,\ldots\}} \mathcal{T}^n$ and that of operations $\mathcal{O}_\times$ generated by the following rules

$$\frac{\sigma \in \mathcal{O}}{\sigma \in \mathcal{O}_\times} \qquad \frac{\sigma_1 : a_1 \to b_1, \ldots, \sigma_n : a_n \to b_n \in \mathcal{O}}{\sigma_1 \times \cdots \times \sigma_n : a_1 \times \cdots \times a_n \to b_1 \times \cdots \times b_n \in \mathcal{O}_\times}$$

plus identities, closure under composition and associativity. Then we define a new family of states $\{S'_a\}_{a\in\mathcal{T}_\times}$:

$$S'_{a_1\times\cdots\times a_n} = S_{a_1} \times \cdots \times S_{a_n} \tag{5}$$

and a family of functions $\{f_\sigma\}_{\sigma\in\mathcal{O}_\times}$ such that: $f_\sigma = op_\sigma$ if $\sigma \in \mathcal{O}$, f_{id_a} is the identity function, for each $a \in \mathcal{T}_\times$, and

$$f_{\sigma_1\times\cdots\times\sigma_n} : (s_1, \ldots, s_n) \mapsto (f_{\sigma_1}(s_1), \ldots, f_{\sigma_n}(s_n)) \ . \tag{6}$$

In addition, $\{f_\sigma\}_{\sigma\in\mathcal{O}_\times}$ is closed under composition.

Then the ISLTS corresponding to $\langle S, \longrightarrow\rangle$ is $\langle\coprod_{a\in\mathcal{T}_\times} S'_a, \longrightarrow_\times\rangle$, where the transition relation is $\longrightarrow$ with the additional transitions:

$$(s^{(1)}_{a_1}, \ldots, s^{(n)}_{a_n}) \xrightarrow{\sigma} f_\sigma(s^{(1)}_{a_1}, \ldots, s^{(n)}_{a_n}) \ ,$$

for each $\sigma \in \mathcal{O}_\times$ with domain $a_1 \times \cdots \times a_n$. Notice that we had to use a larger family of states, unlike what done in section 3.2, due to the polyadicity of the operations.

To coalgebrically represent such ISLTSs we employ the Lawvere theory $\mathcal{L}(\mathcal{T}, \mathcal{O})$ (i.e. the Lawvere theory modeling the polyadic signature $\langle\mathcal{T}, \mathcal{O}\rangle$) as the presheaves domain category. However we cannot consider the whole $\mathbf{Set}^{\mathcal{L}(\mathcal{T},\mathcal{O})}$, because the functor $\mathbf{P} : \mathcal{L}(\mathcal{T}, \mathcal{O}) \to \mathbf{Set}$ induced by $\{S'_a\}_{a\in\mathcal{T}_\times}$ and $\{f_\sigma\}_{\sigma\in\mathcal{O}_\times}$ must satisfy $\mathbf{P}(a_1 \times \cdots \times a_n) = \mathbf{P}a_1 \times \cdots \times \mathbf{P}a_n$ and $\mathbf{P}(\langle\sigma_1, \ldots, \sigma_n\rangle) = \langle\mathbf{P}\sigma_1, \ldots, \mathbf{P}\sigma_n\rangle$, which are restatements of (5) and (6) respectively, i.e. it must be cartesian. Thus the correct base category for coalgebras is $\mathbf{Mod}(\mathcal{L}(\mathcal{T}, \mathcal{O}))$.

6 Conclusions and Future Work

The theory described in this thesis allows to pass from set-based transition systems, provided that bisimilarity is a congruence, to presheaf-based coalgebras mantaining the bisimulations of the original transition system.

We think this could have interesting applications, for instance in *partition refinement algorithms* [19,20], widely used in tools for checking bisimulations equivalences and for computing minimal realizations. In fact some of these algorithms operate on coalgebras and in particular, for coalgebras on **Set**, they exploit the fact that the minimal realization of a coalgebra can be reached by a "path" made of surjective cohomomorphisms, incrementally quotienting the space state according to their kernels; thanks to the fact that kernels are bisimulations, the obtained equivalence classes contain states with the same behaviour. Since we proved that there is an analogous notion of kernel also for coalgebras over presheaves, such algorithms should also be applicable effortlessly in a presheaf setting.

As a future development, we would like to investigate further examples:

- The *explicit fusion calculus* [21], a version of π-calculus with additional *explicit fusion agents* $x = y$, where $x, y \in \mathcal{N}$. Fusion agents represent identifications of names which are activated by parallel composition: for any agent p, $p \mid x = y$ allows p to use x and y interchangeably. They induce an equivalence relation on names for each agent p, which contains (x, y) if and only if the fusion $x = y$ is active (hence $\alpha.p$ has empty equivalence relation, for any prefix α). In this case resources are not just names, but equivalence relations between (free) names, which are modified during the computation as follows: a transition $p \xrightarrow{l} q$ can either activate fusions which are not active in p, e.g. if l is a prefix, or turn a bound name into a free name, if l is a bound output label, which in q may already be involved in a fusion; in the former case the equivalence classes of p are enlarged, in the latter case a new element is added to the relation: as a new singleton equivalence class if it does not appear in any of q's active fusions, otherwise it is added to an existing class. The categorical representation might be similar to that given in [22], where the presheaves index category is $\mathbb{E}$, incorporating equivalence relations and equivalence preserving morphisms, and presheaves $\mathbb{E} \to \mathbf{Set}$ are used to associate each equivalence relation to the agents for which the relation hold. To apply our theory, suitable endofunctors on $\mathbb{E}$ has to defined, modeling the just described operations on equivalence relations.

