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

In this paper we report our preliminary results with a new kind of SAT solver called Dinosat. Dinosat's input is a conjunction of clauses, at-most-one constraints and disjunctive normal form (DNF) formulas. The native support for DNF formulas is motivated by the application domain of SAT based product configuration. A DNF formula can also be viewed as a generalization of a clause, i.e., a clause (disjunction of literals) is special case of a DNF formula, where each term (conjunction of literals) has exactly one literal. Similarly, we can generalize the classical resolution rule and use it to resolve two DNF formulas. Based on that, the CDCL algorithm can be modified to work with DNF formulas instead of just clauses. Using randomly generated formulas (with DNFs) we experimentally show, that in certain relevant scenarios, it is more efficient to solve these formulas with Dinosat than translate them to CNF and use a state-of-the-art SAT solver. Another contribution of this paper is identifying the phase transition points for such formulas.

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 [**?**] and its extension the CDCL algorithm [**?**, **?**].

... 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

5 Efficient Unit Propagation of DNF Formulas

TODO copy from Thomas's thesis

6 Experimental Evaluation

7 Conclusion