Proof Complexity of Resolution over linear inequalities

Stefan Dantchev

Computer Science, Durham University

1. Propositional Proof Complexity
   - Within Computational Complexity
   - Proof Systems and Contradictions
2. Resolution over linear inequalities with integral coefficients
   - Resolution as computational procedure
   - Stabbing Planes
   - Lower bounds for SP
3. Conclusion and open problems

1. Propositional Proof Complexity
   - Within Computational Complexity
   - Proof Systems and Contradictions

2. Resolution over linear inequalities with integral coefficients
   - Resolution as computational procedure
   - Stabbing Planes
   - Lower bounds for SP

3. Conclusion and open problems

$\mathcal{P}$ vs $\mathcal{NP}$ and Circuit Complexity

- $\mathcal{P}$ is the class of decision problems that are efficiently solvable (i.e. in polynomial time).
- $\mathcal{NP}$ is the class of decision problems whose yes-instances are efficiently verifiable.

$\mathcal{P}$ vs $\mathcal{NP}$ and Circuit Complexity

- $\mathcal{P}$ is the class of decision problems that are efficiently solvable (i.e. in polynomial time).
- $\mathcal{NP}$ is the class of decision problems whose yes-instances are efficiently verifiable.
- Generic example of an $\mathcal{NP}$-complete problem is SAT(isfiability): given a boolean formula in cnf, is there a satisfying assignment? (Cook, 1971)

$\mathcal{P}$ vs $\mathcal{NP}$ and Circuit Complexity

- $\mathcal{P}$ is the class of decision problems that are efficiently solvable (i.e. in polynomial time).
- $\mathcal{NP}$ is the class of decision problems whose yes-instances are efficiently verifiable.
- Generic example of an $\mathcal{NP}$-complete problem is SAT(isfiability): given a boolean formula in cnf, is there a satisfying assignment? (Cook, 1971)
- $\mathcal{P}$ vs $\mathcal{NP}$ question asks if a solution that has an efficiently checkable proof of existence can be easily found.

$\mathcal{P}$ vs $\mathcal{NP}$ and Circuit Complexity

- $\mathcal{P}$ is the class of decision problems that are efficiently solvable (i.e. in polynomial time).
- $\mathcal{NP}$ is the class of decision problems whose yes-instances are efficiently verifiable.
- Generic example of an $\mathcal{NP}$-complete problem is SAT(isfiability): given a boolean formula in cnf, is there a satisfying assignment? (Cook, 1971)
- $\mathcal{P}$ vs $\mathcal{NP}$ question asks if a solution that has an efficiently checkable proof of existence can be easily found.
- To separate $\mathcal{P}$ and $\mathcal{NP}$, one only needs to prove a super-polynomial lower bound for a (unrestricted) boolean circuit that solves SAT (or any other $\mathcal{NP}$-problem).

$\mathcal{P}$ vs $\mathcal{NP}$ and Circuit Complexity

- $\mathcal{P}$ is the class of decision problems that are efficiently solvable (i.e. in polynomial time).
- $\mathcal{NP}$ is the class of decision problems whose yes-instances are efficiently verifiable.
- Generic example of an $\mathcal{NP}$-complete problem is SAT(isfiability): given a boolean formula in cnf, is there a satisfying assignment? (Cook, 1971)
- $\mathcal{P}$ vs $\mathcal{NP}$ question asks if a solution that has an efficiently checkable proof of existence can be easily found.
- To separate $\mathcal{P}$ and $\mathcal{NP}$, one only needs to prove a super-polynomial lower bound for a (unrestricted) boolean circuit that solves SAT (or any other $\mathcal{NP}$-problem).
- What about UNSAT, i.e. $\text{co}-\mathcal{NP}$?

Proof Complexity and $\mathcal{NP}$ vs co$-\mathcal{NP}$

- UNSAT: Given a boolean formula in cnf, is it a contradiction?

Proof Complexity and $\mathcal{NP}$ vs co$-\mathcal{NP}$

- UNSAT: Given a boolean formula in cnf, is it a contradiction?
- If all yes-instances of UNSAT have efficiently checkable proofs, $\mathcal{NP} = \text{co}-\mathcal{NP}$.

Proof Complexity and $\mathcal{NP}$ vs co$-\mathcal{NP}$

- UNSAT: Given a boolean formula in cnf, is it a contradiction?
- If all yes-instances of UNSAT have efficiently checkable proofs, $\mathcal{NP} = \text{co}-\mathcal{NP}$.
- On the other hand, $\mathcal{NP} \neq \text{co}-\mathcal{NP}$ trivially implies $\mathcal{P} \neq \mathcal{NP}$.

