Towards Automatic Synthesis of Software Verification Tools

Andrey Rybalchenko

Technische Universität München

Software complexity is growing, so is the demand for software verification. Soon, perhaps within a decade, wide deployment of software verification tools will be indispensable or even mandatory to ensure software reliability in a large number of application domains, including but not restricted to safety and security critical systems. To adequately respond to the demand we need to eliminate tedious aspects of software verifier development, while providing support for the accomplishment of creative aspects.

We believe that the next generation of software verifiers will be constructed from logical specifications designed by quality/verification engineers with expertise in the application domain. Given a specification describing a verification method, a corresponding software verifier will be obtained by implementing a frontend that translates software source code into constraints according to the specification and then coupling the frontend with a highly-tuned general-purpose constraint solver, thus eliminating the need for algorithmic implementation efforts from the ground up. I will discuss the necessary methodology, solving algorithms, and tools for building verifiers of the future [1, 2].

Joint work with Sergey Grebenshchikov, Nuno Lopes, and Corneliu Popeea.

References

1. Grebenshchikov, S., Gupta, A., Lopes, N.P., Popeea, C., Rybalchenko, A.: HSF(C): A Software Verifier Based on Horn Clauses - (Competition Contribution). In: Flanagan, C., König, B. (eds.) TACAS 2012. LNCS, vol. 7214, pp. 549–551. Springer, Heidelberg (2012)
2. Grebenshchikov, S., Lopes, N.P., Popeea, C., Rybalchenko, A.: Synthesizing software verifiers from proof rules. In: PLDI (2012)

A. Donaldson and D. Parker (Eds.): SPIN 2012, LNCS 7385, p. 22, 2012.