
CNF2OBDD or BDD_MINISAT_ALL

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

Last update: 2015-9-30

1. Description

A CNF to OBDD compiler or a formula-BDD caching AllSAT solver, implemented on top of MiniSat-C v1.14.1, is presented. This software has two names because of a double meaning. It may be called *cnf2obdd* to put more emphasis on a CNF to OBDD compiler aspect, which supports rich operations for postprocessing, and it also may be called *bdd_minisat_all* on an AllSAT solver aspect, which simply generates all solutions, a familiar format to wider users.

2. DOWNLOAD

- Latest version: [bdd_minisat_all version 1.0.0](#), released on 30th Sep., 2015.

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 bdd_minisat_all-1.0.0.tar.gz
$ cd bdd_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. 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 you want to use timelimit or status report functionality, define `TIMELIMIT` in `Makefile`. If you want to use refresh option, define `REFRESHOBDD` in `Makefile`.

```
Usage: ./bdd_minisat_all [options] input-file [output-file]
-n<int> maximum number of obdd nodes: if exceeded, obdd is
  refreshed
```

Since variable orderings significantly affect solver's performance, we recommend that users in advance execute the software [MINCE](#) to determine a good variable ordering.

6. MACRO

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

- `CUTSETCACHE`: Cutset is selected as formula-BDD caching. If this is not defined, separator is selected.
- `LAZY`: Cache operations are reduced (Recommended).

Select one of the two base solver types: a blocking solver or a non-blocking solver.

- `NONBLOCKING` non-blocking procedure is selected as a base solver. If this is not defined, blocking procedure is selected. For options of non-blocking solvers, see [nbc_minisat_all](#).
- `REFRESH`: refresh option in command line is enabled. If the number of BDD nodes exceeds a specified threshold, all solutions are dumped to a file (if output file is specified in command line), all caches are refreshed, and search is continued.

Other options are as follows.

- `TRIE_RECURSION`: Recursive version of trie is used. If this is not defined, iterative version is used.
- `REDUCTION`: Reduction of compiled OBDD into fully reduced one is performed using CUDD library (Optional). To do this, obtain CUDD library and modify `Makefile`.
- `GMP`: GNU MP bignum library is used to count solutions. To do this, obtain GMP library and modify `Makefile`.

7. LICENSE

`bdd_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.
- Grumberg, Orna and Schuster, Assaf and Yadgar, Avi: [Memory Efficient All-Solutions SAT Solver and Its Application for Reachability Analysis](#), Formal Methods in Computer-Aided Design, LNCS Vol.3312, pp.275-289, 2004.
- Gebser, Martin and Kaufmann, Benjamin and Neumann, Andr{\e} and Schaub, Torsten: [Conflict-driven Answer Set Enumeration](#), in Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning, pp.136--148, 2007.
- 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.
- 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.
- T. Toda, K. Tsuda: [BDD Construction for All Solutions SAT and Efficient Caching Mechanism](#), Track on Constraint Solving and Programming and Knowledge Representation and Reasoning (CSP-KR) part of the 30th Annual ACM Symposium on Applied Computing, 2015.
- J. Huang, A. Darwiche: [Using DPLL for efficient OBDD construction](#), In Proceedings of the 7th international conference on Theory and Applications of Satisfiability Testing, pages 157--172, 2005.
- Y. Crama, and P. Hammer: [Boolean Functions: Theory, Algorithms, and Applications](#). Encyclopedia of Mathematics and its Applications. Cambridge University Press (2011).
- R.E. Bryant: [Graph-Based algorithm for Boolean function manipulation](#), IEEE Trans. Comput., Vol.35, pp.677-691 (1986)
- D.E. Knuth: [The Art of Computer Programming Volume 4a](#), Addison-Wesley Professional, New Jersey, USA (2011)
- F. Somenzi: [CU Decision Diagram Package: Release 2.5.0.](#), accessed on 4 September 2012.