Appears in A. Balint, A. Belov, M.J.H. Heule, and M. Järvisalo (eds.), *Proceedings of SAT Competition 2013: Solver and Benchmark Descriptions*, volume B-2013-1 of Department of Computer Science Series of Publications B, University of Helsinki 2013. ISBN 978-952-10-8991-6.

Equivalence checking of HWMCC 2012 Circuits

Armin Biere

Institute for Formal Models and Verification
Johannes Kepler University Linz, Austria

Marijn J.H. Heule

Department of Computer Science,
The University of Texas at Austin, United States

Matti Järvisalo

HIIT & Department of Computer Science
University of Helsinki, Finland

Norbert Manthey

Institute of Artificial Intelligence
Technische Universität Dresden, Germany

INTRO

A miter encodes an equivalence check of two Boolean circuits. This is encoded as a combinatorial problem searching for an input for these circuits such that their output is different. Fig 1 shows an illustration of a miter: Two circuits have the same inputs and there is an exclusive-OR (XOR) for each output of the circuits. If the output of one of these XORs can be assigned to true, a certificate is found that shows that the circuits are not equivalent. Miters are generally used as follows: one of the two circuits is an optimized variant of the other one. If the miter has no solution (unsatisfiable), it means that the circuits are equivalent and that the optimization is valid.

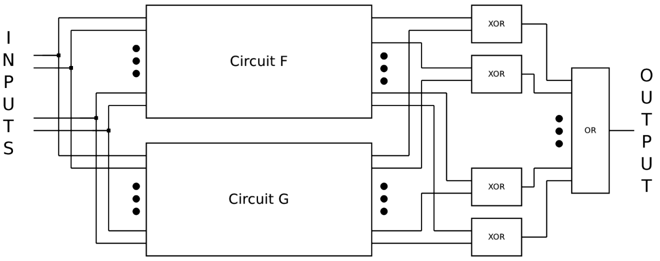

Fig. 1. Illustration of a miter.

GENERATION OF THE BENCHMARKS

We generated two types of miters using the circuits described in the AIGER benchmarks of the hardware model checking competition (HWMCC) 2012[1]. We used circuits with both single and multiple bad state properties (the latter also contain environment constraints). We used `aigmiter` for constructing *combinational* miters, e.g. next state functions of flip-flops are treated as outputs, and then translated them to CNF with `aigtocnf`.

These tools are available from http://fmv.jku.at/aiger. Note that these benchmarks are trivial on the AIG level and can simply be solved by structural hashing. Further, the benchmarks, scripts for generating these miters, as well as log files of the generation process are available from http://fmv.jku.at/miters.

NON-OPTIMIZED MITERS

The first type of miter was constructed using two copies of the same circuit. On the AIG level, these benchmarks are trivial. We showed that these benchmarks can also be solved on the CNF level by the preprocessing technique hyper binary resolution [1], [2] (HBR). However, some of the non-optimized miters can be hard for SAT solvers.

OPTIMIZED MITERS

The ABC tool [3] was used to construct optimized circuits (using the `dc2` command). The miters of this second type encode that the original circuit is equivalent to the optimized one. These benchmarks are much harder than the non-optimized miters.

REFERENCES

[1] F. Bacchus and J. Winter, "Effective preprocessing with hyper-resolution and equality reduction," in *Proc. SAT 2003*, ser. LNCS, vol. 2919. Springer, 2004, pp. 341–355.

[2] M. Heule, M. Järvisalo, and A. Biere, "Revisiting hyper binary resolution," in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, ser. Lecture Notes in Computer Science, C. Gomes and M. Sellmann, Eds. Springer Berlin Heidelberg, 2013, vol. 7874, pp. 77–93. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38171-3_6

[3] R. K. Brayton and A. Mishchenko, "Abc: An academic industrial-strength verification tool," in *CAV*, ser. Lecture Notes in Computer Science, T. Touili, B. Cook, and P. Jackson, Eds., vol. 6174. Springer, 2010, pp. 24–40.

[1] see http://fmv.jku.at/hwmcc12/ for details