SAT4JGTI: SAT4J 2.0 RC6 + SatELite

Daniel Le Berre
http://www.sat4j.org/

April 7, 2008

The SAT4J Project is an open source library of SAT solvers in Java based on the original Minisat specification[3]. The library is the basis of pseudo boolean solvers, maxsat solvers, and a CSP solver. The aim of that library is to provide to Java users an easy access to SAT-based technology. The library has been available under the GNU Lesser General Pulic License (LGPL) until release 2.0. It is now released under the dual Eclipse Public License and LGPL.

SAT4J is currently used in several tools, including Eclipse (http://www.eclipse.org/, Genetic Network Analyzer (http://www.genostar.com/), Alloy (http://alloy.mit.edu/). See SAT4J web site for more applications.

SAT4JGTI uses SatELite[2] as a front end to the default SAT solver that ships with SAT4J 2.0 RC6 since the two qualification phases provided some evidences that using such preprocessing was making a big difference. Here are the details of the default configuration of SAT4J SAT solver:

Heuristics VSIDS heuristics[5] implemented the minisat way, plus RSAT phase saving[6].

Learning Only clauses of size smaller than 10% of the number of variables of the formula are learned. Furthermore, the Expensive Reason Simplification of Minisat 1.13[4] is used to shorten learned clauses.

Restarts Using Armin Biere Inner/Outer restart strategy presented at SAT 2007 in his invited talk[1]. Both the inner and outer loop bounds are increased by 50% when reached. The intial bounds are 100 conflicts.

References

[1] Armin Biere. A short history on sat solver technology and what is next?, May 2007. Invited Talk for the SAT 2007 conference.

[2] Niklas Eén and Armin Biere. Effective preprocessing in sat through variable and clause elimination. In Fahiem Bacchus and Toby Walsh, editors, *SAT*, volume 3569 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2005.

[3] Niklas Eén Niklas Sörensson. An extensible sat-solver. In *Proceedings of the Sixth International Conference on Theory and Applications of Satisfiability Testing, LNCS 2919*, pages 502–518, 2003.

[4] Niklas Eén Niklas Sörensson. Minisat a sat solver with conflict-clause minimization, 2005. SAT 2005 Competition solver description.

[5] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*, pages 530–535, 2001.

[6] Knot Pipatsrisawat and Adnan Darwiche. A lightweight component caching scheme for satisfiability solvers. In João Marques-Silva and Karem A. Sakallah, editors, *SAT*, volume 4501 of *Lecture Notes in Computer Science*, pages 294–299. Springer, 2007.