
BC_MINISAT_ALL: A Blocking AII SAT Solver

Takahisa Toda <todo.takahisa(at)gmail.com>

Last update: 2015-9-30

1. Description

A blocking clause-based AII SAT solver (a blocking solver for short), implemented on top of MiniSat-C v1.14.1, is presented.

2. DOWNLOAD

- Latest version: [bc_minisat_all version 1.1.0](#), released on 30th Sep., 2015.
- Old versions: [bc_minisat_all version 1.0.0](#), [v.2.0.1](#), [v.2.0.0](#), [v.1.0.0](#).

3. FILE FORMAT

Input boolean formula should be in DIMACS CNF format. For details of DIMACS CNF format and benchmark problems, see [SATLIB](#).

4. HOW TO COMPILE

If no option is given, standard mode is selected.

```
$ tar zxvf bc_minisat_all-1.0.0.tar.gz
$ cd bc_minisat_all-1.0.0
$ make [options]
list of options
s  standard: debug information used by debugger is generated
   at compilation time, and detailed solver status is reported at
   runtime.
p  profile: in addition to standard setting, profile information
   used by gprof is generated at compilation time and several tests
   are performed at runtime.
d  debug: in addition to standard setting, several tests are
   performed at runtime and no optimization is applied.
r  release: release version, compiled with dynamic link
rs static: release version, compiled with static link
clean executable files, object files, etc are removed.
```

5. MACRO

Program behavior can be controlled by defining or not defining the following macros in `Makefile`.

- **SIMPLIFY**: Simplification of assignments is enabled. If this is not defined, simplification is not performed, and blocking clauses consisting of decision literals are used instead.
- **NONDISJOINT**: Nondisjoint partial assignments are computed (see Yu et al. 2014). If this is not defined, computed partial assignments are pairwise disjoint: in other words, different partial assignments do not share common total assignments. This macro makes sense only when **SIMPLIFICATION** is defined too.
- **CONTINUE**: Search is continued at the point where a solutions is found. If this is not defined, solver starts from scratch.
- **FIXEDORDER**: Variable selection heuristic is disabled and variables are selected in increasing order of variable indices. If this is not defined, variable selection heuristic is used. This functionality is added to evaluate efficiency of variable selection heuristics.
- **GMP**: GNU MP bignum library is used to count solutions.

6. USAGE

If an output file is specified, all satisfying assignments to a CNF are generated in DIMACS CNF format without problem line. *Notice: there may be as many number of assignments as can not be stored in a disk space.* If simplification is performed, computed assignments are partial, which means some variables may not appear.

```
Usage: ./bc_minisat_all [options] input-file [output-file]
```

7. LICENSE

`bc_minisat_all` is implemented by modifying MiniSat-C_v1.14.1. Please confirm the license file included in this software.

8. NOTICE

A huge number of assignments may be generated and *disk space may be exhausted*. To avoid disk overflow, take measure such as using **ulimit** command.

9. REFERENCES

- N. Eén, N. Sörensson : [MiniSat Page](#): MiniSat-C_v1.14.1, accessed on 15 Dec. 2014.
- N. Eén, N. Sörensson : [An Extensible SAT-solver](#), In Proceedings of the 6th international conference of Theory and Applications of Satisfiability Testing, pages 502--518, 2004.

- Morgado, A. and Marques-Silva, J.: [Good learning and implicit model enumeration](#), In Proceedings of 17th IEEE International Conference on Tools with Artificial Intelligence, 2005. ICTAI 05.
- Yinlei Yu and Subramanyan, P. and Tsiskaridze, N. and Malik, S.: [All-SAT Using Minimal Blocking Clauses](#), In Proceedings of 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems, 2014, pp.86-91.
- McMillan, KenL.: [Applying SAT Methods in Unbounded Symbolic Model Checking](#), Computer Aided Verification, LNCS Vol.2404, pp.250-264. 2002.
- Jin, HoonSang and Han, HyoJung and Somenzi, Fabio: [Efficient Conflict Analysis for Finding All Satisfying Assignments of a Boolean Circuit](#), Tools and Algorithms for the Construction and Analysis of Systems, LNCS Vol.3440, pp.287-300, 2005.
- Weinan Zhao and Weimin Wu: [ASIG: An all-solution SAT solver for CNF formulas](#), In Proceedings of 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics '09, pp. 508-513, 2009.