数理解析研究所講究録
996 巻 1997 年 96-97

Symbolic Bisimulations and Proof Systems for The π-Calculus *
(Abstract)

Huimin Lin†
Laboratory for Computer Science
Institute of Software, Chinese Academy of Sciences

Axiomatisations for basic process algebras such as CCS, CSP, ACP, LOTOS ..., are based on equational logic. But purely equational reasoning is difficult for the π-calculus because here input prefixes introduce *bound* names, i.e. names which are local within the scope of the input prefixes and can be instantiated with arbitrary free names. Reflected in the axiomatisations this amounts to require when an input prefix is introduced the input name must not be used before, a concept difficult to be formulated in the framework of equational logic. The "symbolic" style proof system generalise equational logic by using *conditional equations* of the form $\phi \rhd T = U$ (where ϕ is a boolean condition on names and T, U are process terms) as judgements. For each construct in the calculus there corresponds an introduction rule in the core proof system. For instance the rule for input prefix is

$$\text{INPUT} \quad \frac{\phi \rhd T = U}{\phi \rhd a(x).T = b(x).U} \quad \phi \Rightarrow a = b, \quad x \notin n(\phi)$$

Here ϕ represents the context condition for deriving $T = U$, and the side condition ensures the input name x is fresh.

In [Lin94] it is shown that, together with the standard CCS axioms for strong bisimulation, this core proof system is sound and complete for late strong bisimulation in recursion-free π-calculus. By adding either an inference rule dealing with early input ([HL96]) or an axiom schema ([PS93]) proof systems for early bisimulation can be obtained. In [Lin95a] it is further shown that to extend these results to weak bisimulation equivalences all what needed is the three τ-laws ([Mil89]).

For the π-calculus with recursion the key inference rule is *unique fixpoint induction*. It is syntactically identical to its pure-*CCS* counterpart ([Mil84]), but work at the level of *abstractions*:

$$\text{UFI} \quad \frac{F = G[F/X]}{F = \mu X G} \quad X \text{ guarded in } G$$

*Supported by research grants from the President Fund of Chinese Academy of Sciences and National Natural Science Foundation of China.

†The author's address: Institute of Software, Chinese Academy of Sciences, P.O.Box 8718, Beijing 100080. Email: lhm@ox.ios.ac.cn.

where both F and G are name-closed abstractions. Extended with this rule, and the standard rules for folding/unfolding recursions, the core proof system is sound and complete for strong bisimulation in finite-control π-calculus where recursions are guarded ([Lin95b]). To achieve complete proof system for arbitrary recursions, the following axiom can be adopted to reduce unguarded recursions to guarded ones:

$$\text{UNG} \quad \mathbf{fix}X(\vec{x})(\sum_{i\in I}\phi_i\nu\vec{x}_i'X(\vec{x}_i)+T) = \mathbf{fix}X(\vec{x})(\sum_{i\in I}\phi_i\nu\vec{x}_i'T[\vec{x}_i/\vec{x}]+T)$$

where the summands in the recursion body satisfy certain *saturation* condition.

The proofs of the above mentioned completeness results rely on the notion of *symbolic bisimulation*, first proposed in [HL95] for general message-passing process algebras. As alternative characterisations of the standard definitions of bisimulations in the π-calculus, symbolic bisimulations overcome the infinity inherited in the standard definitions by employing finite partitions over the name space, instead of instantiating input names and name parameters of recursively defined processes. Another powerful technical device upon which the completeness proofs heavily rely is the notion of *maximally consistent* name conditions.

Currently work is going on to extend the above results to weak bisimulation equivalences in finite-control π-calculus.

References

[HL95] M. Hennessy and H. Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138:353–389, 1995.

[HL96] M. Hennessy and H. Lin. Proof systems for message-passing process algebras. *Formal Aspects of Computing*, 8:408–427, 1996.

[Lin94] H. Lin. Symbolic bisimulations and proof systems for the π-calculus. Report 7/94, Computer Science, University of Sussex, 1994.

[Lin95a] H. Lin. Complete inference systems for weak bisimulation equivalences in the π-calculus. In *Proceedings of Sixth International Joint Conference on the Theory and Practice of Software Development*, volume 915 of *Lecture Notes in Computer Science*. Springer–Verlag, 1995.

[Lin95b] H. Lin. Unique fixpoint induction for the π-calculus. In *Proceedings of Sixth International Conference on Concurrency Theory*, volume 962 of *Lecture Notes in Computer Science*. Springer–Verlag, 1995.

[Mil84] R. Milner. A complete inference system for a class of regular behaviours. *J. Computer and System Science*, 28:439–466, 1984.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[PS93] J. Parrow and D. Sangiorgi. Algebraic theories for name-passing calculi. Report ECS-LFCS-93-262, LFCS, University of Edinburgh, 1993.