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