Proof Complexity and Its Relations to SAT-Solving

Albert Atserias
Universitat Politècnica de Catalunya
Centre de Recerca Matemàtica
Barcelona, Catalonia, Spain

Overview of the talk

PART I: PROOF COMPLEXITY AND SAT

1. Propositional Logic
2. SAT-Solvers
3. Frege Systems
4. Cut-Free and Cut-Only Proofs

PART II: COMPLEXITY OF PROOF SEARCH

1. Proof Search and Automatability
2. Proof of NP-hardness for Resolution
3. An Open Problem

Part II

COMPLEXITY OF PROOF SEARCH

Automatability : Searching for *Short* Proofs

Definition [Bonet-Pitassi-Raz'1999]
A proof system P is automatable in time $T(s)$ if there is an algorithm that given a tautology F finds a P-proof of F in time $T(s^*)$, where s^* is the size of the smallest P-proof of F.

Automatability : Searching for *Short* Proofs

Definition [Bonet-Pitassi-Raz'1999]
A proof system P is automatable in time $T(s)$ if there is an algorithm that given a tautology F finds a P-proof of F in time $T(s^*)$, where s^* is the size of the smallest P-proof of F.

A Fundamental Question

Which proof systems are automatable in non-trivial time?

A Non-Trivial Notion

An Early Lower Bound:

Theorem [Krajicek-Pudlak'1994]
Extended Frege systems are not automatable in time $T(s)$,
unless n-bit RSA cryptosystem can be broken in time $T(\mathrm{poly}(n))$.

A Non-Trivial Notion

An Early Lower Bound:

Theorem [Krajicek-Pudlak'1994]
Extended Frege systems are not automatable in time $T(s)$,
unless n-bit RSA cryptosystem can be broken in time $T(\mathrm{poly}(n))$.

An Early Upper Bound:

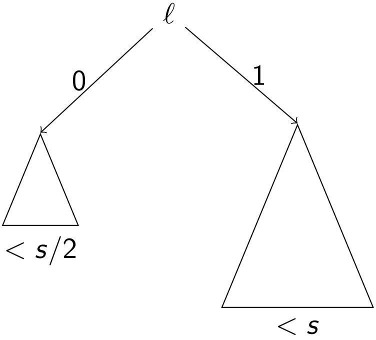
Theorem [Beame-Pitassi'1998]
Tree-like Resolution *is* automatable in time $T(s) = s^{O(\log s)}$.

Beame-Pitassi Algorithm

1. guess the root literal ℓ
   ($2n$ choices only)

2. recurse with parameter $s/2$
   (abort the branch if it fails)

3. recurse with parameter $s-1$
   (it must succeed).



$$T(n,s) \le 2nT(n-1,s/2) + T(n-1,s-1)$$

$$T(n,s) = n^{O(\log s)} \le s^{O(\log s)}.$$

Non-Automatability of Resolution

Theorem [Atserias-Müller'2019]
Resolution is not automatable in time $T(s)$, [poly-time]
unless n-variable SAT is solvable in time $T(\mathrm{poly}(n))$ [P $=$ NP].

Non-Automatability of Resolution

Theorem [Atserias-Müller'2019]
Resolution is not automatable in time $T(s)$, [poly-time]
unless n-variable SAT is solvable in time $T(\mathrm{poly}(n))$ [P = NP].

Theorem [de Rezende'2021]
Tree-Like Resolution is not automatable in time $T(s) = s^{o(\log s)}$,
unless n-variable SAT is solvable in randomized time $2^{o(n)}$.

Non-Automatability of Resolution

Theorem [Atserias-Müller'2019]
Resolution is not automatable in time $T(s)$, [poly-time]
unless n-variable SAT is solvable in time $T(\mathrm{poly}(n))$ [P $=$ NP].

Theorem [de Rezende'2021]
Tree-Like Resolution is not automatable in time $T(s) = s^{o(\log s)}$,
unless n-variable SAT is solvable in randomized time $2^{o(n)}$.

Notes:

- Compare with Beame-Pitassi algorithm!
- Improved earlier results of [Alekhnovich-Razborov'2001]
- Introduced a new method for proving non-automatability
- Correctness of the reduction involves proving a lower bound!

Proof Strategy for NP-Hardness

We want a polynomial-time reduction:

from	n-variable SAT
to	min proof-size approximation for Resolution (R).

$$F \xrightarrow{\mathrm{poly}(n)\ \text{time}} G_F$$

Requirements:

1. If F is satisfiable, then $\mathsf{SIZE}_{\mathsf{R}}(G_F) \leq \mathrm{poly}(n)$.
2. If F is unsatisfiable, then $\mathsf{SIZE}_{\mathsf{R}}(G_F) \not\leq \exp(\Omega(n))$.

