Rsat 3.01: Solver Description

Knot Pipatsrisawat and Adnan Darwiche

Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095 USA
{thammakn,darwiche}@cs.ucla.edu

Abstract. In this document, we briefly describe the techniques employed by the Rsat solver participating in SAT-Race 2008.

1 Introduction

This version of Rsat is based on the one that participated in the SAT Competition 2007 [2].[1] It still utilizes the SatELite preprocessor [1] as in the previous version. The solver is written in C/C++ and is compiled as a 32-bit binary. Two major changes have been incorporated into this version of the solver. We describe each of them next.

2 Bi-asserting clause Learning

This version of Rsat tries to learn 1–empowering bi-asserting clauses [4] whenever possible. A conflict clause is 1–empowering if it allows (empowers) unit resolution to derive a new implication, while a bi-asserting clause is a conflict clause with exactly two literals falsified at the conflict level (as opposed to exactly one in the case of asserting clause). The standard algorithm for deriving conflict clauses can be easily modified to detect any occurrence of a 1–empowering bi-asserting clause. Once a 1–empowering bi-asserting clause is found, Rsat will learn it *instead* of the normal (FUIP) asserting clause if it induces an assertion level that is smaller than the one induced by the asserting clause.[2] Empirically, this new learning scheme tends to improve the performance of our solver on unsatisfiable problems. See [4] for more details.

3 Probing

This version of Rsat also performs probing [3] as a preprocessing step. Prior to the main search routine, the variables are ordered according to their scores from the JW heuristic [5]. Then, variables with high scores are probed for necessary implications. Currently, the solver is configured to probe at most 8000 variables per problem to limit the amount of overhead on large problems. The derived implications are then made permanent prior to the search.

[1] See http://reasoning.cs.ucla.edu/rsat for information on previous versions.

[2] The assertion level of a bi-asserting clause is defined to be the second highest level of any literal in it.

References

1. EÉN, N., AND BIERE, A. Effective preprocessing in sat through variable and clause elimination. In *SAT* (2005), pp. 61–75.
2. LE BERRE, D., SIMON, L., AND ROUSSEL, O. SAT'07 Competition Homepage, http://www.satcompetition.org/2007/.
3. LYNCE, I., AND MARQUES-SILVA, J. Probing-based preprocessing techniques for propositional satisfiability. *Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on* (3-5 Nov. 2003), 105–110.
4. PIPATSRISAWAT, K., AND DARWICHE, A. A new learning scheme for efficient unsatisfiability proofs. In *Proceedings of 23rd National Conference on Artificial Intelligence (AAAI) (to appear)* (2008).
5. WANG, J. A branching heuristic for testing propositional satisfiability. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* (1995), pp. 4236–4238.