Verification of a Simple Distributed Algorithm in Erlang

Gennady Chugunov and Lars-åke Fredlund

Swedish Institute of Computer Science,
Box 1263, SE-164 29 Kista, Sweden
E-mail: `gena|fred@sics.se`

Abstract

In this talk we present the verification of a simple distributed algorithm which has been implemented in the Erlang programming language [1]. The network topology considered in the verification example is a linked list of processes. The correctness properties of the algorithm are expressed in the modal μ-calculus extended with data. For proving that the program satisfies its specification, we use the Erlang Verification Tool [3] (henceforth EVT), a general purpose proof assistant tool in which a small-step operational semantics of Erlang has been embedded. The underlying proof system is a Gentzen-style one, with a particular emphasis on a "lazy" scheme for fixed point induction and proof rules for compositional reasoning [2].

The implementation of the algorithm in Erlang, and its proof in EVT, is interesting for of a number of reasons:

- The implementation is a mix of pure functional code and side-effects (processes communicating via asynchronous message passing, and spawning of new processes). As a consequence, the resulting proof essentially splits into two parts: one considering functional evaluation (on the level of functional expressions), and one considering the side-effects (on the level of communicating processes).
- The proof is parametric in the size of the linked process list, and moreover spawning of new processes takes place. As a result the algorithm cannot (easily) be attacked by traditional "finite-state" model checking techniques.
- The proof is semi-automated and makes heavy use of higher-level tactics (derived proof rules) for automatically solving most proof obligations.
- The proof is compositional: we replace, via sound proof rules, goals talking about the concurrent execution of concrete processes, to the composition of components satisfying formulas in the modal μ-calculus via proof rules of the shape

$$\frac{\begin{array}{c}\Gamma \vdash P_1 : \psi_1, \Delta \\ \Gamma \vdash P_2 : \psi_2, \Delta \\ \Gamma, X_1 : \psi_1, X_2 : \psi_2 \vdash X_1 \parallel X_2 : \phi, \Delta\end{array}}{\Gamma \vdash P_1 \parallel P_2 : \phi, \Delta}$$

  where P_1 and P_2 are concrete processes, ϕ, ψ_1, ψ_2 are formulas in the logic and X_1 and X_2 are fresh variables.

- The non-automated parts of the proof consists concerns mostly the setting up of the induction hypothesis, and compositional reasoning using proof rules like the one shown above.
- In obtaining the proof using EVT we made use of a newly developed graphical user interface that considerably eased the proof challenge.

References

[1] J. Armstrong, R. Virding, C. Wikström and M. Williams. *Concurrent Programming in Erlang*. 2nd edition, Prentice Hall, 1996.

[2] M. Dam, L.-å. Fredlund and D. Gurov. Toward Parametric Verification of Open Distributed Systems. H. Langmaack, A. Pnueli, W.-P. De Roever (eds.), *Compositionality: The Significant Difference*, Springer Verlag, 1998.

[3] T. Arts, G. Chugunov, M. Dam, L.-å. Fredlund, D. Gurov, and T. Noll. *A Tool for Verifying Software Written in Erlang*, Submitted to the International Journal, Software Tools for Technology Transfer.