Algebraic Reasoning for (Un)Solvable Loops

Laura Kovács
TU Wien, Austria

Abstract

Loop invariants describe valid program properties that hold before and after every loop iteration. As such, loop invariants are the workhorses in formalizing loop semantics and automating the formal analysis and verification of programs with loops.

While automatically synthesizing loop invariants is, in general, an uncomputable problem, when considering only single-path loops with linear updates (linear loops), the strongest polynomial invariant is in fact computable [5, 9, 6, 3]. Yet, already for loops with "only" polynomial updates, computing the strongest invariant has been an open challenge since 2004 [8].

In this invited talk, we first present computability results on polynomial invariant synthesis for restricted polynomial loops, called *solvable* loops [11]. Key to solvable loops is that one can automatically compute invariants from closed-form solutions of algebraic recurrence equations that model the loop behaviour [6, 4]. We also establish a technique for invariant synthesis for classes of loops that are not solvable, termed *unsolvable* loops [1].

We next study the limits of computability in deriving the (strongest) polynomial invariants for *arbitrary* polynomial loops. We prove that computing the strongest polynomial invariant of arbitrary, single-path polynomial loops is very hard [10] – namely, it is at least as hard as the Skolem problem [2, 12], a prominent algebraic problem in the theory of linear recurrences. Going beyond single-path loops, we show that the strongest polynomial invariant is uncomputable already for multi-path polynomial loops with arbitrary quadratic polynomial updates [7].

2012 ACM Subject Classification Theory of computation → Invariants; Theory of computation → Program verification; Theory of computation → Algebraic semantics

Keywords and phrases Symbolic Computation, Formal Methods, Loop Analysis, Polynomial Invariants

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.4

Category Invited Talk

Funding We acknowledge funding fom the European Research Council Consolidator Grant ARTIST 10100268, the Vienna Science and Technology Fund WWTF 10.47379/ICT19018 grant ProbInG, the Austrian Science Fund (FWF) project W1255-N23, the SecInt Doctoral College funded by TU Wien, and the European Union's Horizon 2020 research and innovation programme LogiCS@TUWIen under grant agreement No 101034440.

Acknowledgements This talk is based on joint works with a number of authors, including Daneshvar Amrollahi (TU Wien alumni), Ezio Bartocci (TU Wien), George Kenison (TU Wien), Marcel Moosbrugger (TU Wien), Julian Müllner (TU Wien), Miroslav Stankovic (TU Wien), and Anton Varonka (TU Wien).

References

1 Daneshvar Amrollahi, Ezio Bartocci, George Kenison, Laura Kovács, Marcel Moosbrugger, and Miroslav Stankovic. Solving Invariant Generation for Unsolvable Loops. In *Proc. of SAS*, 2022. doi:10.1007/978-3-031-22308-2_3.

2 Graham Everest, Alfred J. van der Poorten, Igor E. Shparlinski, and Thomas Ward. *Recurrence Sequences.* Mathematical Surveys and Monographs. American Mathematical Society, 2003.

3 Ehud Hrushovski, Joël Ouaknine, Amaury Pouly, and James Worrell. Polynomial Invariants for Affine Programs. In *Proc. of LICS*, 2018. doi:10.1145/3209108.3209142.

48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023).
Editors: Jérôme Leroux, Sylvain Lombardy, and David Peleg; Article No. 4; pp. 4:1–4:2
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

4 Andreas Humenberger, Maximilian Jaroschek, and Laura Kovács. Automated Generation of Non-Linear Loop Invariants Utilizing Hypergeometric Sequences. In *Proc. of ISSAC*, 2017. doi:10.1145/3087604.3087623.

5 Michael Karr. Affine Relationships Among Variables of a Program. *Acta Inform.*, 1976. doi:10.1007/BF00268497.

6 Laura Kovács. Reasoning Algebraically About P-Solvable Loops. In *Proc. of TACAS*, 2008. doi:10.1007/978-3-540-78800-3_18.

7 Laura Kovács and Anton Varonka. What Else is Undecidable About Loops? In *Proc. of RAMiCS*, 2023. doi:10.1007/978-3-031-28083-2_11.

8 Markus Müller-Olm and Helmut Seidl. A Note on Karr's Algorithm. In *Proc. of ICALP*, 2004. doi:10.1007/978-3-540-27836-8_85.

9 Markus Müller-Olm and Helmut Seidl. Computing Polynomial Program Invariants. *Inf. Process. Lett.*, 2004. doi:10.1016/j.ipl.2004.05.004.

10 Julian Müllner. Exact Inference for Probabilistic Loops. Master's thesis, TU Wien, 2023.

11 Enric Rodríguez-Carbonell and Deepak Kapur. Automatic Generation of Polynomial Loop Invariants: Algebraic Foundations. In *Proc. of ISSAC*, 2004. doi:10.1145/1005285.1005324.

12 Terrence Tao. *Structure and Randomness.* American Mathematical Society, 2008.