Proof Complexity as a Computational Lens

Final Lecture

Jakob Nordström

University of Copenhagen and Lund University

February 27, 2026

Outline

1. Proof Systems Covered in This Course
   - Resolution
   - Nullstellensatz and Polynomial Calculus
   - Cutting Planes
2. Proof Systems That We Didn't Manage to Cover
   - Stabbing Planes
   - Sherali–Adams and Sum-of-Squares
   - Resolution over Parities
3. More Proof Systems and Perspectives
   - Even Stronger Methods of Reasoning
   - Other Techniques
   - Applications of Proof Complexity in Other Areas

An Apology

- Slides prepared in great haste
- Pretty much all references missing
- See lecture notes for concrete lectures for more details
- Proof complexity chapter in *Handbook of Satisfiability* [BN21] should be good source
- Krajíček's book *Proof Complexity* [Kra19] better for advanced topics
- And semialgebraic proof systems covered in F&TTCS survey *Semialgebraic Proofs and Efficient Algorithm Design* [FKP19]

Resolution Length/Size Lower Bounds

In our lectures on resolution we covered some "classic" size lower bounds:

- Pigeonhole principle (PHP) formulas
- Tseitin formulas
- Random k-CNF formulas
- Clique-colouring formulas

Resolution Length/Size Lower Bounds

In our lectures on resolution we covered some "classic" size lower bounds:

- Pigeonhole principle (PHP) formulas
- Tseitin formulas
- Random k-CNF formulas
- Clique-colouring formulas

And some more recent results:

- Trade-offs between different complexity measures for resolution (length/size, width, space)
- Clique lower bound for regular resolution
- Non-automatability (efficient proof search for resolution is NP-hard)

Proof Techniques for Resolution

- Prosecutor-defendant game
- Random restrictions
- Size-width lower bounds
- Monotone feasible interpolation
- Decision tree reductions

Some Resolution Topics We Didn't Cover

- Pseudorandom generators (more about this later)
- Separations between different subsystems of resolution
- Polynomial simulation of resolution by conflict-driven clause learning (CDCL)

Resolution Width

Resolution width lower bounds for k-CNF formulas imply:

- length/size lower bounds (if width $\gg \sqrt{\#\text{ variables}}$)
- clause space lower bounds
- total space lower bounds (width squared)

Resolution Width

Resolution width lower bounds for k-CNF formulas imply:

- length/size lower bounds (if width $\gg \sqrt{\#\text{ variables}}$)
- clause space lower bounds
- total space lower bounds (width squared)

But width and clause space (almost) maximally separated

Open Problems for Resolution Space

- Does linear clause space lower bounds imply width/length lower bounds?
- Must a refutation in constant clause space also have polynomial length?
- Possible to exhibit supercritical trade-offs for
  - length/size vs. clause space with better parameters?
  - width vs. clause space for space larger than formula size?

More Open Problems for Resolution

- Tight bounds for weak PHP formulas
  - with n pigeonholes and $\gg n^2$ pigeons
  - also for graph PHP formulas
  - use and refine pseudo-width technique?

More Open Problems for Resolution

- Tight bounds for weak PHP formulas
  - with n pigeonholes and $\gg n^2$ pigeons
  - also for graph PHP formulas
  - use and refine pseudo-width technique?
- Clique lower bounds for general resolution
  - worst-case
  - average case

More Open Problems for Resolution

- Tight bounds for weak PHP formulas
  - with n pigeonholes and $\gg n^2$ pigeons
  - also for graph PHP formulas
  - use and refine pseudo-width technique?
- Clique lower bounds for general resolution
  - worst-case
  - average case
- Understand resolution complexity of NP-complete problems? (Using good encodings)

More Open Problems for Resolution

- Tight bounds for weak PHP formulas
  - with n pigeonholes and $\gg n^2$ pigeons
  - also for graph PHP formulas
  - use and refine pseudo-width technique?
- Clique lower bounds for general resolution
  - worst-case
  - average case
- Understand resolution complexity of NP-complete problems? (Using good encodings)
- How hard is it to search for a shortest resolution refutation?

