Lower Bounds on Resolution Theorem Proving Via Games (An Exposition)

William Gasarch-U of MD

Credit Where Credit is Due

1. Stays Jukna's book on Circuit complexity had the material.
2. Original source: Beyersdorff, Galesi, Lauria's paper A Lower Bound for the PHP in Tree-Like Resolution by Asymmetric Prover-Delayer Games. In IPL, 2010.
3. Result itself is old; however this proof is new and wonderful.

Connection of NP=coNP

Problem: Given a CNF-Formula $\varphi \notin SAT$ we want a proof that $\varphi \notin SAT$.

1. If we can always get short proof then NP=coNP.
2. Research Program: Show that in various Logic Systems cannot get a short proof.

RESOLUTION RULE

$$\frac{A \vee x \qquad\qquad B \vee \neg x}{A \vee B}$$

Resolution

Definition

Let $\varphi = C_1 \wedge \cdots \wedge C_L$ be a CNF formula. A *Resolution Proof that* $\varphi \notin SAT$, is a sequence of clauses such that on each line you have either

1. One of the C's in φ (called an AXIOM).
2. $A \vee B$ where on prior lines you had $A \vee x$ and $B \vee \neg x$. Variable that is *resolved on* is x.
3. The last line has the empty clause.

EASY: If there is a Resolution Proof that $\varphi \notin SAT$ then $\varphi \notin SAT$.

Example

$\varphi = x_1 \wedge x_2 \wedge (\neg x_1 \vee \neg x_2)$

1. x_1 (AXIOM)
2. $\neg x_1 \vee \neg x_2$ (AXIOM)
3. $\neg x_2$ (From lines 1,2, resolve on x_1.)
4. x_2 (AXIOM)
5. $\emptyset$ (From lines 3,4, resolve on x_2.)

Another Example

The AND of the following:

1. $x_{11} \vee x_{12}$
2. $x_{21} \vee x_{22}$
3. $x_{31} \vee x_{32}$
4. $\neg x_{11} \vee \neg x_{21}$
5. $\neg x_{11} \vee \neg x_{31}$
6. $\neg x_{21} \vee \neg x_{31}$
7. $\neg x_{12} \vee \neg x_{22}$
8. $\neg x_{12} \vee \neg x_{32}$
9. $\neg x_{22} \vee \neg x_{32}$

Another Example

The AND of the following:

1. $x_{11} \vee x_{12}$
2. $x_{21} \vee x_{22}$
3. $x_{31} \vee x_{32}$
4. $\neg x_{11} \vee \neg x_{21}$
5. $\neg x_{11} \vee \neg x_{31}$
6. $\neg x_{21} \vee \neg x_{31}$
7. $\neg x_{12} \vee \neg x_{22}$
8. $\neg x_{12} \vee \neg x_{32}$
9. $\neg x_{22} \vee \neg x_{32}$

This is Pigeonhole Principle: x_{ij} is putting ith pigeon in j hole!

Another Example

The AND of the following:

1. $x_{11} \vee x_{12}$
2. $x_{21} \vee x_{22}$
3. $x_{31} \vee x_{32}$
4. $\neg x_{11} \vee \neg x_{21}$
5. $\neg x_{11} \vee \neg x_{31}$
6. $\neg x_{21} \vee \neg x_{31}$
7. $\neg x_{12} \vee \neg x_{22}$
8. $\neg x_{12} \vee \neg x_{32}$
9. $\neg x_{22} \vee \neg x_{32}$

This is Pigeonhole Principle: x_{ij} is putting ith pigeon in j hole!
Can't put 3 pigeons into 2 holes!

Let $n < m$. n is NUMBER OF HOLES, m is NUMBER OF PIGEONS. x_{ij} will be thought of as Pigeon i IS in Hole j.

Definition

PHP_n^m is the AND of the following:

1. For $1 \le i \le m$

$$x_{i1} \vee x_{i2} \vee \cdots \vee x_{in}$$

   (Pigeon i is in SOME Hole.)

2. For $1 \le i_1 < i_2 \le n$ and $1 \le j \le m$

$$\neg x_{i_1 j} \vee \neg x_{i_2 j}$$

   (Hole j does not have BOTH Pigeon i_1 and Pigeon i_2.)

NOTE: PHP_n^m has nm VARS and $O(mn^2)$ CLAUSES.

PHP- HOW TO VIEW ASSIGNMENTS

An Assignment is an $m \times n$ array of 0's and 1's.

