Has Resolution-Based Clause Learning Hit the Wall?

Allen Van Gelder
Computer Science Department
University of California
Santa Cruz, CA, USA

avg@cs.ucsc.edu

Preface — Quick Tutorial on Global Warming

Why are Carbon Dioxide and Methane called "Greenhouse Gases"?

- They are transparent to 1 μ radiation, which is most of what the sun sends.
- They are block a good amount of 10 μ radiation, which is what the earth mostly reradiates.
- Water vapor is transparent to both 1 μ and 10 μ. Water droplets block both.

Why are melting polar ice-caps very dangerous?

- Ice reradiates a lot of energy from the sun at the same 1 μ that is *not* blocked by Greenhouse Gases.
- Less ice $\rightarrow$ less 1 μ reradiation $\rightarrow$ more warming $\rightarrow$ less ice.

How will global warming destroy civilization?

- More extreme weather will break down the infrastructure of civilization.
- Recent tornadoes, droughts, floods, snows are all indications.
- If Gulf Stream breaks down, much of Russia may become uninhabitable.

What does this have to do with this workshop?

- The future of Satisfiability is in software verification.
- Reliable computer systems will be needed to manage Global Warming.

Overview — This talk is about propositional logic only

- Theoretical Techniques for Proof-System Comparisons
  - P-Simulation
  - Exponential Separation
  - Some Known and Unknown Relationships
- New Technique — *Effective* P-Simulation
- Implications for Proof *Search* — Automatizability
- Experiments with Explicit Resolution Refutations
- Beyond Resolution
  - Boolean Polynomial Calculus
  - SAT Modulo Theories

Theoretical Techniques for Proof-System Comparisons

Two questions about a proof system (Most results are about question (1).)

1. Proof Complexity: How long is the shortest refutation for a formula?
2. Search Complexity: How hard is it to find a (reasonably) short refutation?

Motivation for refinements of general resolution: Try to gain in (2), possibly giving up something in (1), by placing restrictions on the search choices.

Unit Resolution (*incomplete*)
Input Resolution (*incomplete*)
Model Elimination, a.k.a.[1] SL-Resolution, a.k.a. Connection Tableaux
Variable-Elimination Resolution, a.k.a. Davis-Putnam Resolution

Hyper-Resolution
Linear Resolution
Set of Support, a.k.a. Semantic Resolution

[1] a.k.a.: "also known as"

Proof Complexity Comparisons

Let G (good) and B (bad) be two proof systems.

- *G P-simulates B* iff there exists a polynomial $P(L)$ such that, for all formulas F:

  $$|\text{shortest } B\text{-refutation of } F| = L \text{ implies } |\text{shortest } G\text{-refutation of } F| \leq P(L).$$

- B is *exponentially separated from G* iff there exists a positive constant c and a family of formulas F_i such that, for large enough i:

  $$|\text{shortest } G\text{-refutation of } F_i| = L_i \text{ implies } |\text{shortest } B\text{-refutation of } F_i| \geq e^{L_i^c}$$

- We will say G *dominates* B ($G \rightarrow B$ in pictures) if both are true.

Some Known and Unknown Relationships:

Proof System (*SAT Solver*)		Tree-like (DPLL)	$\leftarrow$	Regular (DP res)	$\leftarrow$	Pool (“CL”)	??	General (??)