Proof Complexity and $\mathcal{NP}$ vs co$-\mathcal{NP}$

- UNSAT: Given a boolean formula in cnf, is it a contradiction?
- If all yes-instances of UNSAT have efficiently checkable proofs, $\mathcal{NP} = \text{co}-\mathcal{NP}$.
- On the other hand, $\mathcal{NP} \neq \text{co}-\mathcal{NP}$ trivially implies $\mathcal{P} \neq \mathcal{NP}$.
- A proof system is an efficiently computable function that takes a proof and returns the tautology being proven. (Cook and Reckhow, 1979)

Proof Complexity and $\mathcal{NP}$ vs co$-\mathcal{NP}$

- UNSAT: Given a boolean formula in cnf, is it a contradiction?
- If all yes-instances of UNSAT have efficiently checkable proofs, $\mathcal{NP} = \text{co}-\mathcal{NP}$.
- On the other hand, $\mathcal{NP} \neq \text{co}-\mathcal{NP}$ trivially implies $\mathcal{P} \neq \mathcal{NP}$.
- A proof system is an efficiently computable function that takes a proof and returns the tautology being proven. (Cook and Reckhow, 1979)
- A polynomially-bounded proof system, i.e. one that allows for a polynomial-size proof of any tautology, exists iff $\mathcal{NP} = \text{co}-\mathcal{NP}$.

Proof Complexity and $\mathcal{NP}$ vs co$-\mathcal{NP}$

- UNSAT: Given a boolean formula in cnf, is it a contradiction?
- If all yes-instances of UNSAT have efficiently checkable proofs, $\mathcal{NP} = \text{co}-\mathcal{NP}$.
- On the other hand, $\mathcal{NP} \neq \text{co}-\mathcal{NP}$ trivially implies $\mathcal{P} \neq \mathcal{NP}$.
- A proof system is an efficiently computable function that takes a proof and returns the tautology being proven. (Cook and Reckhow, 1979)
- A polynomially-bounded proof system, i.e. one that allows for a polynomial-size proof of any tautology, exists iff $\mathcal{NP} = \text{co}-\mathcal{NP}$.
- Research programme: **prove lower bounds** for stronger and **stronger proof systems**.

1. Propositional Proof Complexity
   - Within Computational Complexity
   - Proof Systems and Contradictions
2. Resolution over linear inequalities with integral coefficients
   - Resolution as computational procedure
   - Stabbing Planes
   - Lower bounds for SP
3. Conclusion and open problems

"Natural" proof systems: Resolution and bd-Frege

- Resolution
  - operates on a formula in cnf, taken as a set of clauses;
  - has a single derivation rule: $\frac{A \vee v \quad \neg v \vee B}{A \vee B}$;
  - derives the empty clause iff the original cnf is a contradiction.
- Bounded-depth Frege
  - is a "text-book" Hilbert-style proof system;
  - normally over the basis $\vee$, $\wedge$, $\neg$;
  - and where each proof line is a constant-depth formula;
  - e.g. derivation rules $\frac{\varphi \vee \psi \quad \neg\psi \vee \pi}{\varphi \vee \pi}$, $\frac{\varphi \vee \psi \quad \pi \vee \xi}{\varphi \vee \pi \vee (\psi \wedge \xi)}$ and axioms $\frac{}{\varphi \vee \neg\varphi}$ for any bd-formulae φ, π, ψ, ξ.
- Many others.

"Natural" contradictions: Pigeon-Hole Principle and Ordering Principle

Pigeon-Hole Principle PHP^m_n with m pigeons and n holes, $m > n$, has

- variables p_{ij}, which stand for pigeon i going into hole j, and
- clauses

$$\begin{array}{rl} \bigvee_{j=1}^{n} p_{ij} & 1 \leq i \leq m \\ \neg p_{ij} \vee \neg p_{i'j} & 1 \leq i < i' \leq m,\ 1 \leq j \leq n \end{array}$$

Ordering Principle OP_n says there is transitive, anti-reflexive relation on n items with no least point.