Nullstellensatz

- Only talked briefly about Nullstellensatz
- More interested in polynomial calculus
- Main focus has been on degree measure
- Degree lower bounds $\Leftrightarrow$ existence of designs

Nullstellensatz

- Only talked briefly about Nullstellensatz
- More interested in polynomial calculus
- Main focus has been on degree measure
- Degree lower bounds $\Leftrightarrow$ existence of designs

Open problems:

- Size-degree trade-offs for Nullstellensatz with dual variables
- Also without dual variables, would be nice to have stronger trade-offs — related to reversible pebbling
- Size lower bounds for more concise representation of polynomials than linear combination of monomials — leads to superstrong ideal proof system!

Polynomial Calculus

- Models Gröbner basis computations
- Assumes polynomials represented as linear combinations of monomials
- Exponentially stronger than resolution (assuming use of dual variables)
- Again main focus on degree complexity measure
- Degree lower bounds from pseudo-reductions faking polynomial ideal reductions
- Superpolynomial size lower bounds for constant-degree input if degree $\gg \sqrt{\# \text{ variables}}$
- Less tools in toolbox than for resolution

Some Results for Polynomial Calculus

Some hard formulas for resolution are easy for polynomial calculus:

- Tseitin formulas on expander graphs if $\mathbb{F} = \mathrm{GF}(2)$ (do Gaussian elimination)
- Onto functional pigeonhole principle over any field (count modulo characteristic)

Some Results for Polynomial Calculus

Some hard formulas for resolution are easy for polynomial calculus:

- Tseitin formulas on expander graphs if $\mathbb{F} = \mathrm{GF}(2)$ (do Gaussian elimination)
- Onto functional pigeonhole principle over any field (count modulo characteristic)

But other formulas remain hard for polynomial calculus:

- Tseitin-like formulas for counting $\bmod\ p$ if $p \neq$ field characteristic
- "vanilla" PHP, onto PHP, and functional PHP formulas
- Random k-CNF formulas
- Colouring formulas (worst-case and average-case)

Some Questions Motivated by Algebraic Solving

- Gröbner basis algorithm works with respect to fixed order — obtain proof complexity separations between different orders?
- Efficient algorithms for polynomials with dual variables?
- Conflict-driven algebraic solving?

Polynomial Calculus: Additional Topics

Some topics we didn't talk about:

- Pseudorandom generators
- Lower bound techniques for concrete field characteristics
  - change to "Fourier basis"
  - immunity (axioms without low-degree implications)

Open Problems for Polynomial Calculus Size and Degree

- Combine immunity with generalized constraint-variable incidence graphs (CVIGs)?
- Improve techniques for degree lower bounds
  - dense linear ordering (DLO) formulas
  - homomorphism problems
  - dichotomy results for constraint satisfaction problems (CSPs)
- Lower bounds for pseudorandom generators
- Size lower bounds without using degree
  - weak PHP formulas
  - clique formulas

Open Problems for Polynomial Calculus Space

- Separate monomial space from resolution clause space(?)
- Optimal monomial space lower bounds for
  - Tseitin formulas
  - Functional PHP formulas
- Monomial space $\geq$ resolution width?
- Monomial space lower bounds for pebbling formulas
- Separations of degree and space independent of characteristic
- Supercritical size-space trade-offs independent of characteristic
- Total space lower bounds for polynomial-size formulas

(Easier to prove some space lower bounds without dual variables)

Polynomial Calculus over Roots of Unity

- Some recent, quite mysterious, results — can we gain better understanding?
- Prove implication degree lower bound $\Rightarrow$ size lower bound for single formula?
- Clean general result saying that if
  - constraint-variable incidence graph is expander and
  - constraints have property $\mathcal{P}$

  then size lower bound follows?
- Transformation between $\{0, 1\}$ and roots of unity can be viewed as extension variables — possible to deal with more general definitions?
- What about space lower bounds for polynomial calculus over roots of unity?