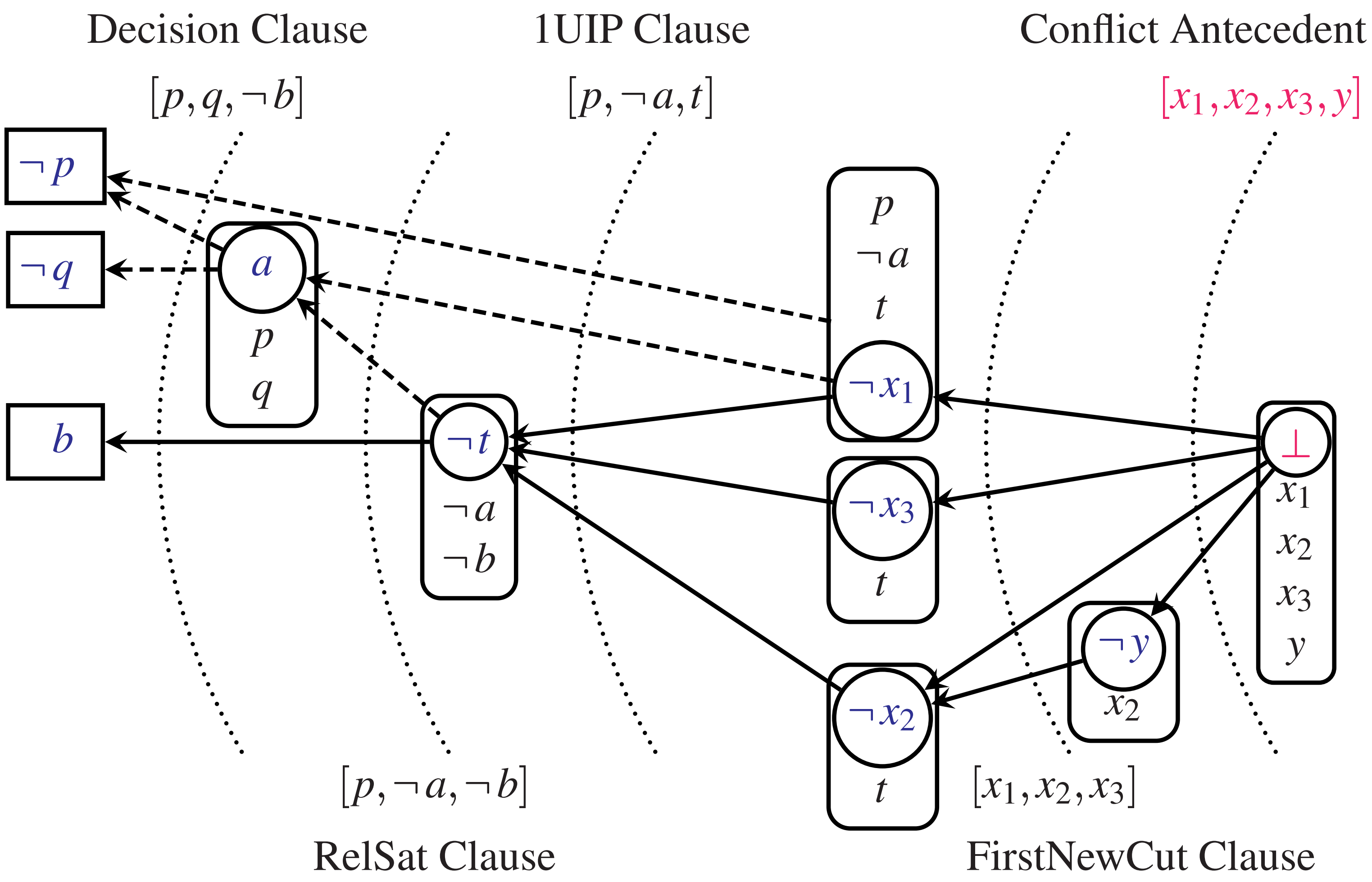
Clause Learning Based on Conflict Graph

Basic Data Structure in Most Modern SAT Solvers

- Introduced in current form in GRASP by Marques-Silva and Sakallah [96,99]
- Adopted into Chaff by Moskewicz, Madigan, Zhao, L. Zhang, and Malik [01]
- Variations in zChaff by L. Zhang, Madigan, Moskewicz, and Malik [01]
- Theoretical analysis by Beame, Kautz, and Sabharwal [*JAIR* 2004]
  - Introduces "first new cut" clause-learning strategy;
  - Raises possibility that (nondeterministic) clause learning might be able to P-simulate general resolution.

Conflict graphs used by `Chaff`, `zChaff`, `BerkMin`, `Jerusat`, `minisat`, `picosat`, ...

Various Cuts in Conflict Graph Yield Different Conflict Clauses to Learn



$\perp$ denotes false.

Dashed lines go to vertices at lower (earlier) "decision levels".

Effective P-simulation: A New Proof-Complexity Comparison

Joint work with Fahiem Bacchus, Philipp Hertel, Toniann Pitassi.

Weaker than P-simulation

Let G (good) and B (bad) be two proof systems.

- *G effectively P-simulates B* iff there exists a polynomial $P(L)$ and a *CNF-encoding transformation $R(F,m)$* such that, for all formulas F:

  $|\text{shortest } B\text{-refut'n of } F| = L \leq m$ implies $|\text{shortest } G\text{-refut'n of } R(F,m)| \leq P(L)$.
- $R(F,m)$ must be computable in time that is polynomial in $|F| + m$.

