

Tool demonstration: The VerCORS Verification Toolset

Wytse Oortwijn

University of Twente, the Netherlands
w.h.m.oortwijn@utwente.nl

1 Abstract

VERCORS [2] is a tool set that specialises in automated verification of parallel and concurrent programs, written in high-level languages like (subsets of) Java and C, annotated with JML-like contracts. VERCORS is able to reason automatically about various concurrency constructs, including: (i) fork/join concurrency, as found in e.g. Java [4,6], (ii) GPU kernels with barriers and atomic operations in OpenCL [1], and (iii) compiler directives for automatic parallelisation in the context of OpenMP for C [3]. The main goal of VERCORS is to lift existing verification technology towards advanced concurrency constructs found in modern programming languages, rather than developing new verification technology from scratch. VERCORS therefore translates input verification problems to the intermediate language of VIPER [5]—the back-end verifier of VERCORS. VERCORS allows to reason about data-race freedom, memory safety, and functional correctness of input programs.

This tool demonstration gives an overview of the various concurrency features that VERCORS can reason about. We discuss the annotation language of VERCORS, as well as its permission system, and demonstrate how this is used to reason about parallelism and concurrency.

References

1. A. Amighi, S. Darabi, S. Blom, and M. Huisman. Specification and verification of atomic operations in GPGPU programs. In *SEFM*, volume 9276 of *LNCS*. Springer, 2015.
2. S. Blom, S. Darabi, M. Huisman, and W. Oortwijn. The VerCORS Tool Set: Verification of Parallel and Concurrent Software. In *iFM*, volume 10510 of *LNCS*, pages 102 – 110. Springer, 2017.
3. S. Darabi, S.C.C. Blom, and M. Huisman. A verification technique for deterministic parallel programs. In C. Barrett, M. Davies, and T. Kahsai, editors, *NASA Formal Methods (NFM)*, volume 10227 of *LNCS*, pages 247 – 264, 2017.
4. C. Haack, M. Huisman, and C. Hurlin. Permission-based separation logic for multithreaded java programs. *Nieuwsbrief van de Nederlandse Vereniging voor Theoretische Informatica*, 15:13–23, 2011.
5. Peter Müller, Malte Schwerhoff, and Alexander J. Summers. Viper: A Verification Infrastructure for Permission-Based Reasoning. In *Verification, Model Checking, and Abstract Interpretation*, pages 41–62. Springer Berlin Heidelberg, 2016.
6. W. Oortwijn, S. Blom, D. Gurov, M. Huisman, and M. Zaharieva-Stojanovski. An Abstraction Technique for Describing Concurrent Program Behaviour. In *VSTTE*, volume 10712 of *LNCS*, pages 191 – 209, 2017.