CNFs with Elementary Symmetric Clauses and their SAT Solving

Masahiko Sakai, Yohei Umano and Yoshizane Hino

Graduate School of Information Science, Nagoya University,
furo-cho Chikusa-ku, Nagoya 4648603 Japan

The satisfiability problem of boolean formulas (SAT problem) is NP-hard, which is known to be a time-consuming problem. Nevertheless, practical SAT solvers have been developed recently. Some combinatorial problems are solved by using a solver after converting instances of the problems into instances of SAT problem, which is easier to develop and often faster than direct solving. Thus more efficient SAT solvers are promising since there are still many hard-to-solve problems [4].

Most of modern practical SAT solvers take formulas in conjunctive normal forms (CNF) as their inputs, adopt DPLL algorithm [4, 1] and gain efficiency by introducing adequate data structures and a lot of heuristics, where DPLL algorithm is an enhanced one obtained from DL algorithm [1] by introducing value-inferences of variables, called boolean-constraint propagation (BCP), and an efficient non-chronological backtracking. Since most of execution time of these solvers is used for BCP and backtracking, improvements of BCP contribute the efficiency.

The authors introduced clauses, called ES_k-clauses or simply ES-clauses, defined based on elementary symmetric functions that return true if and only if exactly k-inputs are true. By summarizing some (OR-)clauses in CNF with the equivalent ES-clause, we often obtain a shorter representation. We improved BCP for CNF with ES-clauses [3]. The reasons why this method contributes efficiency of SAT solver are as follows:

1. CNF generated from an instance of meaningful problems often contains OR-clauses summarizable into ES-clauses.
2. One ES_1-clause is equivalent to ${}_nC_2 + 1$ OR-clauses.
3. Computation time of BCP for ES-clauses is similar to BCP for OR-clauses.
4. An algorithm that transform a CNF into a CNF with ES-clauses is lighter than SAT solving.

We also implemented these extension for ES_1-clauses onto relatively simple SAT solver, called nanosat, based on DPLL with two-counter method, clause learning, non-chronological backtracking, VSIDS heuristic for variable selection and forgetting learnt clauses [3].

In this talk, we present an idea that incorporate the extension into two-pointer based practical solver, like minisat [2], and give some experimental results. Since two-pointer implementation for ES_k is not efficient, we employed hybrid one that deals OR-clauses by two-pointer method and ES-clauses by full-pointer method. The experiments show that this hybrid mechanism works nicely.

References

1. M. Davis, G. Logemann, D. Loveland. A machine program for theorem proving. *Communications of the ACM*, vol. 5 (7), pp. 394–397, 1962.
2. N. Eén, N. Sörensson. An Extensible SAT-solver. In *Proceedings of the Sixth International Conference on Theory and Applications of Satisfiability Testing*, LNCS 2919, pp. 502–518, 2004.
3. Y. Umano, M. Sakai, N. Nishida, T. Sakabe, K. Kusakari. Solving Satisfiability of CNF Formulas with Clauses Based on Elementary Symmetric Functions. *IEICE Transaction on Information and Systems*, vol. J93-D (1), pp. 1–9, 2009 (in Japanese).
4. L. Zhang. Searching for Truth: Techniques for Satisfiability of Boolean Formulas. Ph.D. thesis, Princeton University, 2003.