Theorem:
"CL", the proof system based on a nondeterministic clause learning algorithm, effectively P-simulates general resolution.

- The encoding transformation $R(F,m)$ runs in time $O(n \cdot (|F| + m))$.
- $R(F,m)$ has no knowledge of any refutation of F (but "hopes" there is one of length m or less).

So what? …

Implications for Proof *Search* — Automatizability of "CL"

Informally, a proof system G is automatizable iff there is a deterministic procedure to search for a G-refutation in a "reasonably short" amount of time.

- "Reasonably short" means at most a polynomial $P(L)$, where L is the length of the shortest G-refutation.
- This is a very generous bound, because L might be exponential in $|F|$.

Theorem:
If "CL" is automatizable, then general resolution is automatizable.

Proof idea: For m $= |F|, 2|F|, 4|F|, \ldots$, generate $R(F,m)$, the effective P-simulation encoding, and run the hypothetical algorithm that automatizes "CL" until it succeeds or (if m is too small) it exceeds its time bound.

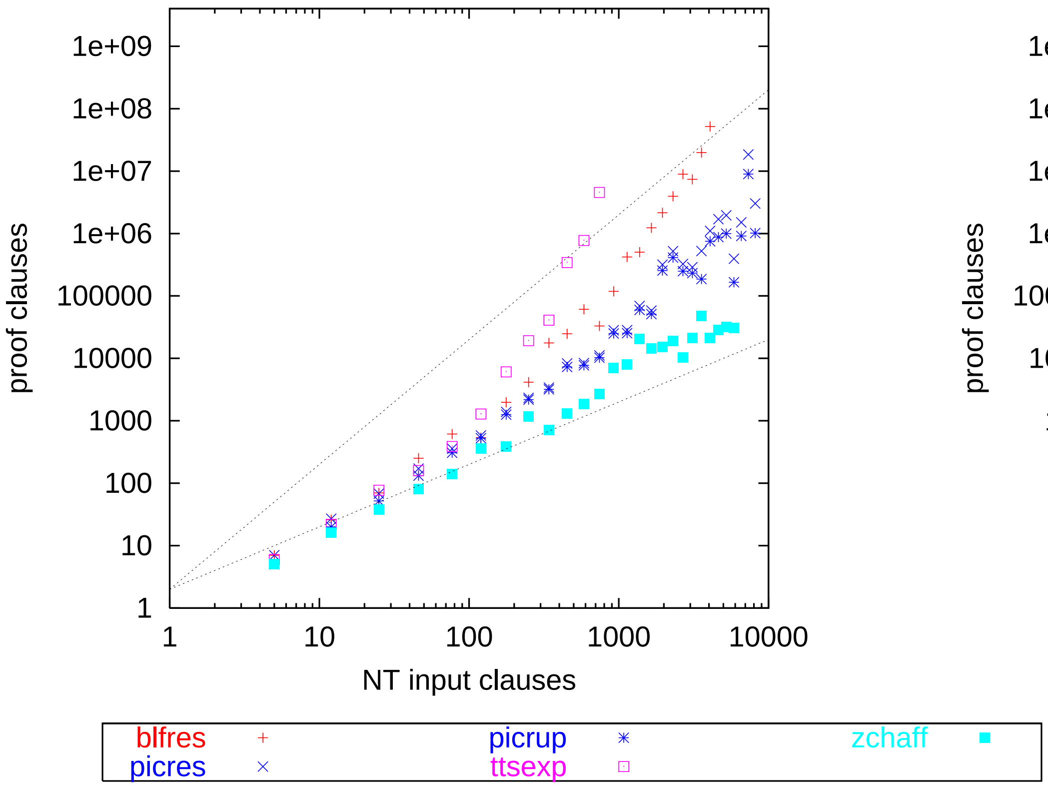
Conclusion:
In this very mild sense, an efficient way to search for refutations in the "CL" framework would provide an efficient way to search for general resolution refutations. However, there is indirect evidence that no efficient way exists to search for general resolution refutations.

