# Modeling Fresh Names in the $\pi$-calculus using Abstractions ⋆

Roberto Bruni [1]

*Dipartimento di Informatica, Università di Pisa,*
*Via F. Buonarroti 2, 56127 Pisa, ITALY.*
*e-mail:* bruni@di.unipi.it

Furio Honsell Marina Lenisa Marino Miculan

*Dipartimento di Matematica e Informatica, Università di Udine,*
*Via delle Scienze 206, 33100 Udine, ITALY.*
*e-mail:* honsell,lenisa,miculan@dimi.uniud.it

**Abstract**

In this paper, we model fresh names in the $\pi$-calculus using *abstractions* with respect to a new binding operator $\theta$. Both the theory and the metatheory of the $\pi$-calculus benefit from this simple extension. The operational semantics of this new calculus is *finitely branching*. Bisimulation can be given without mentioning any constraint on names, thus allowing for a straightforward definition of a coalgebraic semantics, within a category of coalgebras over *permutation algebras*. Following previous work by Montanari and Pistore, we present also a *finite* representation for *finitary* processes and a finite state verification procedure for bisimilarity, based on the new notion of $\theta$*-automaton*.

## 1 Introduction

The $\pi$-calculus [13,18] is a process calculus which provides a conceptual framework for understanding mobility via name passing. Processes can communicate in a network whose topology can change dynamically by passing, possibly local, channel names. As for any other foundational calculus, we need strong mathematical tools for expressing mobile systems and reasoning about their behaviours. However, due to the peculiar behaviour of mobile processes and

⋆ Research supported by the EU projects IST-2001-33477 DART, IST-2001-32747 AGILE.
[1] Partially supported by an Italian CNR fellowship for research on Information Sciences and Technologies and by the CS Department of the University of Illinois at Urbana-Champaign.

names, the well-known tools and techniques which have been successful for CCS-like languages cannot be straightforwardly extended to the $\pi$-calculus.

At the syntactic level, we have the problematic issue of binders and scope of local names. At the operational semantic level, we have the issue of ensuring freshness conditions for names. At the model-theoretic level, we have the issue of providing a coalgebraic (final) semantics. Finally, from the practical point of view, a finite representation for finitary processes is desirable [9,11,14].

All the above issues have been considered in many previous papers. Often, reformulations of the same calculus are introduced, in order to cope with these problematic issues. The question is *how* to present "best" the calculus, in order to achieve the mathematical structure we need for reasoning on its core computational aspects. The answer to this question depends on the metalogical formalism in which we define the calculus. Recently, for the $\pi$-calculus many reformulations, in different metalogics, have been presented. A first-order, de Bruijn-like approach is adopted in [14,3], where processes can be equipped with explicit *permutations* of names. A second-order approach, based on Higher-Order Abstract Syntax, is adopted in [9,10,12], where most issues about freshness of names are simplified, taking advantage of the metalogic notion of capture-avoiding substitution. A somehow mid-way approach is in [6], where the reformulation is given in the logic of Frænkel-Mostowski models of first-order set theory (i.e., sets with atoms and permutations).

In this paper, we provide yet another formulation of the $\pi$-calculus with the aim of expressing generation of fresh names at every level (syntactic, semantic and implementative), and still keeping the metalogical overhead as low as possible. In fact, the calculus that we will present in Section 3, is just a conservative extension of the ordinary $\pi$-calculus with a new unary binding operator $\theta$; for this reason, it is called the *$\pi\theta$-calculus*. This extension is suggested by the higher order presentations of the $\pi$-calculus as in [9,10,12,5].

The operator $\theta$ allows to explain "fresh" names as "locally $\theta$-bound" names. A transition which needs a fresh name is rendered as a transition to a *$\theta$-abstracted process*, i.e. where the fresh name is $\theta$-bound. As we will see, many aspects of the treatment of the theory and metatheory of the $\pi$-calculus will benefit from this simple extension. Differently from the $\pi$-calculus, in the $\pi\theta$-calculus also actions are taken up-to $\alpha$-conversion, yielding a *finitely branching* semantics w.r.t. fresh names. For example, while the $\pi$-process $(\nu y)\bar{x}y.P$ can evolve via $(\nu y)\bar{x}y.P \xrightarrow{\bar{x}(z)} P\{z/y\}$ for any fresh name $z$ (and thus it is infinitely branching), the $\pi\theta$-process has just one move $(\nu y)\bar{x}y.P \xrightarrow{(\theta y)\bar{x}y} (\theta y)P$.

Bisimulation on the $\pi\theta$-calculus can be given without any constraint on bound names in labels. This allows for a direct re-use of the techniques in [1,2,16] for defining a coalgebraic semantics. In particular, the semantics of a process is finitely branching without being parametrized by the set of names of possible partners, as was the case in [9]. Moreover, our semantics is *finitary*, in the sense that any *finitary* process, i.e. with a bound degree of parallelism, gives rise to a finite set of descendant processes, up-to *vacuous* bound names (i.e. abstracted names which do not appear in the body of the process) and ordinary structural congruence.

In Section 4 we define the coalgebraic semantics within a category $Alg_\pi$ of *permutation algebras*, following [14]. The algebra structure we consider is induced by a countable family of unary permutation operators $\{\rho_i\}_{i\in\omega}$. The fact that our coalgebraic semantics is an algebra homomorphism allows us to derive interesting properties on *active names* of processes in the final model.

In order to get a truly finite representation for finitary processes, in Section 5 we introduce the notion of *$\theta$-automaton*. This is the counterpart, in our second-order setting, of the notion of *History Dependent Automaton* of [14]. We associate to each $\pi\theta$-process a $\theta$-automaton which is finite in case the original process is finitary. States of $\theta$-automata are given by collapsing the *orbits* of processes under the action of vacuous $\theta$-operators. We introduce a notion of bisimulation on the states of $\theta$-automata. Bisimilarity between $\pi$-calculus processes can be (finitely) verified by checking the bisimilarity relation on the corresponding $\theta$-automata.

Conclusions, related work, and directions for future work are in Section 6.

## 2 The $\pi$-calculus

In this section, we introduce briefly the $\pi$-calculus; see [13,15] for more details. In particular, we introduce the syntax of the language, the *early* operational semantics, and the equivalence relation of *early* bisimilarity.

In the $\pi$-calculus there are only two primitive entities: *names* and *processes* (or *agents*). Let $\mathcal{N}$ be an infinite set of names, ranged over by $x$, $y$. The set of processes $\mathcal{P}$, ranged over by $P$, $Q$, are closed terms (w.r.t. process variables $Z$) defined by the abstract syntax:

$$P ::= 0 \mid \bar{x}y.P \mid x(y).P \mid \tau.P \mid (\nu x)P \mid Z \mid \mathrm{rec}\, Z.P \mid P_1|P_2 \mid [x=y]P$$

where the bound process variable $Z$ must be guarded in $\mathrm{rec}\, Z.P$. The operators are listed in decreasing order of precedence. The *input prefix* operator $x(y)$ and the *restriction* operator $(\nu y)$ bind the occurrences of $y$ in $x(y).P$ and $(\nu y)P$ respectively. Thus, for each process $P$ we can define the sets of its *free names* $fn(P)$, *bound names* $bn(P)$ and *names* $n(P) \triangleq fn(P) \cup bn(P)$. Processes are taken up-to $\alpha$-equivalence, which is defined as expected. Capture-avoiding substitution of a single name $y$ in place of $x$ in $P$ is denoted by $P\{y/x\}$.

We denote by $\mathcal{P}_X$, where $X$ is a finite set of names, the subset of $\pi$-calculus processes whose free names are in $X$.

