Programming with Miracles

Rajeev Joshi

Laboratory for Reliable Software*
NASA Jet Propulsion Laboratory, Pasadena, CA, USA

Abstract. In his seminal book, *A Discipline of Programming* [EWD 76], Dijkstra proposed that all sequential programs satisfy four laws for their weakest preconditions. By far the catchiest name was reserved for the Law of the Excluded Miracle, which captured the intuition that, started in a given state, a program execution must either terminate or loop forever. In the late 1980s, both Nelson [GN 89] and Morgan [CCM 90] noted that the law was unnecessarily restrictive when writing programs to be used as specifications. In the years since,"miracles" have become a standard feature in specification languages (for instance, the `assume` statement in JML [LLP+00] and BoogiePL [DL 05]).

What is perhaps surprising is that miracles are not as commonly used in programs written as implementations. This is surprising because for many everyday tasks, programming in a language with miracles is often far superior to the popular scripting languages that are used instead. In this talk, we build upon pioneering work by Burrows and Nelson [GN 05] who designed the language LIM ("Language of the Included Miracle"). We describe a language LIMe ("LIM with extensions"), and discuss its application in the context of flight software testing, including the analysis of spacecraft telemetry logs.

References

[EWD 76] Dijkstra, E.W.: A Discipline of Programming. Prentice-Hall, Englewood Cliffs (1976)

[GN 89] Nelson, G.: A Generalization of Dijkstra's Calculus. ACM Transactions on Programming Languages and Systems (TOPLAS) 11(4) (1989)

[CCM 90] Carroll Morgan, C.: Programming from specifications, Prentice Hall International Series in Computer Science, NJ, USA (1990), 2nd edition (1994)

[LLP+00] Leavens, G.T., Leino, K.R.M., Poll, E., Ruby, C., Jacobs, B.: JML: notations and tools supporting detailed design in Java. In: OOPLSA 2000 Companion, pp. 105–106. ACM, New York (2000)

[DL 05] DeLine, R., Leino, K.R.M.: BoogiePL: A typed procedural language for checking object-oriented programs, Microsoft Research Technical Report MSR-TR-2005-70 (March 2005)

[GN 05] Nelson, G.: LIM and Nanoweb, Hewlett-Packard Laboratories Technical Report HPL-2005-41 (February 2005)

* The research described in this talk was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

D. Méry and S. Merz (Eds.): IFM 2010, LNCS 6396, p. 27, 2010.