Hardware Model Checking Certificates

Emily Yu Nils Froleyks
Johannes Kepler University Linz, Austria

Armin Biere Mathias Fleury
University of Freiburg, Germany

I. Certifying model checking results

The proposed benchmark set is obtained from our recent work on certifying k-induction-based model checking [1]. The model checking technique k-induction [2] is widely used for verification. A safety property is said to be k-inductive iff it satisfies the following two cases: it holds for $k-1$ consecutive steps originating from the initial states; if it holds for $k-1$ steps of unrolling, it also holds at the next state after one transition.

The key idea of certifying a model checking result in a generic form is to generate an inductive invariant as a proof certificate which implies the safety property in the given model. The certification framework proposed in [1] reduces the certification problem to five SAT checks and a one-alternation QBF, allowing certification at a low complexity. In a nutshell, the approach extends the given model to a larger circuit (namely k-witness circuit) with an inductive invariant. Two SAT checks and one QBF check are used to verify the so-called combinational simulation relation between the original circuit and the k-witness circuit. Furthermore, another three SAT checks are generated for checking the inductive invariant in the k-witness circuit.

The size of the k-witness circuit is in principle linear in the size of the original circuit and the value of k. One of the most essential features is that it includes k copies of the original machine, and allows multiple ways of initialisation. The new property in the k-witness circuit (the inductive invariant) is composed of five sub-properties.

To verify $\phi(I, L)$ (a formula over the inputs and latches) is indeed an inductive invariant in the k-witness circuit, the following formulas are generated:

- Initiation check ($R(L) \Rightarrow \phi(I, L)$): the inductive invariant must hold at all initial states.
- Consistency check ($\phi(I, L) \Rightarrow P(I, L)$): the inductive invariant must hold at all good states.
- Consecution check ($U_1 \wedge \phi(I_0, L_0) \Rightarrow \phi(I_1, L_1)$): the inductive invariant is preserved during one transition.

II. Generated instances

The certification approach is implemented into a toolkit Certifaiger [1], which takes as inputs a model in AIGER format [3] and a value of k which is usually provided by a model checker (here we used McAiger [4]). The toolkit originally uses Kissat [5] as the underlying SAT solver, however, in this paper we modified it to use MiniSAT instead.

All benchmarks are obtained by running against a subset of HWMCC'2010 [6] benchmarks which McAiger successfully terminated with. We found 7 SAT formulas for which MiniSAT experienced a timeout of 15 minutes. (However, they were originally solved by Kissat and are unsatisfiable.) Among those, one is an initiation check, and the rest are consecution checks. We then add these 7 formulas to our benchmark set.

As the nature of k-induction, if a property is m-inductive in a model for some arbitrary m it is also n-inductive in the same model for any n such that $n > m$. Therefore we obtained another 3 benchmarks from the same set by scaling the inductive depths to 500.

As for the satisfiable instances, we added the 4 instances with the same inductive depth 80 to the benchmark set, which originally were timed out on Kissat mentioned in the paper [1] from the pj20 family. We used McAiger to inspect the inductive depths, which confirms that the values of k are greater than 80, making the consecution check fail thus the formulas satisfiable. With the same logic, we obtained further 6 benchmarks by scaling the inductive depths. The names of the benchmarks are composed by following the convention: original benchmark name + "_k" + inductive depth. There are two exceptions starting with the name "bobsmdct_init", where "bobsmdct" is the model name and "_init" is for explicitly stating initiation checks.

References

[1] E. Yu, A. Biere, and K. Heljanko, "Progress in certifying hardware model checking results," in *CAV (2)*, ser. Lecture Notes in Computer Science, vol. 12760. Springer, 2021, pp. 363–386.

[2] M. Sheeran, S. Singh, and G. Stålmarck, "Checking safety properties using induction and a SAT-solver," in *FMCAD*, ser. Lecture Notes in Computer Science, vol. 1954. Springer, 2000, pp. 108–125.

[3] A. Biere, K. Heljanko, and S. Wieringa, "AIGER 1.9 and beyond," Institute for Formal Models and Verification, Johannes Kepler University, Altenbergerstr. 69, 4040 Linz, Austria, Tech. Rep. 11/2, 2011.

[4] A. Biere and R. Brummayer, "Consistency checking of all different constraints over bit-vectors within a SAT solver," in *FMCAD*. IEEE, 2008, pp. 1–4.

[5] A. Biere, M. Fleury, and M. Heisinger, "CaDiCaL, Kissat, Paracooba entering the SAT Competition 2021," in *Proc. of SAT Competition 2021 – Solver and Benchmark Descriptions*, ser. Department of Computer Science Report Series B, T. Balyo, N. Froleyks, M. Heule, M. Iser, M. Järvisalo, and M. Suda, Eds., vol. B-2021-1. University of Helsinki, 2021, pp. 10–13.

[6] A. Biere and K. Claessen, "Hardware model checking competition 2010," 2010, http://fmv.jku.at/hwmcc10/.