Example: $m = 4, n = 3$.

0	1	0
0	0	1
1	0	0
0	1	0

$x_{12} = x_{23} = x_{13} = x_{42} = 1$. All else 0. Violates PHP since have $x_{12} = x_{42} = 1$.

TWO WAYS TO VIOLATE PHP

1) Have two 1's in a column.

0	1	0
0	0	1
1	0	0
0	1	0

2) Have an all 0's row.

0	1	0
0	0	1
0	0	0
1	0	0

CAN ALWAYS DO A RESOLUTION

$$\varphi(x_1, \ldots, x_v) = C_1 \wedge \cdots \wedge C_L$$

If $\varphi \notin SAT$ then construct Resolution Proof as follows:

1. Form a DECISION TREE with nodes on level i labeled x_i.
2. Every leaf is a complete assignment. Output least indexed clause C that is 0.
3. Turn Decision Tree UPSIDE DOWN, its a Res. Proof.
4. NOTE: Can always do $2^{O(v)}$ size proof.
5. NOTE: The Resolution Proofs are TREE-Resolution.

TREE RESOLUTION

1. Informally- a Tree Resolution proof is one where if written out looks like a tree.
2. Formally- a Tree Resolution proof is one where any clause in the proof is used at most once.

OUR GOAL

Assume $n < m$.

1. PHP_n^m always has a size $2^{O(nm)}$ Tree Resolution Proof.
2. We show $2^{\Omega(n \log n)}$ size is REQUIRED. THIS IS POINT OF THE TALK!!!!!
3. The lower bound is IND of m.
4. There is a matching upper bound of $2^{O(n \log n)}$: Resolution and the weak pigeonhole principle, By Buss and Pitassi. Proceedings of the 1997 Computer Science Logic Conference.

THE PROVER-DEL GAME

Parameters of the game: $a, b \in \mathrm{R}^+$, $p \in \mathrm{N}$, with $\frac{1}{a} + \frac{1}{b} = 1$.

$$\varphi = C_1 \wedge \cdots \wedge C_L \notin SAT.$$

Do the following until a clause is proven false:

1. PROVER picks a variable x that was not already picked.
2. DEL either
   2.1 Sets x to 0 or 1, OR
   2.2 Defers to PROVER .
      2.2.1 If PROVER sets $x = 0$ then DEL gets $\lg a$ points.
      2.2.2 If PROVER sets $x = 1$ then DEL gets $\lg b$ points.

At end if DEL has p points then he WINS; otherwise PROVER WINS.

CONVENTION

We assume that PROVER and DEL play perfectly.

1. *PROVER wins* means *PROVER has a winning strategy*.
2. *DEL wins* means *DEL has a winning strategy*.

PROVER-DEL GAME and TREE RES!

Lemma

Let $a, b \in \mathrm{R}^+$, such that $\frac{1}{a} + \frac{1}{b} = 1$, $p \in \mathrm{N}$, $\varphi \notin SAT$. If φ has a Tree Res proof of size $< 2^p$ then PROVER wins.

Proof.

We do case $a = b = 2$. PROVER Strategy:

1. Initially T is res tree of size $< 2^p$ and DEL has 0 points.
2. PROVER picks x, the LAST var resolved on.
3. If DEL sets x DEL gets no points.
4. If DEL defers then PROVER sets to 1 or 0- whichever yields a smaller tree.NOTE: One of the trees will be of size $< 2^{p-1}$. DEL gets 1 point.
5. Repeat: after ith stage will always have T of size $< 2^{p-i}$, and DEL has $\leq i$ points.

CONTRAPOSITIVE IS AWESOME!

Recall:

Lemma

Let $a, b \in \mathrm{R}^+$, with $\frac{1}{a} + \frac{1}{b} = 1$, $p \in \mathrm{N}$, $\varphi \notin SAT$. If φ has a Tree Res proof of size $< 2^p$ then PROVER wins.

Contrapositive:

Lemma

Let $a, b \in \mathrm{R}^+$, $\frac{1}{a} + \frac{1}{b} = 1$, $p \in \mathrm{N}$, $\varphi \notin SAT$. If DEL wins then EVERY Tree Resolution proof for φ has size $\geq 2^p$.

PLAN: Get AWESOME strategy for DEL when $\varphi = PHP_n^m$.

