ProPi: A Tool for Progress Analysis of Message Passing Programs

André Camacho[1] Vasco T. Vasconcelos[1]
Hugo T. Vieira[2]
[1]University of Lisbon, Portugal
[2]IMT School for Advanced Studies Lucca, Italy

In this document we present ProPi, a prototype tool developed in the context of COST Action IC12012 BETTY.

1 Abstract

Formal analysis of software systems is of crucial importance in the delivery of reliable systems. In the last decades there has been a significant research effort dedicated to the development of verification techniques and tools that allow to ensure correctness properties of programs. One of such research efforts has focused on ensuring programs never incur in deadlocks, namely in the context of message passing programs [3, 4, 5, 6, 7, 8, 10, 11]. In particular in the realm of (binary) session, we address the analysis of progress by extending session types with a notion of *event* and respective prescribed ordering that allows to single out programs that are deadlock free [13]. Our approach is inspired by the work of Padovani [9], and allows to address systems that are out of reach of other previous approaches. We have also extended the approach so as to cope with multiparty interaction [2].

The ProPi tool [1] is a prototype implementation of the progress analysis for binary sessions [13]. The tool takes as input a system specified in the π-calculus [12], together with typing annotations that describe the communications in channels, as well as event annotations and respective ordering so as to capture the overall dependencies of communications. The tool produces as result either a positive or negative answer. In the case of a positive answer (the system type checks), the properties of protocol fidelity (system communications follow the typing prescription) and progress (deadlock absence) hold. In the case of a negative answer, the system exhibits error information so as to allow to identify what is the problem in the system specification. The tool is available as an Eclipse plugin and as a standalone command line interface.

References

[1] ProPi. `http://gloss.di.fc.ul.pt/propi`.

[2] L. Caires and H. T. Vieira. Conversation types. *Theor. Comput. Sci.*, 411(51-52):4399–4440, 2010.

[3] M. Coppo, M. Dezani-Ciancaglini, N. Yoshida, and L. Padovani. Global progress for dynamically interleaved multiparty sessions. *Mathematical Structures in Computer Science*, 26(2):238–302, 2016.

[4] M. Dezani-Ciancaglini, U. de'Liguoro, and N. Yoshida. On Progress for Structured Communications. In G. Barthe and C. Fournet, editors, *Trustworthy Global Computing, Third Symposium, TGC 2007, Revised Selected Papers*, volume 4912 of *LNCS*, pages 257–275. Springer, 2007.

[5] E. Giachino, N. Kobayashi, and C. Laneve. Deadlock Analysis of Unbounded Process Networks. In P. Baldan and D. Gorla, editors, *CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Proceedings*, volume 8704 of *LNCS*, pages 63–77. Springer, 2014.

[6] M. Giunti and A. Ravara. Towards Static Deadlock Resolution in the π-Calculus. In M. Abadi and A. Lluch-Lafuente, editors, *Trustworthy Global Computing - 8th International Symposium, TGC 2013, Revised Selected Papers*, volume 8358 of *LNCS*, pages 136–155. Springer, 2013.

[7] N. Kobayashi. A New Type System for Deadlock-Free Processes. In C. Baier and H. Hermanns, editors, *CONCUR 2006 - Concurrency Theory, 17th International Conference, CONCUR 2006, Proceedings*, volume 4137 of *LNCS*, pages 233–247. Springer, 2006.

[8] Luca Padovani. Deadlock and lock freedom in the linear π-calculus. In T. A. Henzinger and D. Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL)*

and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, pages 72:1–72:10. ACM, 2014.

[9] L. Padovani. From Lock Freedom to Progress Using Session Types. In N. Yoshida and W. Vanderbauwhede, editors, *Proceedings 6th Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software, PLACES 2013*, volume 137 of *EPTCS*, pages 3–19, 2013.

[10] L. Padovani, T. Chen, and A. Tosatto. Type Reconstruction Algorithms for Deadlock-Free and Lock-Free Linear π-Calculi. In T. Holvoet and M. Viroli, editors, *Coordination Models and Languages - 17th IFIP WG 6.1 International Conference, COORDINATION 2015, Proceedings*, volume 9037 of *LNCS*, pages 83–98. Springer, 2015.

[11] L. Padovani and L. Novara. Types for Deadlock-Free Higher-Order Programs. In S. Graf and M. Viswanathan, editors, *Formal Techniques for Distributed Objects, Components, and Systems - 35th IFIP WG 6.1 International Conference, FORTE 2015, Proceedings*, volume 9039 of *LNCS*, pages 3–18. Springer, 2015.

[12] D. Sangiorgi and D. Walker. *The Pi-Calculus - a theory of mobile processes*. Cambridge University Press, 2001.

[13] H. T. Vieira and V. T. Vasconcelos. Typing Progress in Communication-Centred Systems. In R. D. Nicola and C. Julien, editors, *Coordination Models and Languages, 15th International Conference, COORDINATION 2013, Proceedings*, volume 7890 of *LNCS*, pages 236–250. Springer, 2013.