# Software Verification in the Presence of Generated Programs

2023-08-29 05:33:36

## Authors

Dirk Beyer

## Abstract

Our society and economy depends on correctly working computer systems. Correctness can be verified by formal verification, which is supported by many tools for software and hardware verification. Currently, there is a trend to more and more use AI-assisted programming, and preliminary studies show that software developers trust the suggestions from AI system more than they should, leading to errors in the software. This situation increases the importance of software verification in practice. In this work, we concentrate on two ingredients to make software verification more practical: exchangeable invariants and validation of verification witnesses. One of the main challenges is to automatically construct invariants that help to prove the correctness, and because of their key importance, we need to make those invariants first-class, interchangeable objects, not just code annotations. We further need validation tools that take as input a verification task (program and specification) and a candidate invariant, and double check whether the candidate invariant is indeed an inductive and safe program invariant, and if not, reliably reject the candidate invariant. Similarly, we need exchangeable error reports, in order to make it easier for developers to understand and locate a deviation of the program behavior from the specification.

## Keywords

Correctness Certification, Software Verification, Witness Validation