# Automated Verification of Concurrent Software*

Daniel Kroening

University of Oxford

**Abstract.** Effective use of concurrency is key to accelerating computations in a post frequency-scaling era. We review a research programme aimed at automated formal verification of a broad variety of concurrent systems. We briefly survey different forms of asynchronous concurrent computations, with a focus on multi-threaded, multi-core computation. We then highlight semantic and scalability challenges that arise when applying automated reasoning technology to this class of software.

We then discuss two very different techniques to address the challenges in this domain. The key insight behind the first technique is to exploit the symmetry that is inherent in many concurrent software programs: the programs execute a parametric number of identical threads, operating on different input data. Awareness of this design principle enables the application of symmetry reduction techniques such as counter abstraction, and encodings as Petri net coverability problems [2,6,4,3].

The second technique exploits the observation that asynchronous concurrent systems are frequently only very loosely synchronised. This gives rise to an encoding of the system using a set of constraints over partial orders. The constraints can be passed using a modern SAT/SMT solver, which gives rise to an effective bounded verification technique for asynchronous concurrent systems [1,5].

The research presented is joint work with Jade Alglave, Gerard Basler, Alastair Donaldson, Jim Grundy, Alexander Horn, Alexander Kaiser, Lihao Liang, Michele Mazzucchi, Tom Melham, Michael Tautschnig, Celina Val and Thomas Wahl.

## References

1. Alglave, J., Kroening, D., Tautschnig, M.: Partial Orders for Efficient Bounded Model Checking of Concurrent Software. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 141–157. Springer, Heidelberg (2013)
2. Basler, G., Mazzucchi, M., Wahl, T., Kroening, D.: Context-aware counter abstraction. Formal Methods in System Design (FMSD) 36(3), 223–245 (2010)
3. Donaldson, A., Kaiser, A., Kroening, D., Tautschnig, M., Wahl, T.: Counterexample-guided abstraction refinement for symmetric concurrent programs. Formal Methods in System Design (FMSD) 41(1), 25–44 (2012)

* Supported by ERC project 280053, EPSRC project EP/G026254/1 and the Semiconductor Research Corporation (SRC) under task 2269.002.

P.A. Abdulla and I. Potapov (Eds.): RP 2013, LNCS 8169, pp. 19–20, 2013.

4. Donaldson, A., Kaiser, A., Kroening, D., Wahl, T.: Symmetry-aware predicate abstraction for shared-variable concurrent programs. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 356–371. Springer, Heidelberg (2011)
5. Horn, A., Tautschnig, M., Val, C., Liang, L., Melham, T., Grundy, J., Kroening, D.: Formal co-validation of low-level hardware/software interfaces. In: Formal Methods in Computer-Aided Design, FMCAD (2013)
6. Kaiser, A., Kroening, D., Wahl, T.: Dynamic cutoff detection in parameterized concurrent programs. In: Touili, T., Cook, B., Jackson, P. (eds.) CAV 2010. LNCS, vol. 6174, pp. 645–659. Springer, Heidelberg (2010)