There is a plethora of slightly different labeled transition systems for the operational semantics of the $\pi$-calculus, see e.g. [13,15,18]. Here, we present the original one for early operational semantics [13]: the relation $\xrightarrow{\mu}$ is the smallest relation over processes satisfying the rules in Figure 1. (The right

$$\frac{-}{x(z).P \xrightarrow{xy} P\{y/z\}} \text{ (IN)} \qquad \frac{-}{\overline{x}y.P \xrightarrow{\overline{x}y} P} \text{ (OUT)}$$

$$\frac{-}{\tau.P \xrightarrow{\tau} P} \text{ (TAU)} \qquad \frac{P \xrightarrow{\overline{x}y} P'}{(\nu y)P \xrightarrow{\overline{x}(y)} P'} \; y \neq x \text{ (OPEN)}$$

$$\frac{P \xrightarrow{xy} P' \quad Q \xrightarrow{\overline{x}(y)} Q'}{P|Q \xrightarrow{\tau} (\nu y)(P'|Q')} \; y \notin fn(P) \text{ (CLOSE}_l\text{)} \qquad \frac{P \xrightarrow{\mu} P'}{[x = x]P \xrightarrow{\mu} P'} \text{ (MATCH)}$$

$$\frac{P \xrightarrow{\mu} P'}{(\nu y)P \xrightarrow{\mu} (\nu y)P'} \; y \notin n(\mu) \text{ (RES)} \qquad \frac{P\{\mathrm{rec}\, Z.P/Z\} \xrightarrow{\mu} P'}{\mathrm{rec}\, Z.P \xrightarrow{\mu} P'} \text{ (UNFOLD)}$$

$$\frac{P \xrightarrow{\mu} P'}{P|Q \xrightarrow{\mu} P'|Q} \; bn(\mu) \cap fn(Q) = \emptyset \text{ (PAR}_l\text{)} \qquad \frac{P \xrightarrow{xy} P' \quad Q \xrightarrow{\overline{x}y} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \text{ (COM}_l\text{)}$$

Fig. 1. Early Operational semantics of the $\pi$-calculus.

versions of rules PAR, COM and CLOSE have been omitted.)

The early operational semantics exploits four actions, defined by the syntax $(\mathcal{L} \ni)\ \mu ::= \tau \mid xy \mid \bar{x}y \mid \bar{x}(z)$. Their intuitive meaning is the following:

**silent action:** $P \xrightarrow{\tau} Q$ means that $P$ can reduce itself to $Q$ without interacting with other processes;

**free output:** $P \xrightarrow{\bar{x}y} Q$ means that $P$ can reduce itself to $Q$ emitting the name $y$ on the channel $x$;

**free input:** $P \xrightarrow{xy} Q$ means that $P$ can receive from the channel $x$ the name $y$ and then evolve into $Q$;

**bound output:** $P \xrightarrow{\overline{x}(z)} Q$ means that $P$ can evolve into $Q$ emitting on the channel $x$ a name $z$, which is bound in $P$ (but not in $Q$); only upon synchronization, $z$ will be shared with the receiving agents and restricted again.

The functions $fn(\cdot)$ and $bn(\cdot)$ are extended to actions, by putting $fn(\bar{x}(z)) = \{x\}$, $fn(xy) = fn(\bar{x}y) = \{x, y\}$, $fn(\tau) = bn(\tau) = bn(xy) = bn(\bar{x}y) = \emptyset$, $bn(\bar{x}(z)) = \{z\}$. As usual, $n(\mu) \triangleq fn(\mu) \cup bn(\mu)$.

The $\tau$ and free input and free output actions are called *free*, the remaining ones are called *bound*. Note that actions are *not* taken up-to $\alpha$-equivalence.

**Definition 2.1 (Early Bisimilarity)** *A symmetric relation $\mathcal{R}$ over $\pi$-calculus processes is an* early bisimulation *iff, for all processes $P, Q$, if $P \mathrel{\mathcal{R}} Q$ then*

*for each $P \xrightarrow{\mu} P'$ with $bn(\mu) \cap fn(P, Q) = \emptyset$ then there exists $Q'$ such that $Q \xrightarrow{\mu} Q'$ and $P' \mathrel{\mathcal{R}} Q'$.*

*The* early bisimilarity $\sim$ *is the greatest early bisimulation.*

## 3 The $\pi\theta$-calculus

In this section, we introduce the *$\pi\theta$-calculus,* an extension of the $\pi$-calculus, where processes can be prefixed possibly by a finite sequence of the new binding operator $\theta$. This new operator can be used to take care of the allocation of fresh names; essentially, it allows to model a *fresh* name using a *bound* name.

### *3.1 Syntax*

The sets of $\pi\theta$-processes $\mathcal{P}^\theta$ and $\pi\theta$-actions $\mathcal{L}^\theta$ are defined as follows:

$$\mathcal{P}^\theta \triangleq \{(\theta x_1)\dots(\theta x_n)P \mid P \in \mathcal{P},\ x_1,\dots,x_n \in \mathcal{N},\ n \geq 0\}$$
$$\mathcal{L}^\theta \triangleq \{(\theta x_1)\dots(\theta x_n)\mu \mid \mu \in \mathcal{L}_f \cup \mathcal{L}_b,\ x_1,\dots,x_n \in \mathcal{N},\ n \geq 0\}$$

where $\mathcal{P}$ is the set of $\pi$-calculus processes, $\mathcal{L}_f$ is the set of *free actions*, i.e., $\tau$, free input and free output, and $\mathcal{L}_b$ is the set of *bound actions*, i.e. bound input, $x(y)$, and bound output, $\bar{x}(y)$, where $x(\ )$ and $\bar{x}(\ )$ bind $y$. By abuse of notation, $P, Q$ and $\mu$ will range also over $\mathcal{P}^\theta$ and $\mathcal{L}^\theta$, respectively. We will use the abbreviations $(\theta\boldsymbol{x})P$ and $(\theta\boldsymbol{x})\mu$ for the process $(\theta x_1)\dots(\theta x_n)P$, and for the label $(\theta x_1)\dots(\theta x_n)\mu$, respectively, where $P$ and $\mu$ are $\theta$-free. The operator $\theta$ binds the occurrences of $x_1\dots x_n$ in $(\theta\boldsymbol{x})P$ and in $(\theta\boldsymbol{x})\mu$. *Both* processes and labels are taken up-to $\alpha$-equivalence; hence, without loss of generality, $x_1,\dots,x_n$ in $\boldsymbol{x}$ can always be assumed to be all distinct (e.g., $(\theta xx)P$ is the same as $(\theta xy)P\{y/x\}$).

The process $(\theta x)P$ can be viewed as the representation of a process abstraction obtained by instantiating $P$ with a fresh name; the name which has to be fresh remains bound in $P$ so that its freshness is guaranteed implicitly. In a sense, $\theta$-abstractions resemble the $\lambda$-*abstractions* of the alternative presentations of the $\pi$-calculus in, e.g., [15, §5.5]. However, our aims are different; in fact, we do not have a notion of "application" (i.e., *concretion*).

For $X$ a finite set of names, we denote by $\mathcal{P}^\theta_X$ and $\mathcal{L}^\theta_X$, the sets of $\pi\theta$-processes and $\pi\theta$-labels whose free names are in $X$, respectively. The set of *closed* $\pi\theta$-processes is $\mathcal{P}^\theta_\emptyset$. The occurrence $(\theta x_i)$ in the process $(\theta x_1\dots x_n)P$ is *vacuous* if $x_i \notin \mathit{fn}(P)$.