- *CC-Pi* [23], a version of π-calculus with constraints. Constraints are defined as elements of an algebraic structure, called *named c-semirings*, that equip them with a mechanism of constraint combination; the extended syntax provides the prefixes tell c, ask c, check c and retract c for adding, testing and removing constraints. It is reasonable to think of constraints as resources that can be allocated and deallocated, and thus as objects of a category with endofunctors for allocation and deallocation. This observation might be the key idea for a categorical model.

As a step further, an ambitious goal would be organizing our results into an effective meta-theory.

Acknowledgments Ringrazio innanzitutto il prof. Ugo Montanari, sempre disponibile e prodigo di consigli preziosi. Ringrazio la mia famiglia e ringrazio Sabrina, che come al solito mi ha offerto appoggio incondizionato, e infine ringrazio tutti i miei amici. Ringraziamenti speciali a Robertino, che mi ha fatto da meticoloso correttore di bozze, a Daniele che ha fornito i mezzi tipografici, a Giancarmelo e Michele, che mi hanno dato ottimi consigli per la presentazione; ringrazio Michele anche per avermi offerto occasioni di svago geometrico.

References

1. Lawvere, F.W.: Functorial Semantics of Algebraic Theories and Some Algebraic Problems in the context of Functorial Semantics of Algebraic Theories. PhD thesis, Columbia University (1963)
2. Lambek, J.: Cartesian closed categories and typed λ-calculi. In: Combinators and Functional Programming Languages. (1985) 136–175
3. Tennent, R.D.: Functor - category semantics of programming languages and logics. In: CTCS. (1985) 206–224
4. Goguen, J.A., Burstall, R.M.: Institutions: abstract model theory for specification and programming. J. ACM **39**(1) (1992) 95–146
5. Cattani, G.L., Stark, I., Winskel, G.: Presheaf models for the π-calculus. In: Category Theory and Computer Science: Proceedings of the 7th International Conference CTCS '97. Number 1290 in Lecture Notes in Computer Science, Springer-Verlag (1997) 106–126
6. Fiore, M., Turi, D.: Semantics of name and value passing. In: Proc. 16th LICS, IEEE Computer Society Press (2001) 93–104
7. Fiore, M.P., Moggi, E., Sangiorgi, D.: A fully abstract model for the π-calculus. Information and Computation **179**(1) (November 2002) 76–117
8. Turi, D., Plotkin, G.: Towards a mathematical operational semantics. In: Proc. 12^{th} LICS Conf., IEEE, Computer Society Press (1997) 280–291
9. Sangiorgi, D.: A theory of bisimulation for the pi-calculus. Acta Inf. **33**(1) (1996) 69–97
10. Bonchi, F., Montanari, U.: Symbolic semantics revisited. In Amadio, R.M., ed.: FoSSaCS. Volume 4962 of Lecture Notes in Computer Science., Springer (2008) 395–412
11. Ghani, N., Yemane, K., Victor, B.: Relationally staged computations in calculi of mobile processes. Electr. Notes Theor. Comput. Sci. **106** (2004) 105–120
12. Barr, M., Wells, C.: Category Theory for Computing Science. Prentice Hall (1995)
13. Rutten, J.J.M.M.: Universal coalgebra: a theory of systems. Theor. Comput. Sci. **249**(1) (2000) 3–80
14. Adamek, J.: Introduction to coalgebra. Mathematical Structures in Computer Science (15) (2005) 409–432
15. Montanari, U., Sassone, V.: Dynamic congruence vs. progressing bisimulation for ccs. Fundam. Inform. **16**(1) (1992) 171–199
16. Parrow, J. In: An Introduction to the pi-calculus. (2001) 479–543
17. de Bruijn, N.G.: Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. Indagationes Mathematicae (Proceedings) **75**(5) (1972) 381–392
18. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes pt.2. Information and Computation **100**(1) (September 1992) 41–77
19. Kanellakis, P.C., Smolka, S.A.: Ccs expressions, finite state processes, and three problems of equivalence. In: PODC '83: Proceedings of the second annual ACM symposium on Principles of distributed computing, New York, NY, USA, ACM (1983) 228–240
20. Paige, R., Tarjan, R.E.: Three partition refinement algorithms. SIAM J. Comput. **16**(6) (1987) 973–989
21. Wischik, L.: Explicit fusions. In: Proc. MFCS 2000. LNCS 1893, Springer-Verlag (2000) 373–382
22. Bonchi, F., Buscemi, M.G., Ciancia, V., Gadducci, F.: A category of explicit fusions. (2008) 544–562
23. Buscemi, M.G., Montanari, U.: Cc-pi: A constraint-based language for specifying service level agreements. In: ESOP. (2007) 18–32