“Natural” contradictions: Tseitin contradictions

Given an undirected graph $G = (V, E)$ with constants in $a_u \in \mathbb{Z}_2$, $u \in V$, and such that $\bigoplus_{u \in V} a_u = 1$, introduce

- variables x_e for an edge $e \in E$, and
- clauses stating that the variables at each vertex sum up to the constant at the vertex:

$$\bigoplus_{v:\{u,v\} \in E} x_{\{u,v\}} = a_u \qquad u \in V.$$

Some known lower bounds

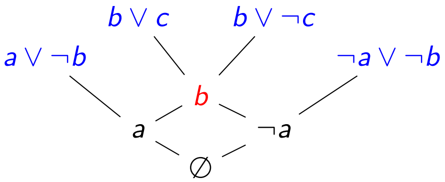
- lower bound for Tseitin in regular Resolution. (Tseitin, 1968)
- exponential lower bound for PHP^{n+1}_{n} in Resolution. (Haken, 1985)
- sub-exponential for PHP^{n+1}_{n} in bd-Frege. (Ajtai, 1988, 1994), (Pitassi, Beame and Impagliazzo,1993), (Krajıcek, Pudlak and Woods, 1995)
- complexity gap for tree-like resolution: a propositional contradiction that is expressible as a first-order formula is hard if and only if the formula has an infinite model. (Riis, 2001)

1. Propositional Proof Complexity
   - Within Computational Complexity
   - Proof Systems and Contradictions
2. Resolution over linear inequalities with integral coefficients
   - Resolution as computational procedure
   - Stabbing Planes
   - Lower bounds for SP
3. Conclusion and open problems

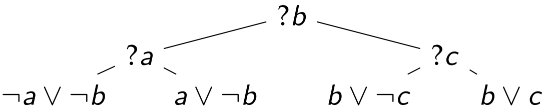
Resolution

Resolution refutation of the clause set

$a \vee \neg b \quad b \vee c \quad b \vee \neg c \quad \neg a \vee \neg b$



D(P)LL



- Queries values of a variable and backtracks (true to the left, false to the right).
- A branch is closed as soon as the current partial assignment violates a clause.

1. Propositional Proof Complexity
   - Within Computational Complexity
   - Proof Systems and Contradictions
2. Resolution over linear inequalities with integral coefficients
   - Resolution as computational procedure
   - Stabbing Planes
   - Lower bounds for SP
3. Conclusion and open problems

D(P)LL on linear inequalities a.k.a. **Stabbing Planes**

D(P)LL on set of linear inequalities

$$\begin{aligned} a + (1 - b) &\geq 1 \\ b + c &\geq 1 \\ b + (1 - c) &\geq 1 \\ (1 - a) + (1 - b) &\geq 1 \end{aligned}$$

- Queries values of linear inequalities with integral coefficients and backtracks ($\alpha^T x \leq \beta$ to the left, $\alpha^T x \geq \beta + 1$ to the right).
- A branch is closed as soon as the current set of inequalities together with the original clause set is an inconsistent linear program.
- The slab, $\beta < \alpha^T x < \beta + 1$, kills fractional points.

1. Propositional Proof Complexity
   - Within Computational Complexity
   - Proof Systems and Contradictions

2. Resolution over linear inequalities with integral coefficients
   - Resolution as computational procedure
   - Stabbing Planes
   - Lower bounds for SP

3. Conclusion and open problems

General Method

1. Create a "big" set of admissible fractional points, i.e. that are consistent with the given set of inequalities. Each such point must be killed by some slab.
2. A query of "small" support can kill many fractional points. We erase such a query by setting all variables in the support to integral values.
3. Only queries of "big" support are left, but they have "small" slabs (that can't kill many fractional points).

Example

Simple PHP:

$$\sum_{j=1}^{n} x_j \geq 2$$

$$x_i + x_j \leq 1, 1 \leq i < j \leq n.$$

An admissible point is a vector of zeros and halves that contains at least four halves.

In a query of support not greater than $\sqrt{n}$ all variables could be set to zero. If there are more than $\sqrt[3]{n}$ such queries, we are done.

A query of support greater that $\sqrt{n}$ can't kill more than $c/\sqrt[4]{n}$ fraction of the admissible points.

Therefore, there must be at least $\sqrt[5]{n}$ queries in any SP refutation of the simple PHP.

More generally

- The method applies to the (standard) PHP and gives an optimal, logarithmic, depth lower bound.

More generally

- The method applies to the (standard) PHP and gives an optimal, logarithmic, depth lower bound.
- It also applies to the OP, but the logarithmic depth lower bound is probably not optimal. **More generally, we don't know how to go beyond logarithmic.**

More generally

- The method applies to the (standard) PHP and gives an optimal, logarithmic, depth lower bound.
- It also applies to the OP, but the logarithmic depth lower bound is probably not optimal. **More generally, we don't know how to go beyond logarithmic.**
- Does it apply to any first-order principle that has an infinite model?

More generally

- The method applies to the (standard) PHP and gives an optimal, logarithmic, depth lower bound.
- It also applies to the OP, but the logarithmic depth lower bound is probably not optimal. **More generally, we don't know how to go beyond logarithmic.**
- Does it apply to any first-order principle that has an infinite model?
- **How do we get more than logarithmic depth lower bound!?**