### *3.2 Operational Semantics*

The operational semantics of the $\pi\theta$-calculus is given by a family of relations. For $X$ a finite set of names, the relation

$$\twoheadrightarrow_X \subseteq \mathcal{P}^\theta_X \times \mathcal{L}^\theta_X \times \mathcal{P}^\theta_X$$

is defined as the smallest relation satisfying the rules in Figure 2. By definition, for any transition of the form $(\theta\boldsymbol{x})P \overset{(\theta\boldsymbol{y})\mu}{\twoheadrightarrow}_X (\theta\boldsymbol{z})Q$, we have $\mathit{fn}((\theta\boldsymbol{y})\mu, (\theta\boldsymbol{z})Q) \subseteq \mathit{fn}((\theta\boldsymbol{x})P)$. Hence, if $(\theta\boldsymbol{x})P$ is closed, then $X$ can be set to $\emptyset$.

$$\frac{-}{\tau.P \overset{\tau}{\twoheadrightarrow}_X P}\ \text{(TAU)} \qquad \frac{P \overset{\overline{x}y}{\twoheadrightarrow}_{X \uplus \{y\}} P'}{(\nu y)P \overset{\overline{x}(y)}{\twoheadrightarrow}_X (\theta y)P'}\ \text{(OPEN)}$$

$$\frac{-}{x(y).P \overset{xz}{\twoheadrightarrow}_X P\{z/y\}}\ z \in X \ \text{(IN)} \qquad \frac{P\{\mathrm{rec}\, Z.P/Z\} \overset{\mu}{\twoheadrightarrow}_X P'}{\mathrm{rec}\, Z.P \overset{\mu}{\twoheadrightarrow}_X P'}\ \text{(UNFOLD)}$$

$$\frac{-}{x(y).P \overset{x(y)}{\twoheadrightarrow}_X (\theta y)P}\ (\mathrm{IN}^\theta) \qquad \frac{P \overset{\mu}{\twoheadrightarrow}_X P'}{[x = x]P \overset{\mu}{\twoheadrightarrow}_X P'}\ \text{(MATCH)}$$

$$\frac{-}{\overline{x}y.P \overset{\overline{x}y}{\twoheadrightarrow}_X P}\ \text{(OUT)} \qquad \frac{P \overset{\mu}{\twoheadrightarrow}_{X \uplus \{y\}} P'}{(\nu y)P \overset{\mu}{\twoheadrightarrow}_X (\nu y)P'}\ y \notin fn(\mu),\ \mu \in \mathcal{L}_f \ \text{(RES)}$$

$$\frac{P \overset{\mu}{\twoheadrightarrow}_X P'}{P|Q \overset{\mu}{\twoheadrightarrow}_X P'|Q}\ \mu \in \mathcal{L}_f \ (\mathrm{PAR}_l) \qquad \frac{P \overset{\mu}{\twoheadrightarrow}_{X \uplus \{y\}} (\theta z)P'}{(\nu y)P \overset{\mu}{\twoheadrightarrow}_X (\theta z)(\nu y)P'}\ y \notin fn(\mu) \ (\mathrm{RES}^\theta)$$

$$\frac{P \overset{\mu}{\twoheadrightarrow}_X (\theta x)P'}{P|Q \overset{\mu}{\twoheadrightarrow}_X (\theta x)P'|Q}\ (\mathrm{PAR}^\theta_l) \qquad \frac{P \overset{x(y)}{\twoheadrightarrow}_X (\theta y)P' \quad Q \overset{\overline{x}(y)}{\twoheadrightarrow}_X (\theta y)Q'}{P|Q \overset{\tau}{\twoheadrightarrow}_X (\nu y)P'|Q'}\ (\mathrm{CLOSE}_l)$$

$$\frac{P \overset{xy}{\twoheadrightarrow}_X P' \quad Q \overset{\overline{x}y}{\twoheadrightarrow}_X Q'}{P|Q \overset{\tau}{\twoheadrightarrow}_X P'|Q'}\ (\mathrm{COM}_l) \qquad \frac{P \overset{\mu}{\twoheadrightarrow}_{X \uplus \{x\}} Q}{(\theta x)P \overset{(\theta x)\mu}{\twoheadrightarrow}_X (\theta x)Q}\ \text{(THETA)}$$

Fig. 2. Early Operational semantics of the $\pi\theta$-calculus.

Note that there are two input rules, IN and $\mathrm{IN}^\theta$. In rule IN the bound name is instantiated with a "previously known" name $z$ in $X$. Rule $\mathrm{IN}^\theta$ takes care of the instantiation with a fresh name, by creating a new $\theta$-bound name $y$. In this way, all $\pi$-calculus input transitions differing by the choice of the new name are collapsed (by $\alpha$-rule) in a single transition, and the $\pi\theta$-system becomes finitely branching. As in rule $\mathrm{IN}^\theta$, also in rule OPEN, the allocation of a fresh name is delegated to the constructor $\theta$. The rules PAR and RES are duplicated, to take into account the case in which a $\theta$-bound name appears in the target process. [2]

We remark that $(\nu x)P$ and $(\theta x)P$ behave differently, in general. Namely, rules RES and OPEN do not allow for actions whose subject is exactly $x$, while the process $(\theta x)P$ could make any transition under $\theta$.

The following lemma clarifies the rôle of $\theta$, and it is the counterpart of [13, Lemma 3] for the $\pi\theta$-calculus.

**Lemma 3.1** *For all $X$ finite, for all $(\theta \boldsymbol{x})P \in \mathcal{P}^\theta_X$:*

*i) for all $(\theta \boldsymbol{x})Q, (\theta \boldsymbol{x})\mu$:* $\quad (\theta \boldsymbol{x})P \overset{(\theta \boldsymbol{x})\mu}{\twoheadrightarrow}_X (\theta \boldsymbol{x})Q$ *iff* $P \overset{\mu}{\twoheadrightarrow}_{X \cup \{\boldsymbol{x}\}} Q$;

*ii) for all $(\theta \boldsymbol{x} y)Q, (\theta \boldsymbol{x})\mu$:* $\quad (\theta \boldsymbol{x})P \overset{(\theta \boldsymbol{x})\mu}{\twoheadrightarrow}_X (\theta \boldsymbol{x} y)Q$ *iff* $P \overset{\mu}{\twoheadrightarrow}_{X \cup \{\boldsymbol{x}\}} (\theta y)Q$.

We can draw a precise correspondence between $\pi$- and $\pi\theta$-derivations:

[2] Strictly speaking, the side conditions "$z \in X$" of rule IN and "$y \notin fn(\mu)$" of rules RES and $\mathrm{RES}^\theta$ are redundant, because they are always ensured by the type of $\twoheadrightarrow_X$.

**Proposition 3.2** *For all $X$ finite, for all $\boldsymbol{z}$ not in $X$, and $P \in \mathcal{P}_{X \cup \{\boldsymbol{z}\}}$, $x, y \in \mathcal{N}$:*

- $P \xrightarrow{\tau} Q$ *iff* $(\theta \boldsymbol{z})P \xrightarrow{(\theta \boldsymbol{z})\tau}\!\!\!\!\rightarrow_X (\theta \boldsymbol{z})Q$;
- $P \xrightarrow{\bar{x}y} Q$ *iff* $(\theta \boldsymbol{z})P \xrightarrow{(\theta \boldsymbol{z})\bar{x}y}\!\!\!\!\rightarrow_X (\theta \boldsymbol{z})Q$;
- $P \xrightarrow{xy} Q$ *iff* $\Big($*if* $y \in X \cup \{\boldsymbol{z}\}$ *then* $(\theta \boldsymbol{z})P \xrightarrow{(\theta \boldsymbol{z})xy}\!\!\!\!\rightarrow_X (\theta \boldsymbol{z})Q$ *else* $(\theta \boldsymbol{z})P \xrightarrow{(\theta \boldsymbol{z})x(y)}\!\!\!\!\rightarrow_X (\theta \boldsymbol{z}y)Q\Big)$;
- $P \xrightarrow{\bar{x}(y)} Q$ *iff* $(\theta \boldsymbol{z})P \xrightarrow{(\theta \boldsymbol{z})\bar{x}(y)}\!\!\!\!\rightarrow_X (\theta \boldsymbol{z}y)Q$.

