Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 340 (2005) 539–576
Theoretical Computer Science
www.elsevier.com/locate/tcs

Structured coalgebras and minimal HD-automata for the π-calculus☆

Ugo Montanari[a], Marco Pistore[b,*]

[a] *Computer Science Department, University of Pisa, Corso Italia 40, 56100 Pisa, Italy*
[b] *Department of Information and Comm. Technology, University of Trento, Via Sommarive 14, 38050 Povo (Trento), Italy*

Abstract

The coalgebraic framework developed for the classical process algebras, and in particular its advantages concerning minimal realizations, does not fully apply to the π-calculus, due to the constraints on the freshly generated names that appear in the bisimulation.

In this paper we propose to model the transition system of the π-calculus as a coalgebra on a category of name permutation algebras and to define its abstract semantics as the final coalgebra of such a category. We show that permutations are sufficient to represent in an explicit way fresh name generation, thus allowing for the definition of minimal realizations.

We also link the coalgebraic semantics with a slightly improved version of history dependent (HD) automata, a model developed for verification purposes, where states have local names and transitions are decorated with names and name relations. HD-automata associated with agents with a bounded number of threads in their derivatives are finite and can be actually minimized. We show that the bisimulation relation in the coalgebraic context corresponds to the minimal HD-automaton.

Keywords: Concurrency and distributed computation; π-calculus; Categorical models and logics; Coinductive techniques; Automata theory

☆ Research supported in part by FET Global project *PROFUNDIS* and by MIUR project *COMETA*.

* Corresponding author.
E-mail addresses: ugo@di.unipi.it (U. Montanari), pistore@dit.unitn.it (M. Pistore).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2005.03.014

1. Introduction

The π-calculus [21] is probably the best studied calculus for name mobility, and the basis for several proposals concerning higher order mobility [31], security [1], and object orientation [36]. Also, π-calculus expressiveness can be considered the touchstone for a number of formalisms [5,14,16,35] exploring the needs of wide area programming. The advantage of the π-calculus is its simplicity and its process algebra flavor, e.g., its operational semantics given by means of a transition system and its abstract semantics based on bisimilarity.

However, while a process calculus like CCS, at least in the strong case, can be easily casted in a coalgebraic framework [30], the π-calculus requires some care. Indeed, consider the definition of early bisimulation.

Definition (*Early bisimulation*). A relation $\mathcal{R}$ over agents is an *early simulation* if $P \mathcal{R} Q$ implies:

- for each $P \xrightarrow{\alpha} P'$ with $\mathrm{bn}(\alpha) \cap \mathrm{fn}(P, Q) = \emptyset$ there is some $Q \xrightarrow{\alpha} Q'$ such that $P' \mathcal{R} Q'$.

A relation $\mathcal{R}$ is an *early bisimulation* if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are early simulations. *Early bisimilarity* $\sim_\pi$ is the largest early bisimulation.

Notice the condition "$\mathrm{bn}(\alpha) \cap \mathrm{fn}(P, Q) = \emptyset$": the first agent is not allowed to use as bound name in a transition a name that is already syntactically present in the second agent. Thus the bisimilarity class of an agent cannot be defined "in isolation", but only relatively to possible partners, or at least to their free names. As a consequence, the coalgebraic framework does not fully apply. In practice, algorithms for checking bisimilarity based on the above definition are only of the "on the fly" kind, and with them it is not possible to construct the minimal equivalent agent. To apply the standard definition of bisimulation also in this context, it is necessary to define a mechanism of name allocation which guarantees that the fresh names are chosen in a consistent way by the agents P and Q. The definition of this mechanism is critical, as it decides whether the obtained transition systems are finite- or infinite-states: to obtain finite-state models, it is necessary not only to define how fresh names are allocated, but also how unused names are deallocated, so that they can be used again as fresh names.

Besides minimal realizations, a fully satisfactory coalgebraic theory for the π-calculus would yield the possibility of applying well understood algebraic techniques and proof methods [6–8,19,34].

A coalgebraic semantics for various versions of the π-calculus has been proposed in [18], but the approach is higher order, and it is not amenable to finite state verification. Moreover, the semantics of [18] is still parametrized by the set of names of the possible partners, so that an infinite set of parametrized states is necessary from the beginning. Also, the set of names that parameterizes an agent grows during the evolution of the agent, due to bound output and input transitions, and never shrinks. For these reasons the semantics of [18] defines intrinsically infinite-state models.

In the paper we propose a standard coalgebraic definition of the π-calculus semantics, which is based on name permutations. The effects of permutations on the behavior of agents is, in our opinion, the smallest information required to define a semantically correct mechanism of name deallocation. Indeed, as we will see, according to our approach the