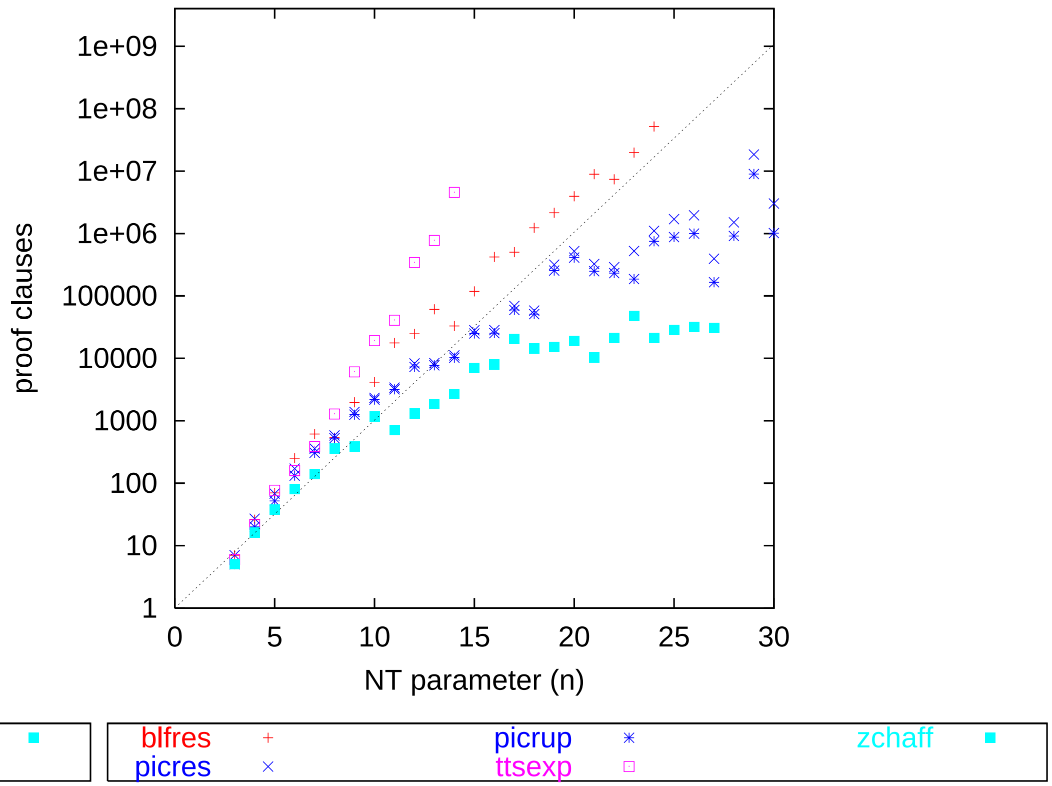
In other words, "clause learning" may have hit the wall in terms of exploiting the capabilities of resolution.

Experiments with Explicit Resolution Refutations — NT Family

The family NT(n) has formulas of length about $n^3 / 3$ clauses and refutations with about $2\,n^3 / 3$ clauses.

This family is a “cleaned up” version of GT(n).