The proof of Proposition 3.2 is straightforward by mutual induction on the structure of derivations. In particular, when $\boldsymbol{z}$ is empty:

**Corollary 3.3** *For all $X$ finite, for all $P \in \mathcal{P}_X$, for all $x, y \in \mathcal{N}$:*

- $P \xrightarrow{\tau} Q$ *iff* $P \xrightarrow{\tau}\!\!\!\!\rightarrow_X Q$;
- $P \xrightarrow{\bar{x}y} Q$ *iff* $P \xrightarrow{\bar{x}y}\!\!\!\!\rightarrow_X Q$;
- $P \xrightarrow{xy} Q$ *iff* $\Big($*if* $y \in X$ *then* $P \xrightarrow{xy}\!\!\!\!\rightarrow_X Q$ *else* $P \xrightarrow{x(y)}\!\!\!\!\rightarrow_X (\theta y)Q\Big)$;
- $P \xrightarrow{\bar{x}(y)} Q$ *iff* $P \xrightarrow{\bar{x}(y)}\!\!\!\!\rightarrow_X (\theta y)Q$.

The relations $\longrightarrow\!\!\!\!\rightarrow_X$ can be seen as a family of coherent "approximations" of the usual early operational semantics. We can recover this semantics by taking the union of all approximations: for $\mu$ action of the $\pi\theta$-calculus, we define $\xrightarrow{\mu}\!\!\!\!\rightarrow \triangleq \bigcup_X \xrightarrow{\mu}\!\!\!\!\rightarrow_X$. Of course, we could drop safely the $X$ parameter and consider all the transition systems simultaneously, without any other consequence but that the operational semantics would then be finitely branching *only* w.r.t. fresh names. At the moment, each $\longrightarrow\!\!\!\!\rightarrow_X$ is *truly* finitely branching, because the names which can be chosen in the rule IN must belong to $X$, which is finite. We prefer to keep the $X$ parameter, in order to provide a sharper analysis of the system.

### *3.3 Bisimulation*

We can now introduce a notion of bisimulation on $\pi\theta$-processes, which provides an alternative characterization of early bisimilar processes.

**Definition 3.4 (Early $\theta$-bisimilarity)** *Let $X$ be a finite set of names. A symmetric relation $\mathcal{R}_X \subseteq \mathcal{P}^\theta_X \times \mathcal{P}^\theta_X$ is an* early $\theta$-bisimulation at stage $X$ *iff, for all $P, Q \in \mathcal{P}^\theta_X$ processes, $P \,\mathcal{R}_X\, Q$ implies:*

- *if $P \xrightarrow{\mu}\!\!\!\!\rightarrow_X P'$, then there exists $Q'$ such that $Q \xrightarrow{\mu}\!\!\!\!\rightarrow_X Q'$ and $P' \,\mathcal{R}_X\, Q'$.*

*The* early $\theta$-bisimilarity at stage $X$, *$\approx_X$, is the greatest early $\theta$-bisimulation at stage $X$. The* early $\theta$-bisimilarity *$\approx$ is defined as $\approx \triangleq \bigcup_X \approx_X$.*

Notice that the notion of early $\theta$-bisimilarity depends, for generic processes, on their free names. Again we could disregard $X$ completely. Anyway, for $\theta$-closed processes, any reference to names disappears altogether. What is more significant, however, is that for all (possibly open) processes, the side condition on the freshness of names necessary for bound output in Definition 2.1 disappears, being implicit in the fact that the new name is bound by $\theta$ in $P'$, $Q'$. The price to pay is that each time a fresh name is needed, an extra (possibly vacuous) $\theta$ is generated, and the set of processes reached during the evolution of a finitary process is finite, only *up-to vacuous $\theta$'s.*

**Example 3.5** *Let us consider the recursive process $P = \mathrm{rec}\, Z.(\nu y).\bar{x}y.Z \in \mathcal{P}_X$, where $X = \{x\}$. In the $\pi$-calculus, the process $P$ can evolve into itself, i.e. $P \xrightarrow{\bar{x}(y)} P \xrightarrow{\bar{x}(y)} \ldots$ while, in the $\pi\theta$-calculus, $P$ can evolve as follows:*

$$P \overset{\bar{x}(y_0)}{\twoheadrightarrow}_X (\theta y_0)P \overset{(\theta y_0)\bar{x}(y_1)}{\twoheadrightarrow}_X (\theta y_0 y_1)P \overset{(\theta y_0 y_1)\bar{x}(y_2)}{\twoheadrightarrow}_X (\theta y_0 y_1 y_2)P \twoheadrightarrow_X \ldots$$

*Notice that the states reached by $P$ after a finite number of transition steps differ by a finite number of vacuous $\theta$'s.*

The following lemma can be viewed as the "higher order" version of Lemma 6 of [13], and it is instrumental to prove Theorem 3.7 below.

**Lemma 3.6** *Let $(\theta\boldsymbol{x})P, (\theta\boldsymbol{x})Q \in \mathcal{P}^{\theta}_X$. Then $(\theta\boldsymbol{x})P \approx_X (\theta\boldsymbol{x})Q$ iff $P \approx_{X\cup\{\boldsymbol{x}\}} Q$.*

As a main correspondence result, $\theta$-bisimilarity is a conservative extension of usual bisimilarity:

**Theorem 3.7** *Let $P, Q \in \mathcal{P}$. Then $P \sim Q$ iff $P \approx Q$.*

**Proof.** Both directions are proved by coinduction, using Proposition 3.2.
($\Rightarrow$) We prove that the relation
$\mathcal{R} = \{((\theta x_1 \ldots x_n)P, (\theta x_1 \ldots x_n)Q \mid n \geq 0, x_1, \ldots, x_n \in \mathcal{N} \ \wedge\ P \sim Q\}$
is an early $\theta$-bisimulation at stage $X$, for $X \supseteq fn(P, Q)$.
($\Leftarrow$) Using Lemma 3.6, we prove that the relation $\mathcal{R} = \{(P, Q) \mid P \approx Q\}$ is an early bisimulation. □

**Remark 3.8** *In the light of Lemma 3.6, one could wonder whether it is possible to simplify our notion of early $\theta$-bisimulation, by getting rid of redundant vacuous $\theta$'s. This would allows us to overcome the problem highlighted by Example 3.5. But even if we restrict ourselves to processes whose occurrences of names are all active*[3]*, independent elimination of vacuous $\theta$'s is not safe, as we can see from the counterexample below.*

***Example 3.9*** *Let $P = (\nu x)\bar{w}x.(\nu y)\bar{x}y.\bar{u}x.0$, $Q = (\nu x)\bar{w}x.(\nu y)\bar{x}y.\bar{u}y.0$. Then $P$ and $Q$ are* not *early bisimilar, because the last action of $P$ consists in communicating the first extruded name, while $Q$ communicates the second extruded name. But $P$ and $Q$ turn out to be erroneously equated if we eliminate vacuous $\theta$'s. Namely, after two transition steps $P$ reduces to $(\theta xy)\bar{u}x.0$, and $Q$ reduces to $(\theta xy)\bar{u}y.0$, but, since $\theta y$ is vacuous in the first process, while $\theta x$ is vacuous in the latter, we reduce ourselves to considering the pair of processes $(\theta x)\bar{u}x$ and $(\theta y)\bar{u}y$, which turn out to be $\alpha$-equivalent and thus bisimilar.*

[3] An occurence of a name is *active* in $P$ if it appears in an action in the evolution of $P$.

# 4 Finitary Coalgebraic Semantics

In this section, we capitalize on the results of the previous sections, in order to give a coalgebraic description of early bisimilarity which is both *not* parametrized on sets of names as well as *finitary* (up-to structural congruence and vacuous bound names) for finitary processes. We focus on *closed* $\pi\theta$-processes, since, by Lemma 3.6, bisimilarity of (possibly open) $\pi\theta$-processes can always be reduced to bisimilarity of $\theta$-closed processes.

