Verifying Functional Correctness of C Programs with VCC

Michał Moskal

Microsoft Research Redmond
michal.moskal@microsoft.com

Abstract. VCC [2] is an industrial-strength verification environment for low-level concurrent systems code written in C. VCC takes a program (annotated with function contracts, state assertions, and type invariants) and attempts to prove the correctness of these annotations. VCC's verification methodology [4] allows global two-state invariants that restrict update of shared state and enforces simple, semantic conditions sufficient for checking those global invariants modularly. VCC works by translating C, via Boogie [1] intermediate verification language, to verification conditions handled by the Z3 [5] SMT solver.

The environment includes tools for monitoring proof attempts and constructing partial counterexample executions for failed proofs and has been used to verify functional correctness of tens of thousands of lines of Microsoft's Hyper-V virtualization platform and of SYSGOs embedded real-time operating system PikeOS.

In this talk, I am going to showcase various tools that come with VCC: the verifier itself, VCC Visual Studio plugin, and Boogie Verification Debugger. I am going to cover the basics of VCC's verification methodology on various examples: concurrency primitives, lock-free data-structures, and recursive data-structures.

The sources and binaries of VCC are available for non-commercial use at `http://vcc.codeplex.com/`. A tutorial [3] is also provided. VCC can be also tried online at `http://rise4fun.com/Vcc`.

References

1. Barnett, M., Chang, B.-Y.E., DeLine, R., Jacobs, B., Leino, K.R.M.: Boogie: A modular reusable verifier for object-oriented programs. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.-P. (eds.) FMCO 2005. LNCS, vol. 4111, pp. 364–387. Springer, Heidelberg (2006)
2. Cohen, E., Dahlweid, M., Hillebrand, M.A., Leinenbach, D., Moskal, M., Santen, T., Schulte, W., Tobies, S.: VCC: A practical system for verifying concurrent C. In: Berghofer, S., Nipkow, T., Urban, C., Wenzel, M. (eds.) TPHOLs 2009. LNCS, vol. 5674, pp. 23–42. Springer, Heidelberg (2009)
3. Cohen, E., Hillebrand, M.A., Moskal, M., Schulte, W., Tobies, S.: Verifying C programs: A VCC tutorial. Working draft, `http://vcc.codeplex.com/`

M. Bobaru et al. (Eds.): NFM 2011, LNCS 6617, pp. 56–57, 2011.

4. Cohen, E., Moskal, M., Schulte, W., Tobies, S.: Local verification of global invariants in concurrent programs. In: Touili, T., Cook, B., Jackson, P. (eds.) CAV 2010. LNCS, vol. 6174, pp. 480–494. Springer, Heidelberg (2010)
5. de Moura, L.M., Bjørner, N.: Z3: An efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 337–340. Springer, Heidelberg (2008)