Choice of the Formula G_F : the REF Formulas

$$\text{REF}_{F,s} \quad = \quad \text{“the CNF formula } F \text{ has an R-refutation of length } s\text{”}$$

Variables:

$D_{u,i,b}$: “line u contains variable x_i with sign $b \in \{0,1\}$”
$I_{u,j}$: “line u is an initial assumption; the j-th clause of F”
$V_{u,i}$: “line u is derived by resolving on variable x_i”
$L_{u,v}$: “line u is derived using v as left assumption”
$R_{u,v}$: “line u is derived using v as right assumption”
A_u : “line u is active; i.e., actually used in the proof”

Clauses (a sample):

$$\begin{array}{lll} \overline{A_u} \vee \overline{V_{u,i}} \vee \overline{L_{u,v}} \vee D_{v,i,1} & \overline{A_u} \vee \overline{V_{u,i}} \vee \overline{R_{u,v}} \vee D_{v,i,0} & \overline{D_{s,i,b}} \\ \overline{A_u} \vee \overline{V_{u,i}} \vee \overline{L_{u,v}} \vee A_v & \overline{A_u} \vee \overline{V_{u,i}} \vee \overline{R_{u,v}} \vee A_v & A_s \end{array}$$

$\dots$

Requirement 1 : The Upper Bound

If F is satisfiable, then $\mathsf{SIZE}_{\mathrm{R}}(\mathsf{REF}_{F,n^c}) \leq \mathrm{poly}(n)$.

Proof idea:

Use a satisfying assignment α of F to nail down the refutation!

Proof sketch:

- Prove that every active line contains a literal satisfied by α.
- Concretely, derive the clauses

$$T_u \quad := \quad \overline{A_u} \vee \bigvee_{i=1}^{n} D_{u,i,\alpha(i)} \qquad \text{for } u = 1, 2, \ldots, L.$$

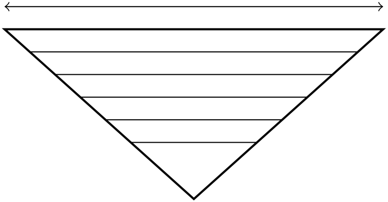
- Produce empty clause by resolving T_s with A_s and the $\overline{D_{s,i,b}}$.

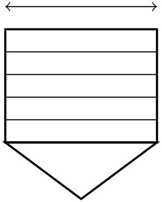
QED

If F is unsatisfiable, then $\mathsf{SIZE}_{\mathsf{R}}(\mathsf{REF}_{F,n^c}) \not\leq \exp(\Omega(n))$.

Proof idea:

Use a model β^* of $\mathsf{REF}_{F,2^n}$ to construct
a collection of "pseudo-models" β for REF_{F,n^c}.





The Lower Bound in Three Steps

- Identify a set H of α such that $\mathrm{REF}_{F,s}|_\alpha \cong \mathrm{REF}_{F,s/2}$.
- Here: let α set $1/2$ of all lines as inactive (but not the last).
- And let α also set all other variables of those lines.

- Identify a notion of weak clause made likely true by random α.
- Here: the clauses that mention more than $n/2$ lines.
- Calculation: $\Pr_{\alpha \in H}[C|_\alpha \neq 1] \leq (3/4)^{n/2}$.

- Prove that refutations of $\mathrm{REF}_{F,s/2}$ must contain weak clauses.
- Walk up the dag from empty clause to axioms, and do:
- Sustain a matching between active lines and the lines in β^*.
- The corresponding assignments are the "pseudo-models" β.

QED

Reminder

Theorem
Resolution is not automatable in time $T(s)$, [poly-time] ✓
unless n-variable SAT is solvable in time $T(\mathrm{poly}(n))$ [P $=$ NP].

Theorem
Tree-Like Resolution is not automatable in time $T(s) = s^{o(\log s)}$,
unless n-variable SAT is solvable in randomized time $2^{o(n)}$.

The Tree-Like Case

We want a reduction:

from n-variable SAT
to min proof-size approximation for tree-like R (called R^*).