Cutting Planes

Recap of some basics

- Models 0–1 integer linear programming
- Exponentially stronger than resolution
- Incomparable to polynomial calculus
- Much more technically challenging to prove lower bounds

Cutting Planes

Recap of some basics

- Models 0–1 integer linear programming
- Exponentially stronger than resolution
- Incomparable to polynomial calculus
- Much more technically challenging to prove lower bounds

Proof techniques:

- Monotone feasible interpolation
- Lifting theorems in "classic" communication complexity
- Lifting theorems in "DAG-like" communication complexity (more recent)
- Bottleneck counting (very recent)

Open Problems for Cutting Planes Size

- Better parameters for DAG-like lifting
- Proof techniques for non-lifted formulas
- Proof techniques for distinguishing syntactic derivation rules (e.g., different cuts)
- Lower bounds for random k-CNF formulas
- Is cutting planes with polynomially bounded coefficients weaker than general cutting planes?

Open Problems for Cutting Planes Space

- General cutting planes refutes any infeasible 0–1 ILP in line space 5
- Possible to prove line space lower bounds for cutting planes with polynomially bounded coefficients?
- True trade-offs for cutting planes with polynomially bounded coefficients that don't apply to general cutting planes?
- Related problems:
  - Round-efficient lifting theorems in other settings
  - "Algorithmic" parity decision tree lower bounds for pebbling formulas
- Size-space trade-offs for general cutting planes with (much) better parameters would also be nice

Algorithmic Challenges for Pseudo-Boolean Solving

Pseudo-Boolean (PB) solvers use cutting planes + SAT-inspired methods for 0–1 ILPs

Challenging to make competitive with conflict-driven clause learning (CDCL)

Algorithmic Challenges for Pseudo-Boolean Solving

Pseudo-Boolean (PB) solvers use cutting planes + SAT-inspired methods for 0–1 ILPs

Challenging to make competitive with conflict-driven clause learning (CDCL)

1. Dealing with 0–1 linear inequalities instead of clauses
   - How to detect unit propagation efficiently?
   - How to keep coefficient sizes down to make integer arithmetic feasible?
   - How to compare and assess quality of constraints?

Algorithmic Challenges for Pseudo-Boolean Solving

Pseudo-Boolean (PB) solvers use cutting planes + SAT-inspired methods for 0–1 ILPs

Challenging to make competitive with conflict-driven clause learning (CDCL)

1. Dealing with 0–1 linear inequalities instead of clauses
   - How to detect unit propagation efficiently?
   - How to keep coefficient sizes down to make integer arithmetic feasible?
   - How to compare and assess quality of constraints?
2. Designing search and conflict analysis
   - Cutting planes much smarter method of reasoning than resolution
   - But this also makes it trickier to design smart search algorithms
   - Also harder to compare and assess quality of 0–1 linear inequalities

Algorithmic Challenges for Pseudo-Boolean Solving

Pseudo-Boolean (PB) solvers use cutting planes + SAT-inspired methods for 0–1 ILPs

Challenging to make competitive with conflict-driven clause learning (CDCL)

1. Dealing with 0–1 linear inequalities instead of clauses
   - How to detect unit propagation efficiently?
   - How to keep coefficient sizes down to make integer arithmetic feasible?
   - How to compare and assess quality of constraints?
2. Designing search and conflict analysis
   - Cutting planes much smarter method of reasoning than resolution
   - But this also makes it trickier to design smart search algorithms
   - Also harder to compare and assess quality of 0–1 linear inequalities
3. Pseudo-Boolean solvers terrible for CNF input
   - Can try to rewrite CNF to more helpful 0-1 linear inequalities
   - Tricky to get this to work well in practice

Stabbing Planes

- Stabbing planes introduced in [BFI$^+$18] to model (more modern) 0–1 ILP solving
- Decision tree that
  - branches over 0–1 linear inequalities
  - gets LP solving for free (so terminate when residual LP infeasible over $\mathbb{R}$)
- Originally believed to be much stronger than cutting planes
- But stabbing planes with polynomially bounded coefficients simulated by general cutting planes with a quasi-polynomial blow-up [DT20, FGI$^+$21]
- And recently, lower bounds for stabbing planes shown via interpolation [GP24]

Sherali–Adams (SA) and Sum-of-Squares (SoS)

Refutation of $p_i \in \mathbb{R}[x_1, \ldots, x_n]$, $i \in [m]$, and $x_j^2 - x_j$, $j \in [n]$

