Dinosat: A SAT Solver with Native DNF Support

Thomas Bartel ✉
CAS Software, Karlsruhe, Germany

Tomáš Balyo ✉
CAS Software, Karlsruhe, Germany

Markus Iser ✉
Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract

We introduce a new SAT solver Dinosat, whose input is a conjunction of clauses, at-most-one constraints and disjunctive normal form (DNF) formulas. The native support of DNF formulas is motivated by the application domain of SAT based product configuration. In the paper we show how to integrate DNF reasoning into a CDCL SAT solver. Our preliminary experimental evaluation shows that our approach can outperform a standard SAT solver when all constraints are translated to CNF.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming

Keywords and phrases Sat Solver, DNF, Product Configuration

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Category Tool Paper

1 Introduction

TODO Tomas

2 Preliminaries

A *Boolean variable* has two possible values: *True* and *False*. A *literal* is a Boolean variable (positive literal) or a negation of a Boolean variable (negative literal). A *clause* is a disjunction ($\vee$) of literals and, finally, a CNF formula (or just formula) is a conjuction of clauses. A clause with only one literal is called a *unit clause*. A positive (resp. negative) literal is satisfied if the corresponding variable is assigned the value *True* (resp. *False*) A clause is satisfied, if at least one of its literals is satisfied and the formula is satisfied, if all its clauses are satisfied.

The satisfiability (SAT) problem is to determine whether a given formula has a satisfying assignment, and if so, also find it. Most complete SAT solvers are based on the DPLL algorithm [1] and its extension the CDCL algorithm [2, 3].

... A clause is DNF formula where each term has exactly one literal

3 Related Work

TODO Markus

There are solvers that natively support XOR (Cryptominisat) or Cardinality constraints (Sat4j,...) but none for DNF???

4 Integrating DNF Reasoning into CDCL

TODO copy from Thomas's thesis

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:2
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

5 Efficient Unit Propagation of DNF Formulas

6 Experimental Evaluation

7 Conclusion

References

1 Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.

2 Joao P. Marques-Silva and Karem A. Sakallah. Grasp: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, 1999.

3 Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient sat solver. In *Proceedings of the 38th annual Design Automation Conference*, pages 530–535, 2001.