π-calculus in (Co)Inductive Type Theory (Abstract)

Furio Honsell* Marino Miculan Ivan Scagnetto
Dipartimento di Matematica e Informatica, Università di Udine
mailto:{honsell|miculan|scagnett}@dimi.uniud.it

We present an interactive proof assistant for Milner's π-calculus, based on a *higher-order abstract syntax* (HOAS) presentation of the calculus in the Intuitionistic Coinductive Type Theory $\mathrm{CC}^{(Co)Ind}$ [3]. The utility and flexibility of this proof assistant is illustrated by developing the formal theory of *strong late bisimilarity* as presented in Section 2 of *A calculus of mobile processes* by Milner, Parrow, and Walker [16].

Like many other processes algebras, the π-calculus is a rather intricate formal system in itself. Informal arguments for equivalence of processes are error prone, since exhaustive case analyses on possible transitions are elusive. Hence, this calculus, as all the field of formal methods for reasoning on concurrent systems, seems particularly in need of practical *computer assisted proof editors*.

Building an implementation of a particular version of the π-calculus, and a given notion of process equivalence, from scratch does not seem to be a tenable approach. The variability in the presentations and the wide range of equivalences under consideration, do not make it worth to put so much effort in an enterprise which is so particular. Hence, in order to achieve more generality, we have explored the possibility of tailoring a *generic* proof editor to the π-calculus using an HOAS encoding, which subsumes a number of different presentations, and which can serve as a paradigm for the plethora of π-calculi.

A *generic proof development environment* is a proof development environment for a particular system which can play the rôle of a logic specification language, *i.e.* of a *Logical Framework* [13, 10]. Since the '80's, higher order predicative, or impredicative, intuitionistic type theories have been successfully experimented as Logical Frameworks [10, 1, 6, 15]. In these theories one can represent (formalize) faithfully and uniformly all the relevant notions and aspects of the inference process in an arbitrary system: syntactic categories, terms, assertions, axiom schemata, rule schemata, tactics. The basic idea is the "judgements-as-types, λ-terms-as-proofs" paradigm.

Following this line of research, our interactive proof assistant for the π-calculus, is obtained directly from Coq [3], which is the full-fledged computerized proof environment for $\mathrm{CC}^{(Co)Ind}$, developed at INRIA. The crucial step is the specification in $\mathrm{CC}^{(Co)Ind}$ of an *adequate* encoding of the π-calculus. Our encoding builds upon a higher order abstract syntax (HOAS) presentation of the π-calculus, in the style of Church, whereby binding operators are represented by constructors of higher order type [10, 6, 11, 15].

We think that we offer a substantial case study in computer assisted formal reasoning which is significant for various reasons.

First, we illustrate, contrary to common belief, how even a complex formal system such as π-calculus can benefit from an HOAS presentation. This allows to delegate conveniently to the metalanguage the details of many syntactic manipulations, which, in the case of the π-calculus, include, *e.g.* α-conversion of binders, substitution of names and the machinery for generating fresh local names.

Secondly, following the approach pioneered by Coquand and Gimenez [5, 9, 8], we investigate the possibility of reasoning on infinite and circular objects using proposition-as-coinductive types. Thus, coinduction proofs are rendered as infinite proof objects, and bisimulations need not be exhibited at the outset, but can be built incrementally using natural tactics. This approach turns out to be extraordinarily successful.

The use of HOAS is critical when reflecting on those properties which HOAS delegates to the metalanguage, namely, substitution, α-conversion and freshness of names. This is the case of many of the Lemmata and propositions in [16, Section 2]. In order to prove such properties, we need to extend the en-

*currently at LFCS Division of Informatics, University of Edinburgh coding with new postulates which reify at the object level, some of the details that HOAS delegates at the metalanguage. We have tried to isolate a restricted set of general properties capturing a general theory of contexts.

This paper is part of an ongoing research programme at the Computer Science Department of the University of Udine, started in 1992, [14, 12, 15] on proof editors, based on HOAS encodings in dependent typed λ-calculus for program logics[1]. Our experience, which is nicely confirmed also in dealing with π-calculus, is that Logical Frameworks allow to encode faithfully the formal systems under consideration, without imposing on the user of the proof editor the burden of cumbersome encodings.

References

[1] A. Avron, F. Honsell, I. A. Mason, and R. Pollack. Using Typed Lambda Calculus to implement formal systems on a machine. *Journal of Automated Reasoning*, 9:309–354, 1992.

[2] *Proceedings of TYPES'93*. LNCS 806, Springer-Verlag, 1994.

[3] *The Coq Proof Assistant Reference Manual - Version 6.2*. INRIA, May 1998. Available at `ftp://ftp.inria.fr/INRIA/coq/V6.2/doc`.

[4] S. Berardi and M. Coppo, editors. *Proceedings of TYPES'95*. LNCS 1158, Mar. 1995. Springer-Verlag, 1996.

[5] T. Coquand. Infinite objects in type theory. In [2], pages 62–78.

[6] J. Despeyroux, A. Felty, and A. Hirschowitz. Higher-order syntax in Coq. In *Proc. of TLCA'95*. LNCS 905, Springer-Verlag, 1995.

[7] M. Felchero. Sistemi di transizione in teoria dei tipi coinduttivi. Laurea's thesis, Università di Udine, Italy, July 1996. In Italian.

[8] E. Giménez. An application of co-inductive types in Coq: Verification of the Alternating Bit Protocol. In Berardi and Coppo [4], pages 134–152.

[9] E. Giménez. Codifying guarded recursion definitions with recursive schemes. In *Proc. of TYPES'94*. LNCS 996, Springer-Verlag, 1995.

[10] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *J. ACM*, 40(1):143–184, Jan. 1993.

[11] F. Honsell, M. Lenisa, U. Montanari, and M. Pistore. Final semantics for the π-calculus. In *Proc. PROCOMET'98*. IFIP and Chapman & Hall, 1998.

[12] F. Honsell and M. Miculan. A natural deduction approach to dynamic logics. In Berardi and Coppo [4], pages 165–182.

[13] P. Martin-Löf. On the meaning of the logical constants and the justifications of the logic laws. TR 2, Dip. di Matematica, Università di Siena, 1985.

[14] M. Miculan. The expressive power of structural operational semantics with explicit assumptions. In [2], pages 292–320.

[15] M. Miculan. *Encoding Logical Theories of Programs*. PhD thesis, Dipartimento di Informatica, Università di Pisa, Italy, Mar. 1997.

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

[17] A. Rossi. Tipi di dati coinduttivi: i reali. Laurea's thesis, Università di Udine, Italy, 1994.

[18] I. Scagnetto. Rappresentazione di algebre di processi in Logical Frameworks. Laurea's thesis, Università di Udine, Mar. 1997. In italian.

[1] Some of the material contained in this paper is the object of three Laurea thesis at the University of Udine [17, 7, 18].