Efficiency of Propositional Proof Systems

Olga Tveretina

Compiler Technology and Computer Architecture Group

Introduction

The classical propositional calculus ... presents some of the most challenging and intriguing problems in modern logic. A. Urquhart

1. The problem of the propositional proof complexity investigated from late 1960s
2. Progress made in determining relative complexity, and in proving strong lower bounds for some proof systems
3. Major problems remain

Propositional logic

1. *Propositional formula:* propositional variables connected with logical connectives
2. *Conjunctive Normal Form (CNF):* conjunction of disjunctions of propositional variables and their negations
3. A propositional formula is a *tautology* if it is TRUE for all possible combinations of values of variables (TAUT)
4. A propositional formula is *unsatisfiable* if it is FALSE for all possible combinations of values of variables

SAT problem

1. *Satisfiability (SAT)*: to determine if there is an assignment of the variables of a given propositional formula φ that evaluates φ to TRUE
2. Equally important to determine that no such assignment exists (φ is FALSE for all assignments of the variables)
3. One of the first provably *NP-complete* problems
4. *Applications*: formal equivalence checking, model checking, formal verification of pipelined microprocessors, planning and scheduling problems etc

Proof system

Cook and Reckhow [1979]: formal notion of proof system

Notion of proof: a string of characters in a certain format

A propositional proof system (PPS) is a polynomial-time computable predicate S such that

$$\varphi \in TAUT \iff \exists p : S(\varphi, p)$$

This property ensures that the PPS S is logically sound and complete

Polynomial simulation

The complexity of a PPS S is the smallest function in terms of asymptotic behaviour at large n

$$B : \mathbb{N} \to \mathbb{N}$$

such that

$$\varphi \in TAUT \;\Leftrightarrow\; \exists p : |p| \leqslant B(\varphi) \wedge S(\varphi, p)$$

A PPS R p-simulates a PPS S if there is a polynomial-time computable function f mapping proofs in S onto proofs in R

$$\forall \varphi \in TAUT : \quad S(\varphi, p) \;\Leftrightarrow\; R(\varphi, f(p))$$

It implies

$$Comp(R) \leqslant Comp(S)^{O(1)}$$

Benchmark: pigeonhole principle as a proposition

The pigeonhole principle: if n items are put into $n-1$ containers, then at least one container must contain more than one item

$$\mathrm{PHP}_n = \bigwedge_{i=1}^{n+1} [\bigvee_{j=1}^{n} p_{i,j}] \wedge \bigwedge_{1 \leqslant i < j \leqslant n+1, 1 \leqslant k \leqslant n} [\neg p_{i,k} \vee \neg p_{j,k}]$$

A standard benchmark to test the efficiency of propositional proof systems

NP versus co-NP

P versus NP is an important question in theoretical CS

The P versus NP problem asks whether every problem whose solution can be quickly verified by a computer can also be quickly solved by a computer

NP versus coNP is the second most important question

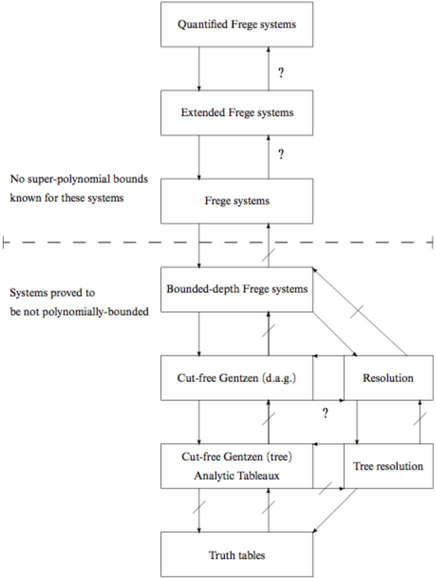
Subset sum problem (NP-complete): given a finite set of integers, is there a non-empty subset that sums to zero?

The complementary problem in co-NP: given a finite set of integers, does every non-empty subset have a non-zero sum?

If NP $\neq$ coNP then P$\neq$ NP

(P is closed under compliment. If NP is not closed under compliment then P and NP cannot be the same sets)

A map of proof systems



A. Urquhart: The Complexity of Propositional Proofs [1995]

An arrow: a proof system in the first box can p-simulate a system in the second box

An arrow with a slash: no p-simulation is possible

The dotted line represents the current frontier of research

Open problems concerning the relative complexity of systems are above the line

Why resolution and OBDDs?

1. SAT solvers and OBDDs are commercially exploited
    1. Hardware verification
    2. Product configuration
2. Yes/NO answers from solvers are not enough
    1. Counterexample of proof needed
    2. Used in proof checking, diagnosis, etc

Today: resolution, extended resolution, OBDD, extended OBDD

Resolution

Introduced by Robinson (1965)
Input: a CNF

Resolution rule:

$$\frac{x \vee C, \neg x \vee D}{C \vee D}$$

Resolution is complete and sound for propositional logic

A refutational proof system: derives the empty clause $\perp$ for an unsatisfiable CNF

Example: $\varphi = (x \vee y) \wedge (\neg x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee \neg y)$
Refutation: $y, \neg y, \perp$

Extended Resolution

Stephen Cook: A short proof of the pigeon hole principle using extended resolution, 1976

$$\underbrace{\text{Resolution rule \& Extension rule}}_{\text{Extended Resolution}}$$

Extension rule:

$$\varphi \wedge \bigwedge_{i \leqslant n} (x_i \leftrightarrow \psi_i)$$

$x_1, \ldots, x_n$ are new variables and $\psi_1, \ldots, \psi_n$ are arbitrary formulas

Theorem (Krajicek and Pudlak, 1989)

Extended resolution p-simulates Frege systems and extended Frege systems.