Nullstellensatz

$$\sum_{i=1}^{m} q_i \cdot p_i + \sum_{j=1}^{n} r_j \cdot (x_j^2 - x_j) = 1$$

Sherali–Adams (SA) and Sum-of-Squares (SoS)

Refutation of $p_i \in \mathbb{R}[x_1, \ldots, x_n]$, $i \in [m]$, and $x_j^2 - x_j$, $j \in [n]$

Nullstellensatz

$$\sum_{i=1}^{m} q_i \cdot p_i + \sum_{j=1}^{n} r_j \cdot (x_j^2 - x_j) = -1$$

Sherali–Adams (SA) and Sum-of-Squares (SoS)

Refutation of $p_i \in \mathbb{R}[x_1, \ldots, x_n]$, $i \in [m]$, and $x_j^2 - x_j$, $j \in [n]$

Nullstellensatz

$$\sum_{i=1}^{m} q_i \cdot p_i + \sum_{j=1}^{n} r_j \cdot (x_j^2 - x_j) = -1$$

Sherali–Adams (SA) $(\alpha_k \in \mathbb{R}^+)$

$$\sum_{i=1}^{m} q_i \cdot p_i + \sum_{j=1}^{n} r_j \cdot (x_j^2 - x_j) + \sum_{k=1}^{t} \alpha_k \prod_{i \in \mathcal{P}_t} (1 - x_i) \cdot \prod_{j \in \mathcal{N}_t} x_j = -1$$

Sherali–Adams (SA) and Sum-of-Squares (SoS)

Refutation of $p_i \in \mathbb{R}[x_1, \ldots, x_n]$, $i \in [m]$, and $x_j^2 - x_j$, $j \in [n]$

Nullstellensatz

$$\sum_{i=1}^{m} q_i \cdot p_i + \sum_{j=1}^{n} r_j \cdot (x_j^2 - x_j) = -1$$

Sherali–Adams (SA) $(\alpha_k \in \mathbb{R}^+)$

$$\sum_{i=1}^{m} q_i \cdot p_i + \sum_{j=1}^{n} r_j \cdot (x_j^2 - x_j) + \sum_{k=1}^{t} \alpha_k \prod_{i \in \mathcal{P}_t} (1 - x_i) \cdot \prod_{j \in \mathcal{N}_t} x_j = -1$$

Sum-of-squares (SoS) $(s_k \in \mathbb{R}[x_1, \ldots, x_n])$

$$\sum_{i=1}^{m} q_i \cdot p_i + \sum_{j=1}^{n} r_j \cdot (x_j^2 - x_j) + \sum_{k=1}^{s} s_k^2 = -1$$

Sherali–Adams, Sum-of-Squares, and Relations to Other Proof Systems

Sherali–Adams models linear programming (LP) hierarchies

Sum-of-squares models semidefinite programming (SDP) hierarchies
Strong connections to several best known approximation algorithms
(But Tseitin formulas are hard)

Sherali–Adams, Sum-of-Squares, and Relations to Other Proof Systems

Sherali–Adams models linear programming (LP) hierarchies

Sum-of-squares models semidefinite programming (SDP) hierarchies
Strong connections to several best known approximation algorithms
(But Tseitin formulas are hard)

Strict hierarchy (over $\mathbb{R}$):

- Nullstellensatz
- Sherali–Adams
- Sum of squares

Sum of squares is strictly stronger than polynomial calculus (over $\mathbb{R}$)
Sherali-Adams and polynomial calculus are incomparable [Ber18]

More Results and Open Problems for Sherali–Adams and Sum-of-Squares

- Separation between general Sherali–Adams and Sherali–Adams with polynomially bounded coefficients (unary Sherali–Adams or uSA) [GHJ$^+$24]
- What about different coefficient sizes in sum-of-squares?
- Average-case clique lower bounds for unary Sherali–Adams [dRPR23]
- Average-case colouring lower bounds for SoS [PX25], but (much) worse parameters than for polynomial calculus
- Size-degree lower bounds analogous to resolution [BW01] and polynomial calculus [IPS99] hold also for Sherali–Adams and SoS [AH19]
- What about size-degree trade-offs?
- Or non-automatability results?

