ACM Communications in Computer Algebra, Vol. 45, No. 2, Issue 176, June 2011

New Approaches to Boolean Quantifier Elimination

Christoph Zengler, Andreas Kübler and Wolfgang Küchlin
Wilhelm-Schickard-Institute for Informatics
University of Tübingen
Tübingen, Germany, 72076
{zengler,kuebler,kuechlin}@informatik.uni-tuebingen.de

Abstract

We present four different approaches for existential Boolean quantifier elimination, based on model enumeration, resolution, knowledge compilation with projection, and substitution. We point out possible applications in the area of verification and we present preliminary benchmark results of the different approaches.

1 Introduction

In the last decade formal verification has benefitted greatly from the developments in the Boolean logic community. Symbolic or bounded model checking relies heavily on SAT solvers and BDD packages. In our work we want to incorporate tools and techniques from this community to develop new algorithms for Boolean quantifier elimination (BQE). In this paper we present four different approaches for existential BQE.

BQE is already used in many different contexts in the area of verification and we additionally spotted some new interesting applications. We can only mention a few examples here: (1) BQE lies at the heart of one of the core operations in symbolic model checking: image computation. (2) With BQE we can compute all paths that lead to an error state in a state transition system. That can be used to construct metrics to measure the severity of errors in e.g. software systems. (3) One can utilize BQE to find generalized counter examples for errors, i.e. what properties do all inputs have in common that lead to an error. (4) BQE can be used to compute Craig interpolants which have recently gained much interest in the verification community as they facilitate e.g. automatic predicate abstraction as well as purely SAT-based unbounded model checking.

2 Approaches for Boolean Quantifier Elimination

Model Enumeration with Projection Boolean model enumeration means listing all variable assignments which turn a formula to true. Current tools [1] can enumerate models w. r. t. a given subset of variables. One can easily transform this set of satisfying assignments into a new formula which is a quantifier-free equivalent to the original formula.

Variable Elimination by Clause Distribution The ideas for this approach go back to Davis and Putnam and were recently used for variable elimination in the QBF Solver Quantor [2]. To eliminate an existentially quantified variable x we (1) perform all resolutions on x and (2) remove all clauses containing x in either phase. In the special case that x occurs only in one phase in the clause set, step (1) is omitted.

DNNF Computation with Projection DNNF (decomposable negation normal form) is a knowledge compilation format for Boolean functions which is considered to be more succinct than BDDs [3]. DNNF representations of a formula φ in negation normal form are gathered by enforcing the decompositional property, i.e. for every sub-formula $\rho = \bigwedge_i \psi_i$ of φ we require $\mathrm{vars}(\psi_i) \cap \mathrm{vars}(\psi_j) = \emptyset$ for all $i \neq j$. After transforming a formula to its DNNF representation, a projection (and therefore an existential BQE) can be computed in polynomial time in $|\varphi|$.

Substitute & Simplify The principle of *substitute & simplify* goes back to the work of Boole and Shannon. In [4] this idea was extended to a quantifier elimination procedure for existential (and universal) quantifiers. We can eliminate a single existential quantifier with $\exists x \varphi \iff \varphi[1/x] \vee \varphi[0/x]$ (*substitue*) and *simplify* the formula afterwards.

3 First Results & Outlook

We implemented the first three approaches on top of state-of-the art tools. For evaluation we chose 1500 instances of `QF_BV`, an SMT benchmark set for bit-vector arithmetic. **Clause distribution** worked only for formulas with a small number of quantified variables, but on the plus side preserves the CNF of the formula. With the **DNNF** approach we could solve 893 formulas within a time limit of one hour per instance. A slightly improved version (with a SAT Solver as preprocessor) could solve 969 formulas. The **model enumeration** approach could solve 1033 instances within the same time limit.

As the next step we want to combine these different approaches. On the one hand we want to build a portfolio-based procedure, i.e. choosing the statistically "best" approach w. r. t. the formula structure and the requirements for the output formula. On the other hand we want to interweave the different approaches to an algorithm which can dynamically switch between the presented procedures, e.g. perform clause distribution on variables where the number of newly introduced clauses is under a certain threshold and then proceed with another approach.

References

[1] Gebser, M., Kaufmann, B., Schaub, T.: Solution enumeration for projected boolean search problems. In: Proceedings of CPAIOR 2009. Springer-Verlag, Berlin, Heidelberg (2009) 71–86

[2] Biere, A.: Resolve and expand. In: Proceedings of SAT 2004. Volume 3542 of LNCS. Springer-Verlag, Berlin, Heidelberg (2005) 59–70

[3] Darwiche, A.: Decomposable negation normal form. Journal of the ACM **48**(4) (2001) 608–647

[4] Seidl, A.M., Sturm, T.: Boolean quantification in a first-order context. In: Proceedings of CASC 2003. Technische Universität München, Garching (2003) 329–345