Recall that a $T$-coalgebra on a category $\mathbf{C}$ (e.g. $\mathcal{S}et$), where $T$ is an endofunctor, is a pair $(A, \alpha)$, where $A$ is an object of $\mathbf{C}$ and $\alpha : A \to T(A)$ is an arrow of $\mathbf{C}$. A $T$-coalgebra morphism $h : (A, \alpha) \to (B, \beta)$ is an arrow $h : A \to B$ of $\mathbf{C}$ (e.g., a function when $\mathbf{C} = \mathcal{S}et$), such that $\beta \circ h = T(h) \circ \alpha$.

According to the final semantics approach [1,2,16], the operational semantics of a calculus is represented as a *$T$-coalgebra* for a suitable endofunctor $T$. If the functor is "well-behaved," there exists a *final $T$-coalgebra*, say $\Omega = (\Omega, \alpha_\Omega)$. Moreover, for any $T$-coalgebra $(A, \alpha)$, the unique arrow $\mathcal{M} : (A, \alpha) \to (\Omega, \alpha_\Omega)$ induces an equivalence on $A$ which is the categorical counterpart of the ordinary notion of bisimulation: a *$T$-bisimulation* on $(A, \alpha)$ is a relation $\mathcal{R} \subseteq A \times A$ for which there exists an arrow $\gamma : \mathcal{R} \to T(\mathcal{R})$, such that the projections $\pi_1$, $\pi_2$ can be lifted to $T$-coalgebra morphisms from $(\mathcal{R}, \gamma)$ to $(A, \alpha)$.

In order to take advantage of the algebraic structure of $\theta$-operators and be able to capture the number of active names in processes, we work in a category of *structured* coalgebras. We consider coalgebras over the category $Alg_\pi$ of *permutation algebras* [14], which we recall next.

Let us consider the permutations of the set of natural numbers $\{1, 2, 3 \ldots\}$. The *kernel* of a permutation $\pi$ is the set $\ker(\pi) \triangleq \{i \mid \pi(i) \neq i\}$. We denote by $\mathit{perm}^{\mathit{fk}} \triangleq \{\pi \mid \ker(\pi) \text{ finite}\}$ the countable group of *finite-kernel* permutations.

**Definition 4.1 (permutation signature and algebras)** *The signature $\Sigma_\pi$ is given by the set of finite-kernel permutations, together with the axioms schemata $id(x) = x$ and $\pi_1(\pi_2(x)) = (\pi_1\pi_2)(x)$.*

*A* permutation algebra *$\mathcal{A} = (A, \{\widehat{\pi}_A\})$ is an algebra for $\Sigma_\pi$. A permutation morphism $\sigma : \mathcal{A} \to \mathcal{B}$ is an algebra morphism, i.e., a function $\sigma : A \to B$ such that $\sigma(\widehat{\pi}_A(x)) = \widehat{\pi}_B(\sigma(x))$. Finally, $Alg(\Sigma_\pi)$ (shortened as $Alg_\pi$) denotes the category of permutation algebras and their morphisms.*

The set $\mathcal{P}^\theta_\emptyset$ of closed $\pi\theta$-calculus processes can be endowed with a structure of a permutation algebra. For $\pi \in \mathit{perm}^{\mathit{fk}}$, and $n \in \omega$, let us define $\max_n \pi \triangleq \max\{\pi(i) \mid i = 1, \ldots, n\}$ (thus $\max_n \pi \geq n$). We define the *name reindexing* operator $\rho_\pi : \mathcal{P}^\theta_\emptyset \to \mathcal{P}^\theta_\emptyset$ as $\rho_\pi((\theta \boldsymbol{x})P) \triangleq (\theta x_1 \ldots x_{\max_n \pi})P[\pi]$, where $P[\pi]$ denotes the process obtained by the application of $\pi$ to the free names occurring in $P$. Let $\rho$ be the countable set of unary operators $\rho \triangleq \{\rho_\pi \mid \pi \in \mathit{perm}^{\mathit{fk}}\}$.

**Proposition 4.2** *The pair $(\mathcal{P}^\theta_\emptyset, \rho)$ is a permutation algebra.*

The operational semantics induces a structure of coalgebra on the permutation algebra of closed $\pi\theta$-processes for a functor similar to that of CCS:

**Definition 4.3** *Let $T : Alg_\pi \to Alg_\pi$ be the functor defined by the canonical extension to arrows of the function:*

$$T(A, \rho^A) \triangleq \left(\wp_{\mathrm{f}}(\mathcal{L}^\theta \times A), \rho^{\wp_{\mathrm{f}}(\mathcal{L}^\theta \times A)} = \{\rho_\pi^{\wp_{\mathrm{f}}(\mathcal{L}^\theta \times A)} \mid \pi \in perm^{fk}\}\right)$$

*where, for any $u \in \wp_{\mathrm{f}}(\mathcal{L}^\theta \times A)$, $\pi \in perm^{fk}$,*

$$\rho_\pi^{\wp_{\mathrm{f}}(\mathcal{L}^\theta \times A)}(u) \;=\; \{((\theta x_1 \dots x_{\max_n \pi})\mu[\pi], \rho_\pi a) \mid ((\theta x_1 \dots x_n)\mu, a) \in u\}$$

**Proposition 4.4** *Let $\alpha : (\mathcal{P}^\theta_\emptyset, \rho) \to T(\mathcal{P}^\theta_\emptyset, \rho)$ be defined as $\alpha(P) \triangleq \{(\mu, Q) \mid P \xrightarrow{\mu}\!\!\!\!\to_\emptyset Q\}$. Then, $\mathcal{C}_{\pi\theta} \triangleq (\mathcal{P}^\theta_\emptyset, \rho, \alpha)$ is a $T$-coalgebra.*

Since $T$ is the lifting of the corresponding polynomial functor $T_s : \mathcal{S}et \to \mathcal{S}et$, by [4, Proposition 28] we can lift the adjuction $F \dashv V$ (where $F : \mathcal{S}et \to Alg_\pi$ is the free construction and $V : Alg_\pi \to \mathcal{S}et$ is the forgetful functor) to the categories of $T_s$-coalgebras and $T$-coalgebras, thus obtaining an adjunction $T_s\text{-Coalg} \underset{V_T}{\overset{F_T}{\rightleftarrows}} T\text{-Coalg}$ (with $\perp$). We have therefore the following:

**Proposition 4.5** *The functor $T : Alg_\pi \to Alg_\pi$ has a final coalgebra $\Omega = (\Omega, \rho^\Omega, \alpha_\Omega)$, and moreover $V_T(\Omega)$ is the final $T_s$-coalgebra.*

**Proof.** The final coalgebra is given $R_T(1)$, where $R_T : Alg_\pi \to T$-Coalg is the right adjoint of the forgetful functor. □

**Remark 4.6** *One may wonder whether Proposition 4.4 still holds if we consider other constructors in the algebraic structure beside $\rho$'s. The point is that $\alpha$ has to be a morphism between permutation algebras, i.e. it has to respect the algebraic structures. This holds for $\nu, |$ (providing an alternative proof that bisimilarity is a congruence w.r.t. $\nu, |$) but it does not hold in the case of input prefix. For example, let us consider the operator $\iota_{zx}(\cdot) = z(x).\cdot$ acting on processes as follows: $\iota_{zx}((\theta \boldsymbol{y})P) \triangleq (\theta \boldsymbol{y} z)z(x).P$. If $P = (\theta y w)\bar{y}w$, then $\alpha(\iota_{zx}(P)) \neq T(\iota_{zx})(\alpha(P))$, whatever is the action of $\iota_{zx}$ on labels.*

Using Lemma 3.6 and Theorem 3.7, we can easily prove that:

**Theorem 4.7** *Let $P, Q \in \mathcal{P}$ be such that $(\theta \boldsymbol{x})P, (\theta \boldsymbol{x})Q \in \mathcal{P}^\theta_\emptyset$. Then $P \sim Q$ iff there exists a $T$-bisimulation, $\mathcal{R}$, on the coalgebra $\mathcal{C}_{\pi\theta}$ such that $(\theta \boldsymbol{x})P \;\mathcal{R}\; (\theta \boldsymbol{x})Q$.*

The following proposition characterizes $T$-bisimilarity by finality, and hence, by Theorem 4.7, also bisimilarity on $\pi$-calculus processes.