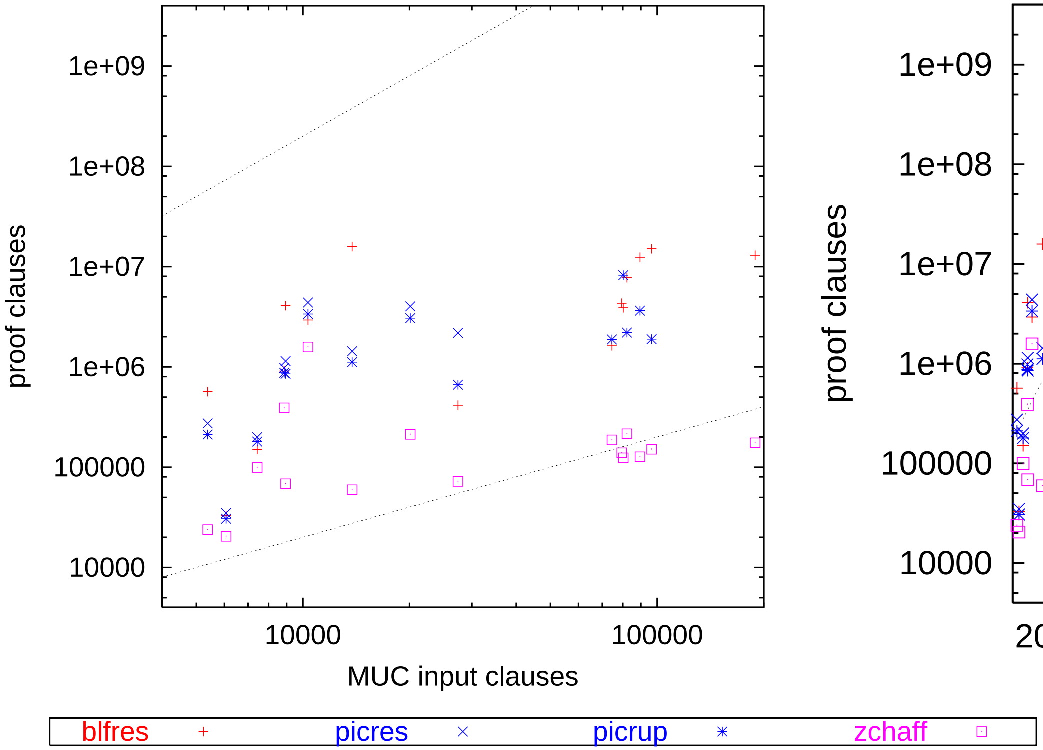


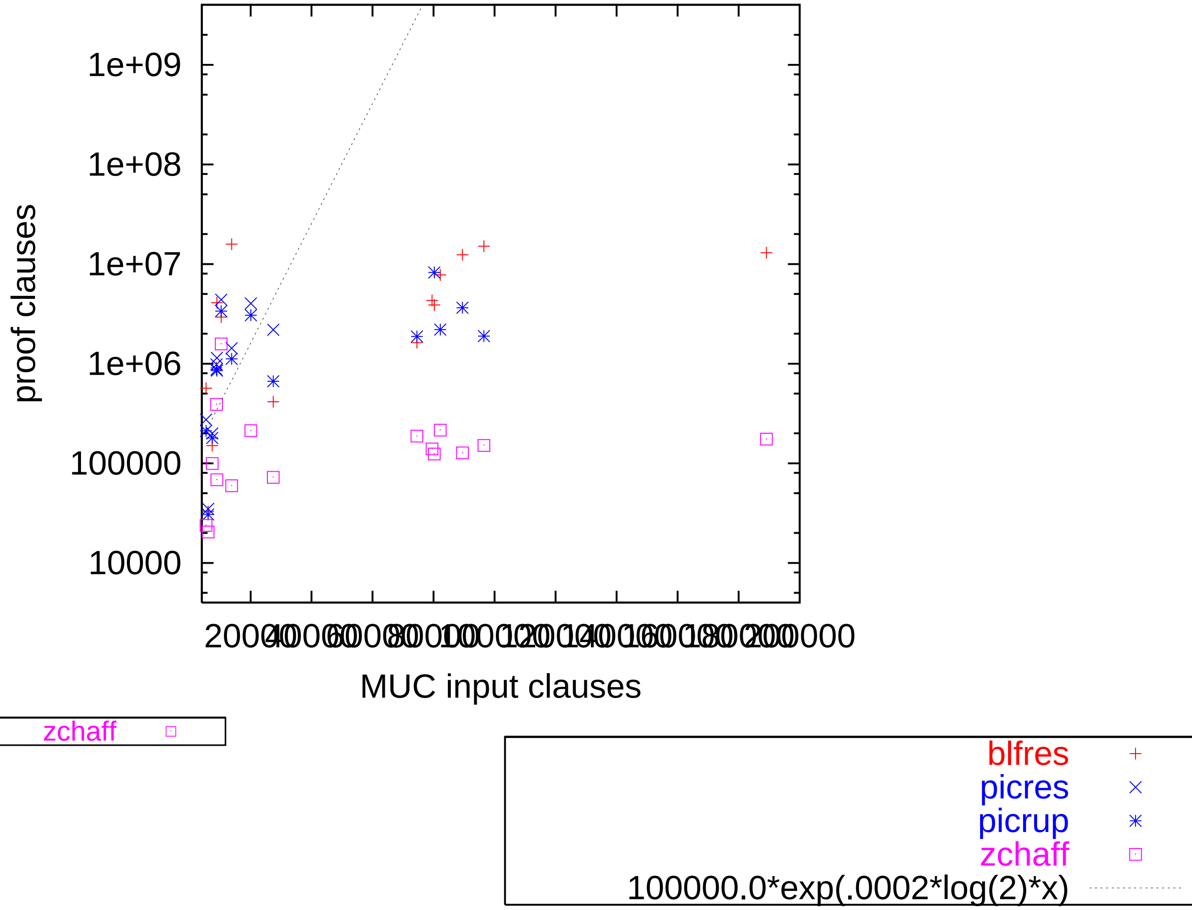


Experiments with Explicit Resolution Refutations — Industrial Benchmarks