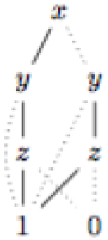
An Ordered Binary Decision Diagram (OBDD)

a canonical data structure to represent propositional formulas

a fixed variable ordering (for canonical representation)

shared sub-graphs (for compression)

OBDDs are extensively used in formal verification and CAD software for circuit synthesis



OBDDs as a proof system

Atserias, Kolaitis & Vardi [2004]: Constraint Propagation as a Proof System

OBDD based proof systems are a special case of CSP

Definition

An OBDD refutation of an unsatisfiable CNF φ is a sequence of OBDDs $B_1, \ldots, B_n$ such that

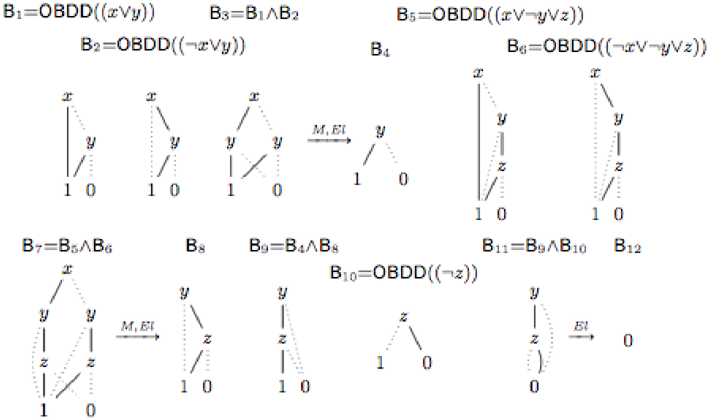
1. (Axiom rule) B_i is an OBDD corresponding to one of the clauses
2. (Join) $B_k = B_i \wedge B_j$

'Extended' OBDDs: (NB! Not standard terminology)

Axiom & $B_k \geqslant B_i \wedge B_j$ ($f \geqslant g$ denotes that f majorizes g)

An exponential lower bound (Krajicek [2008])

OBDD refutation example



Resolution versus OBDDs (arbitrary formulas)

Jan Friso Groote and Hans Zantema: Resolution and binary decision diagrams cannot simulate each other polynomially [2002]

1. Not comparable behaviour:
    1. Many examples where resolution-based techniques out-perform OBBDs with a major factor
    2. For some benchmarks OBDDs have a significant increase of the scale of systems
    3. The benchmark studies say very little about the real relation of resolution and OBDDs
2. A formal comparison of these methods is not straightforward:
    1. OBDDs work on arbitrary formulas
    2. Resolution can take as an input only CNFs

Resolution versus OBDDs (arbitrary formulas)

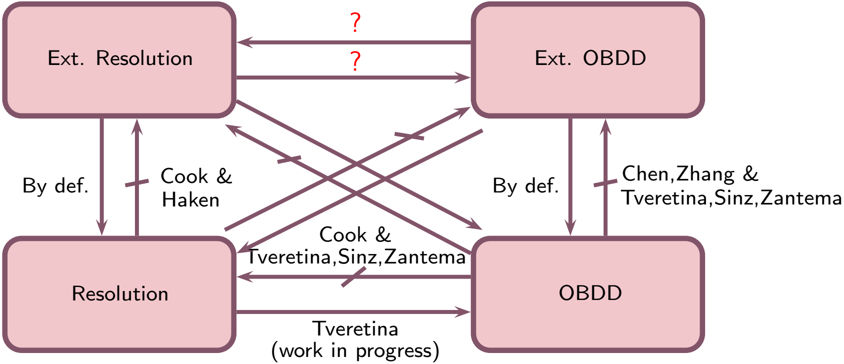
Groote & Zantema:

Are there CNFs having polynomial OBDD proofs and requiring exponentially long resolution proofs?

1. Biconditional formulas have short OBDD proofs
2. After transforming them into CNFs they require exponentially long resolution proofs
3. OBDD proofs of the transformed formulas need exponential size OBDD proofs too

Resolution versus OBDDs (CNFs)

Ext resolution $\to$ OBDD (Peltier [2008])
OBDD $\not\to$ Ext resolution (Cook [1976] & Tveretina, Sinz, Zantema [2008])
Ext OBDD $\to$ resolution (Atserias, Kolatis, Vardi [2004])
Resolution $\not\to$ ext OBDD (Haken [1984] & Chen, Zhang [2009])



Experiments

With Carsten Sinz, Karlsruhe Institute of Technology

MiniSAT 2.0 (conflict driven clause learning), ZRes (original Davis Putnam procedure), and the OBDD package *buddy* 2.4 (usual Boolean operations on OBDDs) on PHP_n

Run-time in seconds, on a logarithmic scale