**Proposition 4.8** *The equivalence induced by the unique morphism $\mathcal{M} : \mathcal{C}_{\pi\theta} \longrightarrow \Omega$ coincides with the union of all $T$-bisimulations on the $T$-coalgebra $\mathcal{C}_{\pi\theta}$.*

Finally, we can put to use the permutation algebraic structure of coalgebras. In [14], the structure of coalgebras given by permutation algebras was used to show that the support of the interpretation of a $\pi$-calculus process in the final model amounts exactly to the active names of the process. In our setting we can obtain a similar result:

**Proposition 4.9** *For $(\theta \boldsymbol{x})P \in \mathcal{P}^\theta_\emptyset$ and $n = |\boldsymbol{x}|$, the family $\{\rho^\Omega_\pi(\mathcal{M}((\theta \boldsymbol{x})P)) \mid \pi \in \mathit{perm}^{\mathit{fk}}, \pi(\{1 \dots n\}) \subseteq \{1 \dots n\}\}$ has at most $n!/k!$ distinct elements, where $k$ is the number of active names of $P$.*

**Proof.** (Outline) The action of $\rho^\Omega_\pi$ commutes with the final semantics. By definition of active names, swapping non-active names in the process does not change its bisimilarity class. Therefore, the number of distinct elements in the family $\{\rho^\Omega_\pi(\mathcal{M}((\theta \boldsymbol{x})P)) \mid \pi \in \mathit{perm}^{\mathit{fk}}, \pi(\{1 \dots n\}) \subseteq \{1 \dots n\}\}$ is bounded by the number of different permutations of $n$ objects, $k$ of which are equal. $\square$

## 5 $\theta$-automata

In this section, drawing inspiration from [14], we introduce a notion of automaton, called *$\theta$-automaton*, for representing in a *finite* way the evolution of *finitary* $\pi\theta$-processes. These are processes whose descendants have a bounded number of possible parallel actions (*degree of parallelism*):

**Definition 5.1 (Finitary processes)** *The* degree of parallelism $\deg(P)$ *of a $\pi\theta$-process $P$ is defined as follows (for $\pi$ a generic action prefix):*

$$\deg(0) = \deg(Z) \triangleq 0 \qquad \deg(\pi.P) \triangleq 1 \qquad \deg(P \mid Q) \triangleq \deg(P) + \deg(Q)$$
$$\deg((\nu x)P) = \deg((\theta x)P) = \deg([x = y]P) = \deg(\mathrm{rec}\, Z.P) \triangleq \deg(P)$$

*Let $P \in \mathcal{P}^\theta_X$; the set of* descendants *of $P$ is $\mathit{des}(P) \triangleq \{Q \mid P \twoheadrightarrow^*_X Q\}$, where $\twoheadrightarrow^*_X$ is the reflexive and transitive closure of $\twoheadrightarrow_X$. A process $P \in \mathcal{P}^\theta_X$ is* finitary *if $\overline{\deg}(P) \triangleq \sup\{\deg(Q) \mid Q \in \mathit{des}(P)\} < \infty$.*

Structurally congruent $\pi\theta$-processes will be represented by the same state:

**Definition 5.2 (Structural Congruence)** *The* structural congruence $\equiv$ *on $\pi$-calculus processes is the smallest congruence that satisfies the following:*

**(par)** $\quad P \mid 0 \equiv P \qquad P \mid Q \equiv Q \mid P \qquad P \mid (Q \mid R) \equiv (P \mid Q) \mid R$

**(res)** $\quad (\nu x)0 \equiv 0 \qquad (\nu x)(P \mid Q) \equiv P \mid (\nu x)Q,\ \mathit{if}\ x \notin \mathit{fn}(P)$

$\qquad (\nu x)(\nu y)P \equiv (\nu y)(\nu x)P \qquad (\nu x)\pi.P \equiv \pi.(\nu x)P,\ \mathit{if}\ x \notin \mathit{fn}(\pi)$

**(match)** $\quad [x = x]P \equiv P \quad [x = y]0 \equiv 0 \qquad$ **(unfold)** $\quad \mathrm{rec}\, Z.P \equiv P\{\mathrm{rec}\, Z.P/Z\}$

*where $\pi$ stands for a generic action/matching prefix.*

*The $\pi\theta$-processes $(\theta \boldsymbol{x})P$, $(\theta \boldsymbol{x})Q$ are* structurally congruent *(also denoted by $\equiv$) if and only if $P \equiv Q$.*

For each class $\mathcal{S} \subseteq \mathcal{P}^\theta$ of congruent processes, let us fix a representative process $P$ such that $|\mathit{fn}(P)| = \mathit{min}\{|\mathit{fn}(Q)| \mid Q \in \mathcal{S}\}$.

Moreover, since processes differing by vacuous $\theta$'s will be collapsed in the same state of the $\theta$-automaton, we need to introduce a *canonical* representative for classes of congruent processes together with all processes differing by vacuous $\theta$'s. For the sake of simplicity, but without loss of generality by Lemma 3.6, we introduce $\theta$-automata only for *closed* $\pi\theta$-processes.

**Definition 5.3 (canonical terms, orbit)** *A $\pi\theta$-process $(\theta x_1 \dots x_n)P \in \mathcal{P}^\theta_\emptyset$ is* canonical *if it is the representative of a $\equiv$-class and $x_i \in fn(P)$, for all $1 \le i \le n$. Let* $\mathrm{can}(\mathcal{P}^\theta_\emptyset)$ *denote the set of canonical $\pi\theta$-processes.*

*For a canonical process $(\theta\boldsymbol{x})P$, we define its* orbit *as the set* $\mathrm{orbit}((\theta\boldsymbol{x})P) \triangleq \{(\theta\boldsymbol{z}_0 x_1 \dots \boldsymbol{z}_{n-1}x_n\boldsymbol{z}_n)P' \in \mathcal{P}^\theta_\emptyset \mid P' \equiv P \ \wedge\ |\boldsymbol{z}_0|, \dots, |\boldsymbol{z}_n| \ge 0 \ \wedge\ \forall i,j.x_j \notin \boldsymbol{z_i}\}$.

*We denote the* normalization *function by $\|\cdot\| : \mathcal{P}^\theta_\emptyset \to \mathrm{can}(\mathcal{P}^\theta_\emptyset)$, defined by $\|(\theta x_1 \dots x_n)P\| \triangleq (\theta y_1 \dots y_m)P'$, where $P'$ is the representative of the equivalence class of $P$, $\{y_1, \dots, y_m\} = fn(P')$, and $x_1 \dots x_n = \boldsymbol{z}_0 y_1 \boldsymbol{z}_1 \dots y_m \boldsymbol{z}_m$ (where $|\boldsymbol{z}_i| \ge 0$). The* reindexing *of $(\theta x_1 \dots x_n)P$ is the partial strictly monotone function $\xi((\theta x_1 \dots x_n)P)$ defined by $\xi((\theta x_1 \dots x_n)P)(i) = j \iff x_i = y_j$.*

In the following, we denote by $\mathbb{M}(n,m)$ the set of partial strictly monotone functions from $\{1, \dots, n\}$ to $\{1, \dots, m\}$.

