A Data Driven Approach for Skolem Function Synthesis*

Priyanka Golia[1,2], Kuldeep S. Meel[1], and Subhajit Roy[2]

[1] School of Computing, National University of Singapore, Singapore
[2] Computer Science and Engineering, Indian Institute of Technology Kanpur, India

Abstract. Synthesizing Skolem functions is one of the challenging problems in Computer Science. It has seen multiple proposals, including incremental determination, decomposition techniques from knowledge compilation, and counterexample guided refinement techniques via self-substitutions. In this work, we propose a novel data-driven approach to synthesis that employs constrained sampling techniques for generation of data, machine learning for candidate Skolem functions, and automated reasoning to verify and refine to generate Skolem functions. Our approach achieves significant performance improvements by solving 63/609 benchmarks that could not be solved by any of the existing synthesis tools.

Given a propositional formula $\exists Y F(X,Y)$, Skolem functiona synthesis is to synthesis a function Ψ such that $\exists Y F(X,Y)$ is equivalent to $F(X,\Psi(X))$. It has many applications in different areas like certified QBF solving, automated program repair and synthesis, and cryptography. In this work, we propose a synergistic interaction of learning and formal methods based techniques to synthesize Skolem functions. We design a data-driven approach, Manthan, to synthesize Skolem function that utilizes a novel sampling algorithm to gather data, apply a machine learning algorithm to learn the candidate Skolem functions, and successively refines it via a proof-guided refinement algorithm.

- **Data Generation:** The data here represents the relationship between universally and existentially quantified variables. We use a subset of satisfying assignment of $F(X,Y)$ as data. We view the problem of synthesizing Skolem function as the classification of valuation of the existentially quantified variable over a data. We want to tailor our sampling subroutines to allow the discovery of Skolem functions with *small* description. To this end, we design a novel biased sampling technique that samples the existentially quantified variables at the cost of the universally quantified variables.
- **Learning Candidate Skolem Function:** Manthan learns candidate Skolem functions as decision trees with data projected on universally quantified variables as features, and existentially quantified variables as labels. Candidate

* This is an extended abstract of our paper: Manthan: A Data Driven Approach for Skolem Function Synthesis, CAV-20

Skolem functions can be represented as the disjunction of all the paths from the root to the leaves in the learnt decision tree. We call this the LearnSkF phase of Manthan.

- **Proof Based Refinement:** Manthan uses a MaxSAT solver to identify the erring candidates Skolem functions that may need to be repaired. It utilizes UNSAT core from the infeasibility proofs of the candidate function meeting its specifications to generate a repair formula. Manthan updates the candidate Skolem function with its repair formula. The candidate Skolem functions converges to the actual function through a sequence of such minor repair. We call this the Refine phase of Manthan.

We compared Manthan performance with the state of the art tools, viz. BFSS [3], C2syn [2], and CADET [4] on a set of benchmarks drawn from the datasets QBFEval-17-18 [1], Disjunctive, Factorization and Arithmetic data set [3]. Figure 1 shows that Manthan significantly improves upon state of the art, and solves 356 benchmarks. To put the runtime performance statistics in a broader context, the number of benchmarks solved by techniques developed over the past five years range from 206 to 280, i.e., a difference of 74, which is same as an increase of 76 (i.e., from 280 to 356) due to Manthan; in particular, Manthan solves 60 more benchmarks that could not be solved by any of the tools.

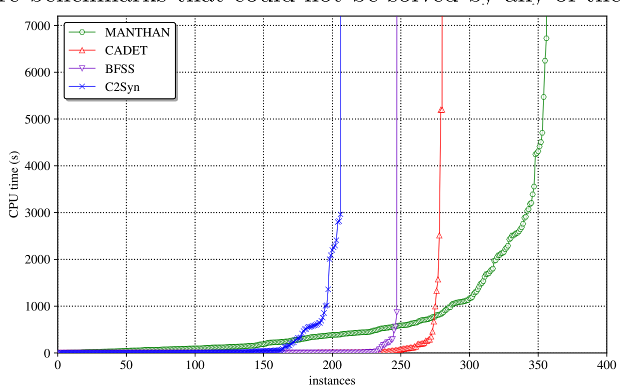

Fig. 1: Manthan vis a vis camparision with state of the art tools

Our approach achieves significant performance improvements and opens up several interesting directions for future work at the intersection of machine learning, constrained sampling, and automated reasoning.

References

1. QBF solver evaluation portal 2018, http://www.qbflib.org/qbfeval18.php
2. Akshay, S., Arora, J., Chakraborty, S., Krishna, S., Raghunathan, D., Shah, S.: Knowledge compilation for boolean functional synthesis. In: Proc. of FMCAD (2019)
3. Akshay, S., Chakraborty, S., Goel, S., Kulal, S., Shah, S.: Whats hard about boolean functional synthesis? In: Proc. of CAV (2018)
4. Rabe, M.N., Tentrup, L., Rasmussen, C., Seshia, S.A.: Understanding and extending incremental determinization for 2QBF. In: Proc. of CAV (2018)