AIGSolve: An AIG Based QBF Solver

Florian Pigorsch and Christoph Scholl

Albert-Ludwigs-Universität Freiburg, Institut für Informatik,
D-79110 Freiburg im Breisgau, Germany
{pigorsch, scholl}@informatik.uni-freiburg.de

AIGsolve [4] is a QBF solver for instances in prenex normal form. It consists of four phases: (1) the *preprocessing* phase, in which various techniques are used to simplify the input formula, (2) the *structure extraction* phase, where functional definitions of variables are extracted from the QBF formula, (3) the *early quantification* phase, where a non-linear quantifier structure is extracted from the QBF yielding a tree-shaped formula, and (4) the *symbolic quantifier elimination* phase, in which the QBF formula is translated to a symbolic, AIG (*And-Inverter Graph*) [1] based representation, allowing the application of AIG operations to eliminate the quantifiers present in the formula.

1 QBF Preprocessing

In the preprocessing phase we apply several well-known techniques, e.g. unit propagation, subsumption checking, for-all reduction, equivalence reduction, as well as novell techniques, such as the detection of units implied by the matrix of the QBF, in order to simplify the input formula.

2 Structure Extraction

In this phase, the QBF is scanned for sets of clauses establishing functional definitions of variables, e.g. $(\overline{y} \vee x_1), \ldots, (\overline{y} \vee x_n), (y \vee \overline{x_1} \vee \ldots \vee \overline{x_n})$ which defines y to be equivalent to $x_1 \wedge \ldots \wedge x_n$. Instead of substituting the found definitions into the QBF and applying the distributive law to produce a flat CNF representation, we create a non-CNF representation of the QBF which is later directly transformed into an AIG.

3 Early Quantification

Here, the quantifiers from the linear prefix are "pushed" into the non-CNF matrix of the QBF, reducing the scope of the individual quatifiers and producing a tree-shaped non-CNF formula.

4 Symbolic Quantifier Elimination

The solver's core is the symbolic quantifier elimination phase. Here the tree-shaped non-CNF formula from the previous phase is traversed in a depth-first

manner, creating AIG representations for all sub-formulas and eliminating quantifiers using AIG operations. The procedure terminates when all quantifiers are eliminated from the QBF, leading either to a *satisfiable* of *unsatisfiable* result. To avoid the potential blow-up during quantifier elimination, we apply several optimization techniques which help to compress intermediate AIG representations: *Functional reduction* [3] identifies and merges equivalent AIG nodes by a combination of simulation and SAT. BDD-sweeping [5] speculatively creates BDDs for intermediate AIGs. Reasonably small BDDs are transformed back to structurally equivalent AIG representations, replacing the original AIG structures by the new, smaller AIGs. AIG rewriting [2] iteratively replaces AIG subgraphs by smaller, precomputed subgraphs, preserving their functionality and decreasing the overall size of the AIG.

References

1. Andreas Kuehlmann, Malay K. Ganai, and Viresh Paruthi. Circuit-based Boolean Reasoning. In *Proc. of the 38th Design Automation Conf. (DAC 2001)*, pages 232–237, 2001.
2. Alan Mishchenko, Satrajit Chatterjee, and Robert Brayton. DAG-aware AIG rewriting a fresh look at combinational logic synthesis. In *Proc. of the 43rd annual Conf. on Design Automation (DAC 2006)*, pages 532–535. ACM, 2006.
3. Alan Mishchenko, Satrajit Chatterjee, Roland Jiang, and Robert K. Brayton. FRAIGs: A unifying representation for logic synthesis and verification. Technical report, EECS Dept., UC Berkeley, 03 2005.
4. Florian Pigorsch and Christoph Scholl. Exploiting structure in an AIG based QBF solver. In *DATE*, pages 1596–1601. IEEE, 2009.
5. Florian Pigorsch, Christoph Scholl, and Stefan Disch. Advanced Unbounded Model Checking Based on AIGs, BDD Sweeping, And Quantifier Scheduling. In *Proc. of the Conf. on Formal Methods in Computer Aided Design (FMCAD 2006)*, pages 89 – 96. IEEE Computer Society Press, Nov 2006.