Data-Driven Design and Evaluation of SMT Meta-Solving Strategies

Malte Mues[1] Falk Howar[1,2]

Abstract: The 36th IEEE/ACM International Conference on Automated Software Engineering (2021) accepted the paper 'Data-Driven Design and Evaluation of SMT Meta-Solving Strategies: Balancing Performance, Accuracy, and Cost' [MH21a] and selected it for an ACM SIGSOFT Distinguished Paper Award. The paper presents four generally applicable patterns for the combination of multiple SMT decision procedures in a meta-solving strategy and demonstrates how a meta-solving strategy for string constraints can be developed in a data-driven approach based on these patterns: The paper cleans up and merges existing collections of SMT benchmarks in string theory solving to evaluate and compare derived meta-solving strategies. Notably, we can demonstrate on the available data that commonly used strategies as earliest returning SMT solver do not always return the most reliable result if all available SMT solvers are combined. Instead, cross-checking strategies work slightly better at moderate overhead.

Keywords: SMT Solving; Formal Methods; Software Verification

1 Integration Patterns for Formal Methods

Advances in formal methods are often implemented in research tools and libraries, which are then used in other analyses or commercial applications. The most prominent example of the past decade are constraint solvers that are used in many applications today. While the underlying decision procedures are usually sound, implementations optimize for different aspects of a problem and — just like other software — can contain bugs. One strategy for making formal methods robust in practice is redundancy: i.e., using multiple analyses of a similar type. Here, again, constraint solvers can serve as an example: so-called portfolios of solvers are used to optimize the number of conclusive verdicts or response times by analyzing a problem with multiple solvers. The paper [MH21a] identifies four basic patterns for integrating different formal analyses that have been proposed in the literature (c.f. [WHDM09, HJM19, Ri17, MH21b]).

The four patterns, sketched in Figure 1, are briefly described below. The *Majority Vote* (middle right) integrates different constraint solvers (or multiple instances of the same solver) *A* to *N* by analyzing a problem instance with all solvers. A majority vote is computed as a final verdict. The *Earliest Verdict* (middle left) integrates constraint solvers *A* to *N* by analyzing a problem instance with all solvers. The first obtained verdict is used as the final

[1] TU Dortmund University, 44227 Dortmund, Deutschland {malte.mues,falk.howar}@tu-dortmund.de
[2] Fraunhofer ISST, 44227 Dortmund, Deutschland

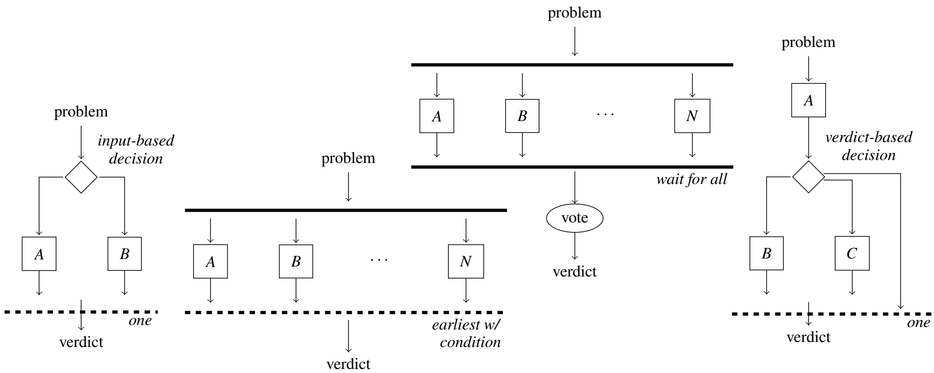

Fig. 1: Four Basic integration patterns for formal anayses [MH21a].

verdict. The *Verdict-based Second Attempt* (right) integrates multiple constraint solvers by first calling solver *A* and then deciding on a next step on the basis of the obtained verdict. Potential next steps are invoking another solver on the original problem or using another solver or tool to validate the result computed by *A*. The *Feature-based Attempt* (left) integrates multiple solvers by selecting one constraint solver for a problem instance based on features of the problem instance. In the paper, strengths and drawbacks of the patterns are discussed and performance of patterns is analyzed for integrating string solvers.

2 Data Availability

The reproduction package for the paper with all experimental data and scripts for computing results is publicly available at `https://zenodo.org/record/5102174`.

Bibliography

[HJM19] Howar, Falk; Jabbour, Fadi; Mues, Malte: JConstraints: a library for working with logic expressions in Java. In: Models, Mindsets, Meta: The What, the How, and the Why Not?, pp. 310–325. Springer, 2019.

[MH21a] Mues, Malte; Howar, Falk: Data-Driven Design and Evaluation of SMT Meta-Solving Strategies: Balancing Performance, Accuracy, and Cost. In: Proc. of ASE, 2021. ACM (to appear), 2021.

[MH21b] Mues, Malte; Howar, Falk: JDart: Portfolio solving, breadth-first search and smt-lib strings (competition contribution). Proc. of TACAS 2021, 12652:448–452, 2021.

[Ri17] Riener, Heinz; Haedicke, Finn; Frehse, Stefan; Soeken, Mathias; Große, Daniel; Drechsler, Rolf; Fey, Goerschwin: metaSMT: focus on your application and not on solver integration. STTT, 19(5):605–621, 2017.

[WHDM09] Wintersteiger, Christoph M; Hamadi, Youssef; De Moura, Leonardo: A concurrent portfolio approach to SMT solving. In: Proc. of CAV 2009. Springer, pp. 715–720, 2009.