

EBMiniSAT: learning new clause classes

Bryan Matsuo

Department of Computer Science, UCSC

March 2, 2011

Abstract

For the SAT Competition 2011 MINISAT Hack Track, an extension of MINISAT 2.2.0, EBMINISAT, that embeds a technique to learn a new and fairly unstudied class of clauses introduced and first implemented by the authors of RSAT in [PD08]. This report summarizes the techniques employed and implementation decisions made in EBMINISAT. The “EB” prefix in the name EBMINISAT comes from the 1-Empowering Bi-asserting (EB) clause learning technique which has been embedded in this solver.

1 Introduction

This paper describes a hacked version of MINISAT that learns 1-empowering bi-asserting clauses as given in [PD08]. When implemented, the new clause class reportedly gave RSAT an enhanced ability to solve UNSAT instances of an industrial nature [PD08]. The solver presented here, EBMINISAT, was developed in conjunction with EBGLOUSE, a solver in the SAT Competition 2011 Main Track, to give more experimental knowledge about the usefulness of this new clause class.

2 A new class of learnt clause

A class of learnt clauses was presented in [PD08] and implemented in the solver RSAT for SAT competition 2009. This technique involves attempting to learn clauses lying in the intersection of the following two clause classes if they have low enough backtrack levels (in relation to the FUIP).

Definition 2.1 (1-Empowerment). *A clause $c = (\alpha \Rightarrow \ell)$ for some literal ℓ and conjunction of literals α is 1-empowering with respect to a clause set Δ if*

1. $\Delta \models c$: Δ logically implies clause c .

2. $\Delta \wedge \alpha$ is 1-consistent: Asserting α does not result in a conflict detectable by unit resolution.
3. $\Delta \wedge \alpha \not\models \ell$: Literal ℓ cannot be derived from $\Delta \wedge \alpha$ via unit-propagation.

The literal ℓ is known as an empowering literal of c .

Definition 2.2 (Bi-asserting Clause). *A conflict clause c in at a given state solver state is bi-asserting if contains exactly two literals on the highest decision level.*

Clauses in this class intersection are non-asserting but are generally shorter than the FUIP. Checking for 1-Empowerment in conflict clauses with respect to the entire clause database is too costly to do. However, [PD08] gives an efficient algorithm for detecting 1-Empowerment with respect to the clauses used in the derivation of the conflict clause (which tends to be accurate enough) incurring minimal overhead.

It is possible to track which literals in a learnt clause are empowering. In practice however, this has not shown to lead to any significant improvement. So the first found 1-empowering bi-asserting clause is always the one learnt (if a bi-asserting clause is learnt).

As with the authors of [PD08], 1-empowering bi-asserting clauses are learned if their decision level is at least two levels below that of the FUIP corresponding to the same conflict.

3 Using bi-asserting clauses

Because learnt bi-asserting clauses are non-asserting, it isn't immediately clear how a solver should proceed after learning them. There are a number of combinations of ways you can affect the next decision, including a forceful assumption. Preliminary results run by the author have suggested that it's most useful to increase the first-propagated bi-asserting variable's VSIDS score by the standard method two times and the

VSIDS score of the later-propagated bi-asserting variable three times. EBMINISAT then simply defers the next decision to the main decision engine from MINISAT .

4 Acknowledgements

Many thanks are due to Allen Van Gelder for helpful implementation discussions and guidance

in benchmarking.

References

- [PD08] Knot Pipatsrisawat and Adnan Darwiche, *A new clause learning scheme for efficient unsatisfiability proofs*, Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI), 2008.