Encodings of Reactive Synthesis

Peter Faymonville[1], Bernd Finkbeiner[1], Markus N. Rabe[2], and Leander Tentrup[1]

[1] Saarland University
[2] University of California, Berkeley

Abstract. In this talk, we present and compare several encodings of the bounded synthesis problem for linear-time temporal logic (LTL). The bounded synthesis problem for natural bound n is to decide whether there exists a strategy (generated by a transition system with n states) that satisfies an LTL specification—and in the positive case to construct such a strategy. We give an overview of previously studied encodings that use SMT, antichains, and BDDs, as well as new encodings using (quantified) propositional logics. Furthermore, we evaluate the constraint based approaches (SMT, SAT, QBF, etc.) with respect to solving time and implementation quality using certifying theory solvers.

1 Extended Abstract

Synthesis is the task of creating correct-by-construction implementations from formal specifications, thus avoiding the need for manual implementations. In recent years, synthesis has gained a lot of attention and modern synthesis tools emerged [1–4]. Last year, this development culminated in the first competition of synthesis tools [7].

In this talk, we consider the bounded synthesis [6] problem, that is the problem of synthesizing a strategy of size n, such that the strategy satisfies the LTL specification φ. For an LTL formula φ, we assume a partitioning into variables O that are controllable by the strategy and variables I that are given by the environment. A strategy $f : (2^I)^* \to 2^O$ maps sequences of valuations from the environment to a valuation of the controllable variables. We represent strategies as finite-state transition systems and identify the size of a strategy with the size of the transition system.

Given such a specification φ, we build a universal co-Büchi automaton $\mathcal{U}_\varphi$. A transition system is accepted by $\mathcal{U}_\varphi$ if each run in the unique run graph on $\mathcal{U}_\varphi$ has only finitely many visits to the rejecting states of $\mathcal{U}_\varphi$. The acceptance of a finite-state transition systems on $\mathcal{U}_\varphi$ can be characterized by the existence of an annotation on the product of transition system and automaton [6]. This annotation maps a pair (s, q), where s is a state in the transition system and q is a state in the automaton, to the number of maximal visits to rejecting states on all runs that lead to (s, q). In the original formulation [5], the labeling and transition functions of the transition system as well as the correct annotation were encoded as SMT constraints. The SMT encoding uses uninterpreted functions and a theory that supports ordering constraints (like the theory of integers).

We show how to modify the encoding to use only uninterpreted functions and propositional constraints. Based on this modification, we give a reduction to the satisfiability problem for quantified Boolean formulas (QBF) and propositional satisfiability (SAT). Let $S = \{s_1, \dots, s_n\}$ be the number of states in the transition system. The quantifiers in the QBF encoding make the transition function symbolic in the inputs, i.e., a part of the quantification header has the form $\forall \boldsymbol{i}.\, \exists \boldsymbol{t}_{s,s'}$ for all $s, s' \in S$, meaning that there is a transition from state s to s' in the transition system, if the Skolem function $f_{t_{s,s'}}$ evaluates to true for the given environment input $\boldsymbol{i}$.

We investigated experimentally which encoding is best given the current state of solver technology. With regard to the SMT encoding, we compare different theories to encode the ordering constraints and different levels of quantifications. For the propositional encoding, we compare the QBF encoding, which uses quantification for input-symbolic transition functions, with the variant that unrolls the quantification to a pure SAT encoding.

Modern solvers have the ability to construct models from satisfiable queries, e.g., Skolem functions in the case of QBF. As the existence of a transition system is encoded in the bounded synthesis query, we can easily construct an implementation using certifying solvers. We compare the quality of these implementations with respect to the different encodings, ranging from models generated by an SMT solver, Skolem functions extracted from QBF proofs, and assignments given by a SAT solver.

References

1. Bloem, R., Egly, U., Klampfl, P., Könighofer, R., Lonsing, F.: SAT-based methods for circuit synthesis. In: Proceedings of FMCAD. pp. 31–34 (2014)
2. Bloem, R., Gamauf, H., Hofferek, G., Könighofer, B., Könighofer, R.: Synthesizing robust systems with RATSY. In: Proceedings of SYNT. pp. 47–53 (2012)
3. Bohy, A., Bruyère, V., Filiot, E., Jin, N., Raskin, J.: Acacia+, a tool for LTL synthesis. In: Proceedings of CAV. pp. 652–657 (2012)
4. Ehlers, R.: Symbolic bounded synthesis. Formal Methods in System Design 40(2), 232–262 (2012)
5. Finkbeiner, B., Schewe, S.: SMT-based synthesis of distributed systems. In: Proceedings of AFM (2007)
6. Finkbeiner, B., Schewe, S.: Bounded synthesis. STTT 15(5-6), 519–539 (2013)
7. Jacobs, S., Bloem, R., Brenguier, R., Ehlers, R., Hell, T., Könighofer, R., Pérez, G.A., Raskin, J., Ryzhyk, L., Sankur, O., Seidl, M., Tentrup, L., Walker, A.: The first reactive synthesis competition (SYNTCOMP 2014). CoRR abs/1506.08726 (2015), http://arxiv.org/abs/1506.08726