Practical CNF Interpolants Via BDDs

Alexander Legg[1] Nina Narodytska[12] Leonid Ryzhyk[12]

[1] NICTA* and UNSW, Sydney, Australia
alexander.legg@nicta.com.au
[2] University of Toronto, Toronto, Canada
ninan@cse.unsw.edu, leonidr@cse.unsw.edu

Craig interpolation has been recently shown to be useful in a wide variety of problem domains. One use is in strategy extraction for two player games, as described in our accompanying submission. However, interpolation is not without its drawbacks. It is well-known that an interpolant may be very large and highly redundant. Subsequent use of the interpolant requires that it is transformed to CNF or DNF, which will further increase its size.

We present a new approach to handling both the size of interpolants and transformation to clausal representation. Our approach relies on the observation that in many real-world applications, interpolants are defined over a relatively small set of variables. Additionally, in most cases there likely exists a compact representation of the interpolant in CNF. For instance, in our application to games an interpolant represents a set of winning states that is likely to have a simple structure.

Our approach is to produce a compact CNF or DNF representation from an interpolant by first compiling it to a BDD. We expect that in many practical applications of interpolants this computation can be performed efficiently, resulting in a compact BDD representation of the interpolant. This representation can be used to produce compact CNF or DNF using existing efficient algorithms.

We will present encouraging initial results from our application of this technique. We have produced very compact CNF from very large interpolants. Part of our future work is to compare this technique to other solutions, including BDD-sweeping [2, 1]. Sweeping is more efficient than our approach but does not produce as compact a circuit nor does it enable an efficient conversion to CNF or DNF.

References

[1] Z. Hassan, Y. Zhang, and F. Somenzi. A study of sweeping algorithms in the context of model checking. In *International Workshop on Design and Implementation of Formal Tools and Systems*, 2011.

[2] A. Kuehlmann and F. Krohm. Equivalence checking using cuts and heaps. In *Design Automation Conference*, pages 263–268, 1997.

*NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.