`pipe_k 4pipe 4pipe_1_ooo 4pipe_2_ooo 4pipe_3_ooo 4pipe_4_ooo 4pipe_k 5pipe_k`

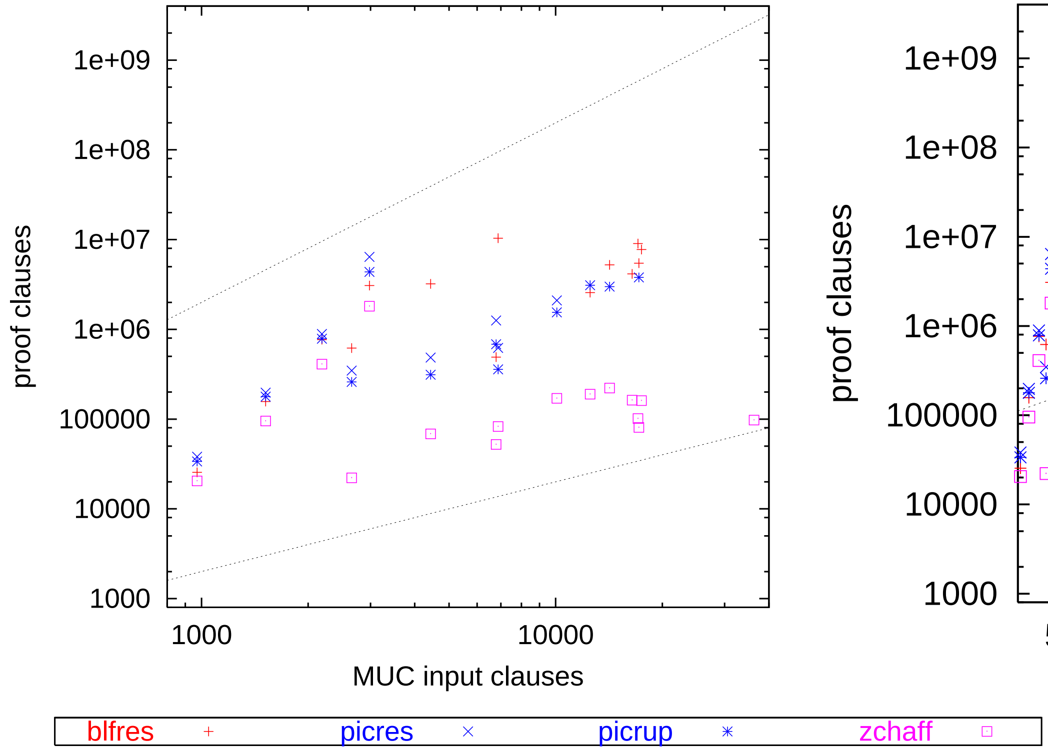
`barrel5 barrel6 barrel7 barrel8 longmult4 longmult5 longmult6 longmult7`

