Local Completeness for Program Correctness and Incorrectness

Roberto Bruni

Computer Science Department, University of Pisa, Italy

Abstract

Program correctness techniques aim to prove the absence of bugs, but can yield false alarms because they tend to over-approximate program semantics. Vice versa, program incorrectness methods are aimed to detect true bugs, without false alarms, but cannot be used to prove correctness, because they under-approximate program semantics. In this invited talk we will overview our ongoing research on the use of the abstract interpretation framework to combine under- and over-approximation in the same analysis and distill a logic for program correctness and incorrectness.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Programming logic; Theory of computation → Hoare logic; Theory of computation → Abstraction

Keywords and phrases Program analysis, program verification, Hoare logic, incorrectness logic, abstract interpretation, local completeness

Digital Object Identifier 10.4230/LIPIcs.CALCO.2023.2

Category Invited Talk

Funding Research supported by Italian MIUR PRIN2017 project no. 201784YSZ5 “Analysis of Program Analyses (ASPRA)”, INdAM – GNCS Project, code no. CUP_E53C22001930001 and by a Meta research gift.

1 Extended abstract

Floyd-Hoare logic for program correctness [12, 13] was an eye-opening contribution to the use of over-approximation in program verification aimed to prove the absence of errors. From the perspective of programmers, the benefit of the feedback provided by program correctness analyses within the software development ecosystem is appreciated if warnings are reported early and truly [11]. The use of over-approximation is necessary to make the correctness problem tractable and to develop automatic tools, but inevitably it introduces some imprecision. As a consequence verification tools can produce false alarms, i.e., potential errors that are reported by the analysis but that do not correspond to any execution.

Possibly inspired by the consequence rule of Reverse Hoare logic [10], Peter O’Hearn’s recent studies on the use of under-approximation in program analysis have led to the definition of a logic for program incorrectness [17, 18, 19, 16, 14], which, dualising the over-approximation approach of Hoare logic, can be used to exhibit the presence of errors, without false alarms, but not for proving program correctness.

In this talk we will overview our ongoing research [3, 5, 4, 1, 6, 15, 2] on the use of the abstract interpretation framework [8, 9, 7] to combine under- and over-approximation in the same analysis and distill a logic for program correctness and incorrectness. Any triple provable in the logic can be used either to guarantee the correctness of the program or to expose some (true) errors. A key role is played by the notion of locally complete abstraction that provides the necessary proof obligations in logic derivations. Notably different abstract domains can be combined in the same derivation and the logic can be instantiated to different settings, like imperative programming languages and strategy languages for rewrite systems.
References