KEY TO STRATEGY FOR DEL

Lemma

Let $a, b \in \mathrm{R}^+$. Let $n < m$. Let $\varphi = PHP_n^m$. There is a strategy for DEL that earns at least

$$\min\{\Omega(n \log b), \Omega(n^2 \log a)\}.$$

KEY to STRATEGY FOR DEL:

1. DEL does NOT allow two 1's in a column. EVER!!!!
2. DEL is wary of the all-0's row. But not too wary. DEL puts a 1 in a row if PROVER has put many 0's in that row.

STRATEGY FOR DEL

PROVER has picked x_{ij}.

1. If there is a i' such that $x_{i',j} = 1$ then set $x_{i,j} = 0$. (DEL gets no points, but averts DISASTER.)
2. If the ith row has $\frac{n}{2}$ 0's that PROVER put there, and no 1's, then DEL puts a 1 (DEL gets no points, but DEL delays an all-0 row.)
3. Otherwise defer to PROVER (and get some points!).

ANALYSE STRATEGY

1. Games over when some row is ALL 0's- say row i.
2. $\le \frac{n}{2}$ 0's set by PROVER. $\ge \frac{n}{2}$ 0's set by DEL.
3. x_{ij} set to 0 by DEL: $\exists x_{i'j}$ set to 1.
4. PROVER set $x_{i'j} = 1$: $\lg b$ points for DEL.
5. DEL set $x_{i'j} = 1$: PROVER set $\frac{n}{2}$ in that row to 0. $\frac{n}{2} \lg a$ points for DEL.
6. DEL gets $\ge \frac{n}{2} \min\{\lg b, \frac{n}{2} \lg a\} = \min\{\Omega(n \log b), \Omega(n^2 \log a)\}$.

UPSHOT

Theorem

For $n < m$ any Tree Res Proof of PHP_n^m requires $2^{\Omega(n \log n)}$ size.

Proof.

Let $a, b, p \in \mathrm{R}^+$ such that $\frac{1}{a} + \frac{1}{b} = 1$, to be determined. We use parameters a, b, p, PHP_n^m for game.

1. If DEL has winning strategy then ANY Tree Res Proof of PHP_n^m has size $\geq 2^{\Omega(p)}$.
2. There IS strategy, Del gets $\min\{\Omega(n \log b), \Omega(n^2 \log a)\}$.
3. Need a, b such that $\min\{\Omega(n \log b), \Omega(n^2 \log a)\} \geq \Omega(n \log n)$.

UPSHOT

Theorem

For $n < m$ any Tree Res Proof of PHP_n^m requires $2^{\Omega(n \log n)}$ size.

Proof.

Let $a, b, p \in \mathrm{R}^+$ such that $\frac{1}{a} + \frac{1}{b} = 1$, to be determined. We use parameters a, b, p, PHP_n^m for game.

1. If DEL has winning strategy then ANY Tree Res Proof of PHP_n^m has size $\geq 2^{\Omega(p)}$.
2. There IS strategy, Del gets $\min\{\Omega(n \log b), \Omega(n^2 \log a)\}$.
3. Need a, b such that $\min\{\Omega(n \log b), \Omega(n^2 \log a)\} \geq \Omega(n \log n)$.
   3.1 Set $b = \frac{n}{\ln n}$. HAVE: $n \lg b \geq \Omega(n \log n)$.
   3.2 Set $a = 1 + \frac{1}{b-1}$. HAVE: $\frac{1}{a} + \frac{1}{b} = 1$.
   3.3 $a = 1 + \frac{1}{b-1} \sim e^{1/b-1} \sim e^{1/b} = n^{1/n}$.
   3.4 HAVE: $n^2 \lg a \geq \Omega(n^2 \frac{\log n}{n}) = \Omega(n \log n)$.

OPEN PROBLEMS (Mine)

1. RAM_n^m is every 2-coloring of the edges of K_m has a COMPLETE MONO K_n. For $n = \lceil 0.5 \lg m \rceil$ this is TRUE. Can make into a formula.
2. Relevant Prover-Delayer Game:
   2.1 PROVER picks an EDGE (i, j).
   2.2 DEL either colors edge RED or BLUE or DEFERS.
   2.3 If defers then DEL gets a point.

   Strategies for Delayer lead to lower bounds on proofs of RAM_n^m.
3. UGLY: RAM_n^m is of SIZE n^m, NOT Poly.
4. BETTER: $RAMCYCLE_n^m$, $RAMPATH_n^m$, all have POLY number of clauses.
5. ALSO BETTER: c-COLORING $c^3 \times c^3$ GRID yields a mono rectangle.