Formal Methods in System Design (2021) 58:469–470
https://doi.org/10.1007/s10703-021-00386-0

Special Issue on Syntax-Guided Synthesis Preface

Dana Fisman[1] · Rishabh Singh[1] · Armando Solar-Lezama[1]

Accepted: 10 December 2021 / Published online: 28 February 2022

Syntax-guided synthesis has emerged as a practical solution to the computationally hard problem of program synthesis. *Program synthesis*, dating back to [3,5,7] asks the problem whether it is possible to derive an executable implementation from a specification of a system given as a logical formula. The exact complexity of the problem depends on the type of the logical formalism and the corresponding type of systems and is either undecidable or of very high computational complexity. To overcome the computational hardness a thread of research works in program synthesis and program optimization emerged, in which in addition to the logical formula describing the functionality of the desired implementation, some syntactic set of constraints is given in order to limit the search space of viable solutions. Following the uniformization of these approaches into a coherent logical formalism [1,2] this area of research is now referred to as *syntax-guided synthesis*.

This special issue of the Journal on Formal Methods in System Design features three papers that span different directions from the spectrum of research evolving around syntax-guided synthesis.

- *Refutation-based synthesis in SMT* [8] introduces the first program synthesis engine implemented inside an SMT solver. While prior work in the area used SMT solvers as a black-box, this work shows the benefits one can gain by accessing the inners of an SMT solver to solve the problem at hand.
- *Alloy* a general-purpose higher-order relational constraint solver* [6] presents a general-purpose constraint solver that augments syntax-guided synthesis from first order-logics to higher-order logics.
- *An empirical study of adaptive concretization for parallel program synthesis* [4] presents a technique to parallelize the search of assignments for unknown variables in a synthesis problem by identifying that some variables are best suited for explicit search and some are best suited for symbolic search.

References

1. Alur R, Bodík R, Dallal E, Fisman D, Garg P, Juniwal G, Kress-Gazit H, Madhusudan P, Martin MMK, Raghothaman M, Saha S, Seshia SA, Singh R, Solar-Lezama A, Torlak E, Udupa A (2015) Syntax-guided synthesis. In: Irlbeck M, Peled DA, Pretschner A (eds) Dependable software systems engineering, vol 40.

✉ Dana Fisman
firstname@cs.bgu.ac.il

[1] Ben-Gurion University, Beersheba, Israel

NATO science for peace and security series, D: information and communication Security. Amsterdam, IOS Press, pp 1–25

2. Alur R, Singh R, Fisman D, Solar-Lezama A (2018) Search-based program synthesis. Commun ACM 61(12):84–93
3. Church Alonzo (1963) Application of recursive arithmetic to the problem of circuit synthesis. J Symbol Logic 28(4):289–290
4. Jeon J, Qiu X, Solar-Lezama A, Foster JS (2017) An empirical study of adaptive concretization for parallel program synthesis. Formal Methods Syst Des 50(1):75–95
5. Manna Z, Waldinger Richard J (1971) Toward automatic program synthesis. Commun ACM 14(3):151–165
6. Milicevic Aleksandar, Near Joseph P, Kang Eunsuk, Jackson Daniel (2019) Alloy*: a general-purpose higher-order relational constraint solver. Formal Methods Syst Des 55(1):1–32
7. Pnueli A, Rosner R: A framework for the synthesis of reactive modules. In: Friedrich H Vogt, editor, *Concurrency 88: International Conference on Concurrency, Hamburg, FRG, October 18-19, 1988, Proceedings*, volume 335 of *Lecture Notes in Computer Science*, pages 4–17 Springer, (1988)
8. Reynolds A, Kuncak V, Tinelli C, Barrett CW, Deters Morgan (2019) Refutation-based synthesis in SMT. Formal Methods Syst Des 55(2):73–102