1

CNF Clause Trimming with Gröbner Bases via Term Ordering

S. Clegg, M. Huff, D. Janibagian

ECE Dept., Univ. of Utah

spencermclegg@gmail.com, ocarinahuff@msn.com, drew.janibagian@gmail.com

Abstract

This project proposes a CNF SAT transformation algorithm employing a partitioning technique followed by efficient monomial ordering, polynomial to CNF conversion, and finally, run through a SAT solver engine. Gröbner bases as a means to analyze the problem structure.

1. Introduction

The Boolean Satisfiability (SAT) has become a key tool in formal circuit design and verification. Using a SAT solver can find solutions that satisfy a set of Boolean equations (SAT) or determine that there are no solutions that exist (UNSAT). Representing existing and optimized solutions in Conjunctive Normal Form (CNF) equivalence can be checked by finding solutions to literal-disjunctions (clauses) through a variable assignment.

In practice heuristical SAT-solvers can solve many problems rather efficiently; however, this is not true for all SAT problems. Many problems are not efficiently solved by SAT. Since SAT beginnings [1] techniques such as pruning [2] and clause learning [3] have increased their efficiency and success.

To improve SAT efficiency studies have looked at preprocessing the CNF-formulae in an attempt to decrease the problem size. Some experiments contrarily increased problem size while others demonstrated improved SAT-search efficiency in regard to solve time, conflicts, number of decisions, etc. [4]

1. Theory
2. *Gröbner bases*

A Gröbner basis is a set of an ideal in a polynomial ring over a field [5]. Gröbner basis is a standard approach to solving systems of polynomial equations. Applying Gröbner bases to a SAT instance can improve problem solving performance by deriving reduced bases of sets of polynomials.

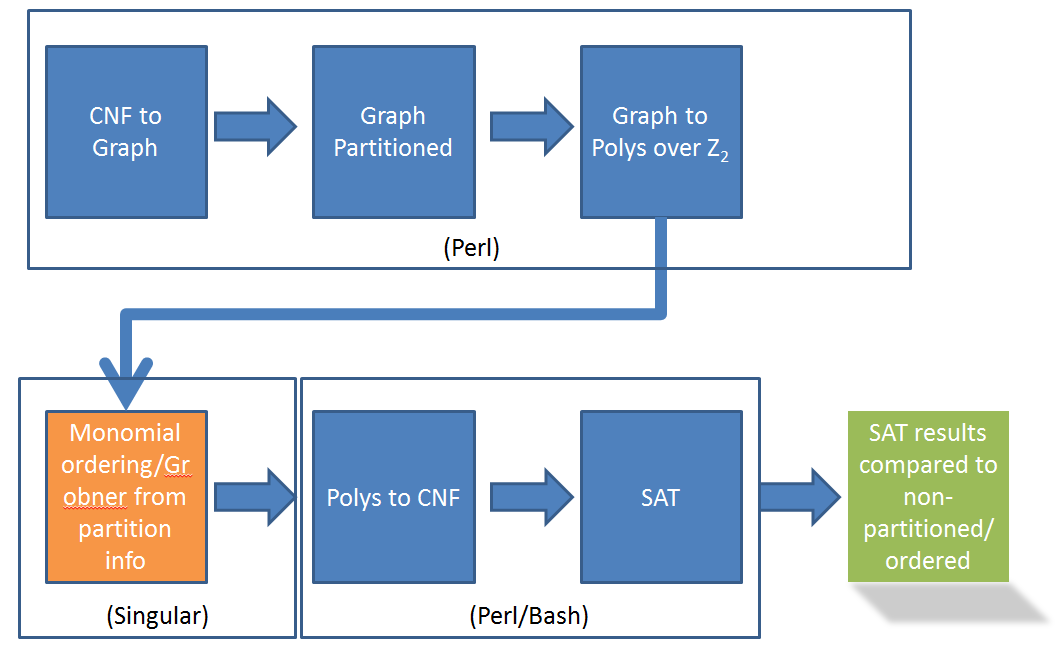
*G* = *GröbnerBasis*(*I*) *⇐⇒ ∀f ∈ I* : *f \_*= 0*, ∃gi ∈ G* : *lp*(*gi*)*|lp*(*f*) (1)

1. *Partitioning*

Efficiency of Gröbner basis computations can be increased by manipulating or partitioning the ideal set and remerging in the original set.

1. *Monomial Ordering*
2. Algorithm Implintation (CNF to Graph, Partitioning, Graph to Poly, Ordering, Ply to CNF, Sat)

Talk about details of what we did, post code and paragraph on each tool here.



1. Results

Show table with results. Comment on the results

1. Conclusion

Wrap up the summary here.

1. Bibliography

[1] M. Davis and H. Putnam. A computing procedure for quantification theory.

CACM, 7:201–215, 1960.

[2] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem

proving. CACM, 5:394–397, 1962.

[3] P. Beame, H. Kautz, and A. Sabharwal. Understanding and harnessing the

potential of clause learning. JAIR, 22:319–351, Dec. 2004.

[4] A Groebner Basis Approach to CNF formulae Preprocessing. Chris Condrat and Priyank Kalla. Intl. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), O. Grumberg and M. Huth (Eds.) Lecture Notes in Computer Science (LNCS) vol. 4424, pp. 618-631, March 2007.

[5] Gröbner basis. Encyclopedia of Mathematics. URL: http://www.encyclopediaofmath.org/index.php?title=Gr%C3%B6bner\_basis&oldid=32386