**Definition 5.4 ($\theta$-automaton)** *Let $P \in \mathcal{P}^\theta_\emptyset$. The* $\theta$-automaton *$\mathcal{A}_P$ induced by $P$ is the triple $(\mathcal{S}, \|P\|, \mapsto)$, where:*

- *$\mathcal{S}$ is the set of* states*. Each state is the orbit of the canonical process corresponding to a descendant of $P$, and it is denoted by the canonical representative itself.*
- *$\|P\|$ is the* initial state.
- *$\mapsto \subseteq \mathcal{S} \times \mathcal{L}^\theta \times \mathcal{S}$ is the* transition relation *defined by:*

$$P_1 \overset{\mu}{\mapsto} P_2 \text{ iff there exists } P_2' \text{ such that } P_1 \xrightarrow{\mu}\!\!\!\!\twoheadrightarrow_\emptyset P_2' \text{ and } \|P_2'\| = P_2 \ .$$

In order to prove the fundamental Theorem 5.7 below, which motivates the notion of $\theta$-automaton, we first need the following technical definition.

**Definition 5.5** *Let $P$ be a $\pi\theta$-process (possibly with free process variables). The* set of subprocesses *of $P$ is defined as $sub(P) \triangleq \{P\} \cup sub'(P)$, where:*

$$\begin{array}{l}
sub'(0) = sub'(Z) \triangleq \emptyset \qquad sub'(P|Q) \triangleq sub(P) \cup sub(Q) \\
sub'(\pi.P) = sub'([x=y]P) = sub'((\nu x)P) = sub'((\theta x)P) \triangleq sub(P) \\
sub'(\mathrm{rec}\, Z.P) \triangleq \{Q\{\mathrm{rec}\, Z.P/Z\} \mid Q \in sub(P)\}
\end{array}$$

**Lemma 5.6** *If $P \in \mathcal{P}^\theta$, then the set $sub(P)$ is finite, and for all $Q \in sub(P)$: $sub(Q) \subseteq sub(P)$.*

**Theorem 5.7** *Let $(\theta\boldsymbol{x})P \in \mathcal{P}^\theta_\emptyset$ be a finitary process. Then $\mathcal{A}_{(\theta\boldsymbol{x})P}$ is finite.*

**Proof.** Let $n_0 = \overline{\deg}(P)$, and $Q$ be any descendant of $(\theta\boldsymbol{x})P$. Then, by definition of $\equiv$, either $Q$ admits a canonical form $Q \equiv (\theta\boldsymbol{z})(\nu\boldsymbol{y})(Q_1|\dots|Q_n)$, where $n \le n_0$, and $Q_i$ are *sequential* processes (i.e., non-null processes whose top operator is either an action prefix or a non-trivial matching) or $Q \equiv (\theta\boldsymbol{z})0$. Moreover, by the definition of the transition system, each component $Q_i$ is a subprocess of $P$, up-to-$\equiv$ and (possibly non injective) name substitution, i.e. $Q_i \equiv P_i\sigma$, for a name substitution $\sigma$, and $P_i$ is a subprocess of $P$. Since the subprocesses of $P$ are finite, then there are only finitely many possible $Q_i$, up-to bijective name substitutions. Moreover, the number of free names in $Q_i$ is bounded by the number of (either free or bound) names in $P$, $n(P)$, and hence the number of non vacuous $\nu$'s and $\theta$'s in $(\theta\boldsymbol{z})(\nu\boldsymbol{y})(Q_1 \mid \ldots \mid Q_n)$ is bounded by $n_0 \times |n(P)|$. Hence, the number of descendants $Q$ of a finitary process is finite, up-to-$\equiv$ and vacuous $\theta$'s. $\square$

One can easily check that the $\theta$-automaton corresponding to (the $\theta$-closure of) the recursive process of Example 3.5 is finite (actually it consists of exactly one state and one transition edge).

One could develop a complete theory of $\theta$-automata, and recover classical results, such as minimalization. But here we shall only investigate how to use $\theta$-automata in order to devise an effective procedure for establishing bisimilarity of $\pi\theta$-processes. Given two processes, in order to use the induced $\theta$-automata to check bisimilarity, we need to keep track, at each step, of the correspondence between $\theta$-bound names in the canonical processes (Example 3.9 shows that elimination of vacuous $\theta$'s could otherwise compromise bisimilarity). To this aim, we use a finitary reindexing function, mapping the positions of variables which have to be identified in the two processes. Let us denote by $\mathbb{M}$ the set $\bigcup_{n,m\in\omega} \mathbb{M}(n,m)$:

**Definition 5.8 (Indexed Bisimilarity)** *Let $\mathcal{A} = (\mathcal{S}, P_0, \mapsto)$, $\mathcal{A}' = (\mathcal{S}', Q_0, \mapsto')$ be $\theta$-automata. An* indexed bisimulation *$\mathcal{R} \subseteq \mathcal{S} \times \mathbb{M} \times \mathcal{S}'$ is a relation such that, for $(\theta\boldsymbol{x})P \in \mathcal{S}$, $(\theta\boldsymbol{y})Q \in \mathcal{S}'$, for $f \in \mathbb{M}(|\boldsymbol{x}|, |\boldsymbol{y}|)$, if $((\theta\boldsymbol{x})P, f, (\theta\boldsymbol{y})Q) \in \mathcal{R}$ then: let $dom(f) = \{i_1, \ldots, i_k\}$, $\boldsymbol{x} = \boldsymbol{u}_0 x_{i_1} \boldsymbol{u}_1 \ldots x_{i_k} \boldsymbol{u}_k$, $\boldsymbol{y} = \boldsymbol{v}_0 y_{f(i_1)} \boldsymbol{v}_1 \ldots y_{f(i_k)} \boldsymbol{v}_k$, $\boldsymbol{x} \cap \boldsymbol{y} = \emptyset$, $\boldsymbol{z} = \boldsymbol{u}_0\boldsymbol{v}_0 x_{i_1} \boldsymbol{u}_1\boldsymbol{v}_1 \ldots x_{i_k} \boldsymbol{u}_k\boldsymbol{v}_k$,*

- *if $(\theta\boldsymbol{z})P \overset{(\theta\boldsymbol{z})\mu}{\twoheadrightarrow}_\emptyset (\theta\boldsymbol{z})P'$, then there exists $(\theta\boldsymbol{z})Q'$ such that*
  - $(\theta\boldsymbol{z})Q\{x_{i_1}/y_{f(i_1)}, \ldots, x_{i_k}/y_{f(i_k)}\} \overset{(\theta\boldsymbol{z})\mu}{\twoheadrightarrow}_\emptyset (\theta\boldsymbol{z})Q'$
  - $(\|(\theta\boldsymbol{z})P'\|, f', \|(\theta\boldsymbol{z})Q')\| \in\mathcal{R}$, *where* $f' = \xi((\theta\boldsymbol{z})Q') \circ (\xi((\theta\boldsymbol{z})P'))^{-1}$.
- *if $(\theta\boldsymbol{z})P \overset{(\theta\boldsymbol{z})\mu}{\twoheadrightarrow}_\emptyset (\theta\boldsymbol{z}z')P'$, then there exists $(\theta\boldsymbol{z}z')Q'$ such that*
  - $(\theta\boldsymbol{z})Q\{x_{i_1}/y_{f(i_1)}, \ldots, x_{i_k}/y_{f(i_k)}\} \overset{(\theta\boldsymbol{z})\mu}{\twoheadrightarrow}_\emptyset (\theta\boldsymbol{z}z')Q'$
  - $(\|(\theta\boldsymbol{z}z')P'\|, f', \|(\theta\boldsymbol{z}z')Q')\| \in\mathcal{R})$, *where* $f' = f_1 \cup f_2$, $f_1 = \xi((\theta\boldsymbol{z}z')Q') \circ (\xi((\theta\boldsymbol{z}z')P'))^{-1}$ *and*

$$f_2 = \begin{cases} \{(\max(\mathrm{dom}(f_1)) + 1, \max(\mathrm{cod}(f_1)) + 1)\} & \textit{if } z' \in fn(P') \cap fn(Q') \\ \emptyset & \textit{otherwise} \end{cases}$$

- $((\theta\boldsymbol{y})Q, f^{-1}, (\theta\boldsymbol{x})P) \in\mathcal{R}$ .