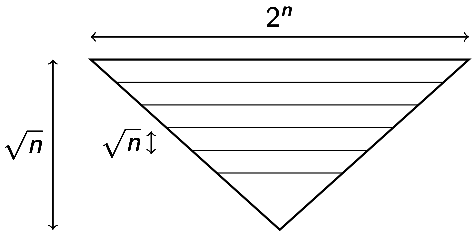
$$F \xrightarrow{\exp(o(n)) \text{ time}} G_F$$

Requirements:

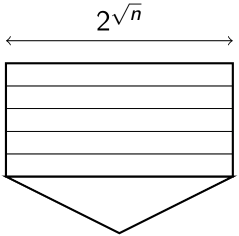
1. If F is satisfiable, then $\mathrm{SIZE}_{\mathrm{R}^*}(G_F) \leq \exp(O(\sqrt{n}))$.
2. If F is unsatisfiable, then $\mathrm{SIZE}_{\mathrm{R}^*}(G_F) \not\leq \exp(\Omega(n))$.

Key Observation:

In the "F is unsatisfiable" case,
the model β^* of $\mathrm{REF}_{F,2^n}$ happens to be:
tree-like and layered, and have depth n.



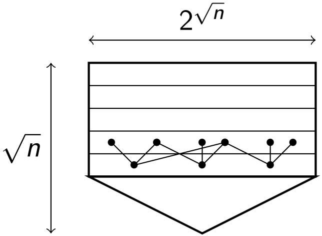
$\beta^* \models \mathrm{REF}_{F,2^n}$



$\beta \not\models \neg\mathrm{REF}_{F,2^{\sqrt{n}}}$

Modification of the Formula in More Details

- Modify the formula G_F; now $\mathrm{REF}_{F,s}|_\gamma$ with $s = 2^{\sqrt{n}}$.
- The γ restricts A, D, I, V, L, R in a way compatible with β^*:
- Instead of arbitrary dag-depth, impose depth n.
- Instead of arbitrary structure, impose $\sqrt{n}$ layers of depth $\sqrt{n}$.
- Instead of $\mathrm{poly}(n)$-size layers, allow layers of size $2^{\sqrt{n}}$.
- Instead of full connectivity between layers, place expanders.
- Their bounded degree d ensures tree-like size $d^{\sqrt{n}} = 2^{O(\sqrt{n})}$.
- Their expansion property ensures matchability with β^*.



Reminder

Theorem
Resolution is not automatable in time $T(s)$, [poly-time]
unless n-variable SAT is solvable in time $T(\mathrm{poly}(n))$ [P $=$ NP]. ✓

Theorem
Tree-Like Resolution is not automatable in time $T(s) = s^{o(\log s)}$,
unless n-variable SAT is solvable in randomized time $2^{o(n)}$. ✓

Big Remaining Open Problem

WEAK AUTOMATABILITY OF RESOLUTION?

For Resolution specifically:

Is it computationally feasible to
distinguish satisfiable formulas
from shortly refutable formulas?

WEAK AUTOMATABILITY OF RESOLUTION?

For Resolution specifically:

Is it computationally feasible to
distinguish satisfiable formulas
from shortly refutable formulas?

Notes:

WEAK AUTOMATABILITY OF RESOLUTION?

For Resolution specifically:

Is it computationally feasible to
distinguish satisfiable formulas
from shortly refutable formulas?

Notes:

- Automatability is about short vs. not short refutability.

WEAK AUTOMATABILITY OF RESOLUTION?

For Resolution specifically:

Is it computationally feasible to
distinguish satisfiable formulas
from shortly refutable formulas?

Notes:

- Automatability is about short vs. not short refutability.
- Weak automatability is about short vs. impossible refutability.

WEAK AUTOMATABILITY OF RESOLUTION?

For Resolution specifically:

Is it computationally feasible to
distinguish satisfiable formulas
from shortly refutable formulas?

Notes:

- Automatability is about short vs. not short refutability.
- Weak automatability is about short vs. impossible refutability.
- Therefore: it cannot be harder than NP $\cap$ co-NP.

Big Remaining Open Problem

WEAK AUTOMATABILITY OF RESOLUTION?

For Resolution specifically:

Is it computationally feasible to
distinguish satisfiable formulas
from shortly refutable formulas?

Notes:

- Automatability is about short vs. not short refutability.
- Weak automatability is about short vs. impossible refutability.
- Therefore: it cannot be harder than NP $\cap$ co-NP.
- For Resolution, the problem is PARITY GAMES hard [BPT].

WEAK AUTOMATABILITY OF RESOLUTION?

For Resolution specifically:

Is it computationally feasible to
distinguish satisfiable formulas
from shortly refutable formulas?

Notes:

- Automatability is about short vs. not short refutability.
- Weak automatability is about short vs. impossible refutability.
- Therefore: it cannot be harder than NP $\cap$ co-NP.
- For Resolution, the problem is PARITY GAMES hard [BPT].
- For (Extended) Frege, the problem is RSA-hard [KP,BPR].

END OF PART II
AND OF TUTORIAL