Separation logic style reasoning in a refinement based language

Gergely Dévai, Zoltán Csörnyei
Eötvös Loránd University,
Department of Programming Languages and Compilers
deva@elte.hu, csz@inf.elte.hu

Abstract

Separation logic [8, 9, 10] is an extension of classical logic to reason about programs that involve pointers and dynamic memory management. It is known that separation logic can be expressed in classical logic [5, 2], so its power lies in its reasoning style.

Refinement based programming [7, 1, 6] starts with the formal description of the requirements concerning the program. This specification is then refined in several steps towards an implementation, which is correct by construction.

In this paper we show how to build proofs in a refinement based language using the style of separation logic. We transform special elements of separation logic back into classical logic in order to be able to handle them in the selected system [3, 4]. Using the resulting types, functions and axioms, we present a well known example proof of separation logic.

References

[1] J.-R. Abrial. *The B-book: assigning programs to meanings.* Cambridge University Press, New York, NY, USA, 1996.

[2] C. Calcagno, P. Gardner, and M. Hague. From separation logic to first-order logic. In *FOSSACS*, 2005.

[3] G. Dévai. Programming language elements for correctness proofs. In *Volume of extended abstracts of the 5th Conference of PhD students in Computer Science*, pages 40–41, 2006.

[4] G. Dévai. Programming language elements for proof construction. In *Volume of abstracts of the 6th Joint Conference on Mathematics and Computer Science*, 2006.

[5] Etienne Lozes. Comparing the expressive power of separation logic and classical logic.

[6] J. McDonald and J. Anton. Specware - producing software correct by construction, 2001.

[7] Joseph M. Morris. A theoretical basis for stepwise refinement and the programming calculus. *Sci. Comput. Program.*, 9(3):287–306, 1987.

[8] Peter O'Hearn, John Reynolds, and Hongseok Yang. Local reasoning about programs that alter data structures. *Lecture Notes in Computer Science*, 2142, 2001.

[9] J. Reynolds. Separation logic: a logic for shared mutable data structures. *Invited Paper, LICS'02*, 2002.

[10] Hongseok Yang and Peter W. O'Hearn. A semantic basis for local reasoning. In *Foundations of Software Science and Computation Structure*, pages 402–416, 2002.