*The* indexed bisimilarity, *$\simeq$, is the greatest indexed bisimulation. We say that the automata $\mathcal{A}$ and $\mathcal{A}'$ are $f$*-bisimilar *if $(P_0, f, Q_0) \in\simeq$, for some $f \in \mathbb{M}(|\boldsymbol{x}|, |\boldsymbol{y}|)$.*

Using Lemma 3.6, one can prove that:

**Theorem 5.9** *Let $P, Q \in \mathcal{P}_{\{x_1,\ldots,x_n\}}$. Then, $P \sim Q$ iff $\mathcal{A}_{(\theta\boldsymbol{x})P}$ and $\mathcal{A}_{(\theta\boldsymbol{x})Q}$ are $f$-bisimilar, for $f = \xi((\theta\boldsymbol{x})Q) \circ (\xi((\theta\boldsymbol{x})P))^{-1}$.*

If the sets of states $\mathcal{S}$ and $\mathcal{S}'$ of the automata are finite, then indexed bisimulations are finite objects.

**Theorem 5.10** *Let* $\mathcal{A}_{(\theta\boldsymbol{x})P} = (\mathcal{S}, \|(\theta\boldsymbol{x})P\|, \mapsto)$ *and* $\mathcal{A}_{(\theta\boldsymbol{x})Q} = (\mathcal{S}', \|(\theta\boldsymbol{x})Q\|, \mapsto')$, *for* $P, Q \in \mathcal{P}_{\{x_1,\ldots,x_n\}}$ *finitary. Then* $P \sim Q$ *iff there exists an indexed bisimulation* $\mathcal{R} \subseteq \mathcal{S} \times \mathbb{M}(k,k) \times \mathcal{S}'$, *with* $k = \max(\overline{\deg}(P) \times |n(P)|, \overline{\deg}(Q) \times |n(Q)|)$, *such that* $(\|(\theta\boldsymbol{x})P\|, f, \|(\theta\boldsymbol{x})Q\|) \in \mathcal{R}$, *where* $f = \xi((\theta\boldsymbol{x})Q) \circ (\xi((\theta\boldsymbol{x})P))^{-1}$.

Notice that $k$ is an upper bound of the domain of the reindexing functions between all descendents of $P$ and $Q$. Thus, since $\mathbb{M}(k,k)$ is finite, there are only finitely many candidate relations to be indexed bisimulations. Hence, we have an algorithm for deciding bisimilarity of finitary processes.

## 6 Final Remarks and Directions for Future Work

In this paper, we have introduced the *$\pi\theta$-calculus,* a conservative extension of the $\pi$-calculus, which allows to explain away the mechanism of name creation by means of a new unary binding operator. We have used the $\pi\theta$-calculus to give a coalgebraic description of early bisimilarity. This semantics is finitary, in the sense that it is finitely branching and moreover, for any finitary process, the set of descendants is finite, up-to vacuous $\theta$'s and structural congruence.

Furthermore, we have also introduced *$\theta$-automata* which we use to get a truly finite representation for finitary processes, by equating in a single state a process together with all the processes differing from it by sequences of vacuous $\theta$'s and structural congruence. We could push further the study of $\theta$-automata, by introducing a general notion of $\theta$-automaton, independent from the $\pi$-calculus. Standard results on automata such as minimalization could then be naturally recovered. Moreover, there should be a corresponding notion of transition system, generalizing the transition system of the $\pi\theta$-calculus.

In [14], an alternative finitary coalgebraic semantics for the $\pi$-calculus is proposed for early bisimilarity, and a corresponding notion of automaton, the *History Dependent Automaton*, is discussed. The problem of generating a fresh name in bound output transitions is solved by applying a suitable permutation on the names of the process, so as to guarantee that a special concrete name is always fresh in the permuted process. This latter name is the new name used in the bound output transition. In a loose sense, this can be viewed as a first-order approach, whereas the one using $\theta$-closure operators of this paper is second-order. However, it is an interesting future work to investigate which is the precise connection between HD-automata and $\theta$-automata, and to understand which class of automata provides a more convenient (i.e. more compact, easier to manipulate) representation of the behavior of a given process. Furthermore, in [7] both HD-automata and $\theta$-automata are claimed to be instances of a the more general concept of *graph with binding*, although the details still remains to be spelled out. Either reconciling all the three different approaches above or understanding and formalising their differences is a necessary task that we intend to investigate.

As in the case of [14], the final coalgebra we have considered has the structure of a *permutation algebra*. In [14], this allows to prove that e.g., the support of the interpretation of a $\pi$-calculus process in the final model consists exactly of the active names of the given process. In our development, the counterpart of this result is Proposition 4.9: in the permutation algebra of the final model, there are only finitely many different $\rho_\pi$-closures of the interpretation of a closed process $P$, corresponding to the number of different permutations of the active names of $P$.

In this paper we have considered early bisimilarity, but similar techniques can be used to account for late and weak bisimilarities. Moreover, it would be interesting to explore also denotational (i.e. compositional) models for early/late congruences based on the $\pi\theta$-calculus. We expect to get simpler models than the ones based on functor categories.

Finally, due to the "symbolic" nature of our operational semantics, it would be interesting to compare it to other symbolic approaches that are at the basis of different $\pi$-calculus implementations and tools, e.g. [8,17].

## References

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

[2] P. Aczel and N. Mendler. A final coalgebra theorem. In *Proc. CTCS'89*, LNCS 389, pages 357–365, 1989. Springer-Verlag.

[3] M. G. Buscemi and U. Montanari. A first order coalgebraic model of $\pi$-calculus early observational equivalence. In *Proc. CONCUR 2002*, LNCS 2421, pages 449–465. Springer-Verlag, 2002.

[4] A. Corradini, R. Heckel, and U. Montanari. Compositional SOS and beyond: A coalgebraic view of open systems. *Theor. Comput. Sci.* 280, 163–192, 2002.

[5] J. Despeyroux. A higher-order specification of the pi-calculus. In *Proc. of the IFIP TCS'2000*, Sendai, Japan, 2000.

[6] M. J. Gabbay. Theory and models of the $\pi$-calculus using Fraenkel-Mostowski generalized. Submitted, 2002.

[7] M. J. Gabbay. Fresh graphs. Submitted, 2004.

[8] M. Hennessy and H. Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138:353–389, 1995.

[9] F. Honsell, M. Lenisa, U. Montanari, and M. Pistore. Final semantics for the $\pi$-calculus. In D. Gries, editor, *Proc. PROCOMET'98*. Chapman&Hall, 1998.

[10] F. Honsell, M. Miculan, and I. Scagnetto. $\pi$-calculus in (co)inductive type theory. *Theoretical Computer Science*, 253(2):239–285, 2001.

[11] M. Lenisa. *Themes in Final Semantics.* PhD thesis, Dipartimento di Informatica, Università di Pisa, Italy, 1998.

[12] D. Miller and C. Palmidessi. Foundational aspects of syntax. *ACM Computing Surveys (CSUR)*, 31(3es):11, 1999.

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

[14] U. Montanari and M. Pistore. $\pi$-calculus, structured coalgebras, and minimal HD-automata. In *Proc. MFCS 2000*, LNCS 1893, pages 569–578, 2000.

[15] J. Parrow. An introduction to the $\pi$-calculus. In *Handbook of Process Algebra*, pages 479–543. Elsevier Science, 2001.

[16] J. J. M. M. Rutten and D. Turi. On the foundations of final semantics: Non-standard sets, metric spaces, partial orders. In *REX Conf. Proc.*, volume 666 of *Lecture Notes in Computer Science*, pages 477–530. Springer-Verlag, 1993.

[17] D. Sangiorgi. A theory of bisimulation for the $\pi$-calculus. *Acta Informatica*, 33:69–97, 1996.

[18] D. Sangiorgi and D. Walker. *The $\pi$-calculus: a Theory of Mobile Processes.* Cambridge Universtity Press, 2001.