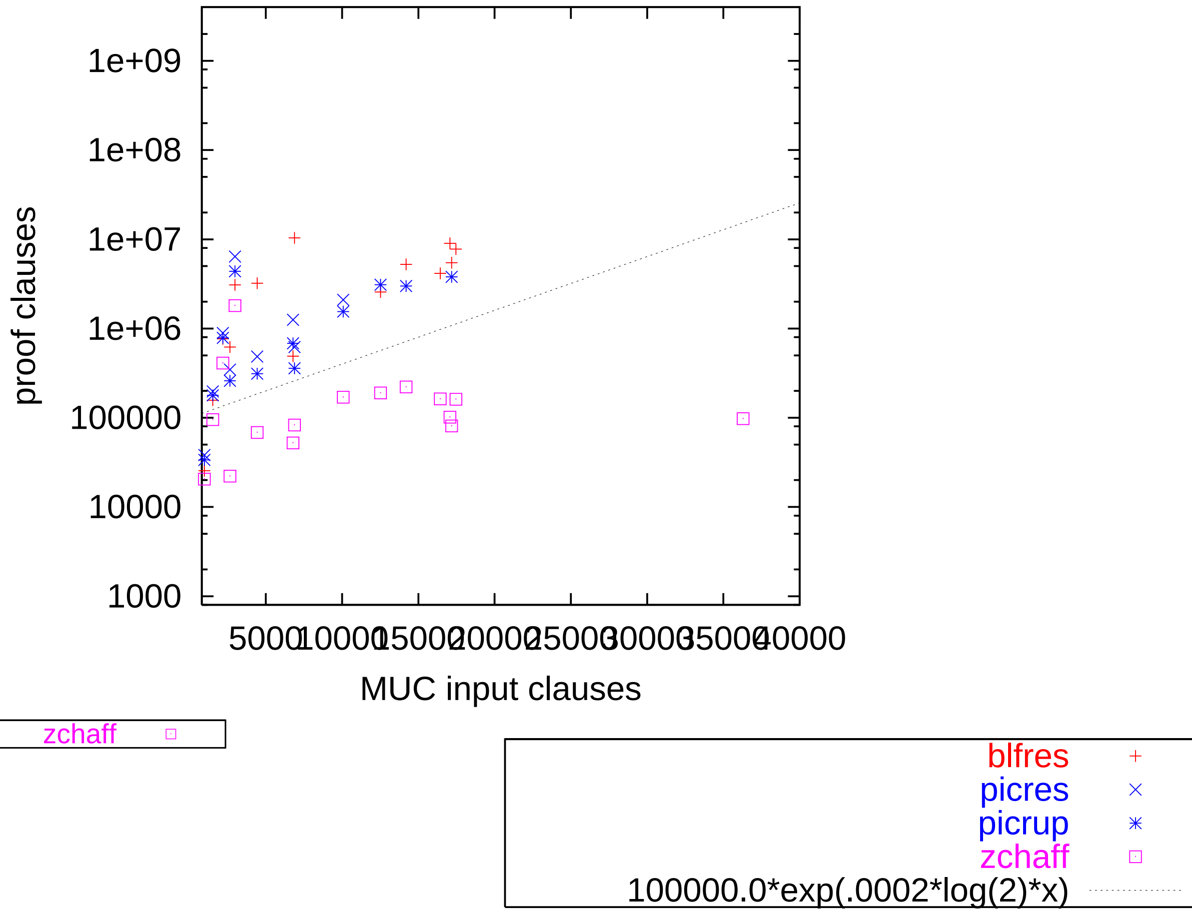




Experiments with Explicit Resolution Refutations — Minimal Unsat Cores

Thanks to Alexander Nadel for regenerating these MUCs as reported in Dershowitz, Hanna, and Nadel, “A Scalable Algorithm for Minimal Unsatisfiable Core Extraction,” SAT 2006.





Beyond Resolution

Future major gains in SAT solving may require getting out of the “resolution box”.

Boolean Polynomial Calculus

- Recall talk by Alyson Reeves.
- Boolean Polynomial Calculus can directly simulate resolution.
- Can combine clauses where resolution is useless:
  - $(x \vee y \vee z)$ and $(x \vee \overline{y} \vee \overline{z})$ resolve to tautologous clause.
  - $(x+1)(y+1)(z+1)$ and $(x+1) \cdot y \cdot z \;\vdash\; (x+1)(y+z)$.
- Exponentially shorter proofs than resolution on some families (?)
- Most theoretical analysis based on degree of polynomial as metric.
  - Analogous to clause-width metric for resolution.
  - But ... Clause-width metric for resolution is doubtful (Bonet-Galesi 2001).
- Good heuristics will be needed to achieve practical success, in any case.

Beyond Resolution — SAT Modulo Theories (SMT)

Integrate decidable theories with propositional logic.

- Natural for software verification and other design checking.
  - Linear real arithmetic LRA
  - Linear integer arithmetic LIA
  - Difference logic DL
  - Lists, trees, arrays, others
- WARNING: Many decision procedures use inference rules equivalent to resolution steps: at most a constant-factor benefit.

Final question (of this talk):
Can Boolean Polynomial Calculus be brought into the SMT fold?