Resolution over Parities

- Resolution, but clauses are disjunctions over parities
- First obstacle towards proving lower bounds for bounded-depth Frege with MOD connectives (more later)
- Currently very active area of research
- Size lower bounds, but only for bounded depth
- Current barrier at quadratic depth
- Better lifting theorems needed (ideally DAG-like)

Frege Proof Systems

- Standard natural deduction proof system taught in intro logics course
- Different flavours are polynomially equivalent
- Currently seems way beyond techniques for (unconditional) lower bounds
- Even lack of good candidates for hard formulas (except random k-CNF and other formulas that are too hard to prove lower bounds for)
- What about conditional lower bounds for assumptions weaker than NP $\neq$ coNP?

Bounded-Depth Frege Proof Systems

k-DNF resolution: clauses are k-DNF formulas (disjunctions of conjunctions)

- Random k-CNF formulas are hard
- Weak PHP formulas are not well understood
- Random restrictions turn into switching lemmas

Bounded-Depth Frege Proof Systems

k-DNF resolution: clauses are k-DNF formulas (disjunctions of conjunctions)

- Random k-CNF formulas are hard
- Weak PHP formulas are not well understood
- Random restrictions turn into switching lemmas

Bounded-depth Frege: formulas of arbitrary but constant depth

- Lower bounds for
  - PHP formulas
  - Tseitin formulas
- But weak PHP formulas are easy
- Major challenges to prove lower bounds for
  - random k-CNF formulas
  - random k-XOR formulas (not Tseitin formulas)

Bounded-Depth Frege and Circuit Complexity

Known results in circuit complexity:

- Depth hierarchy for bounded-depth circuits
- Strong lower bounds for bounded-depth circuits with MOD gates

Bounded-Depth Frege and Circuit Complexity

Known results in circuit complexity:

- Depth hierarchy for bounded-depth circuits
- Strong lower bounds for bounded-depth circuits with MOD gates

Analogous problems remain open in proof complexity:

- Depth hierarchy for bounded-depth Frege (for CNF formulas)
- Lower bounds for bounded-depth Frege with MOD connectives (this is why resolution over parities is so interesting)
- Switching lemmas for bounded-depth Frege are very complex
- Need other tools

Extended and Substitution Frege Proof Systems

- Extended Frege: Introduce new variable to be equivalent to subformula
- Substitution Frege: Recycle any subderivation in single step
- Believed to be exponentially stronger than Frege
- Known to be polynomially equivalent
- Open problem: Does this hold also if we define extension and substitution for weaker proof systems?

Ideal Proof System

- **Very** rough explanation of ideal proof system:
  Nullstellensatz, but represent polynomials as you like
- For instance, with arithmetic circuits
- Yields very strong proof system!
- Conditional results establishing relations with extended Frege and other proof systems
- Unconditional results for restricted arithmetic circuits

Bounded Arithmetic

- Bridge between logic and computational complexity theory
- Weak formal theories of arithmetic
- Peano Arithmetic, but
  - restricted power of induction hypotheses
  - restricted quantifiers
- Designed to capture feasible reasoning
- Correspondence between bounded arithmetic theories and proof systems
- Bounded arithmetic proof can be translated to family of propositional logic proofs

Some Interesting Proof Techniques Worth Closer Study

- Duality
- Reductions
  - via low-depth decision trees
  - via low degree polynomials
- Switching lemmas
- Pseudo-width
- Top-down analysis

Proof Complexity Applications in Computational Complexity Theory

Total NP search problems (TFNP)

- Tight correspondence between TFNP problems and proof systems
- Breakthrough results from proof complexity separations
- And also new proof systems

Proof Complexity Applications in Computational Complexity Theory

Total NP search problems (TFNP)

- Tight correspondence between TFNP problems and proof systems
- Breakthrough results from proof complexity separations
- And also new proof systems

Circuit complexity

- "Moral parallels" between circuit complexity and proof complexity
- But proof complexity gets stuck earlier
- Intriguing interplay between proof complexity, circuit complexity, and communication complexity

Proof Complexity Applications in Computational Complexity Theory

Total NP search problems (TFNP)

- Tight correspondence between TFNP problems and proof systems
- Breakthrough results from proof complexity separations
- And also new proof systems

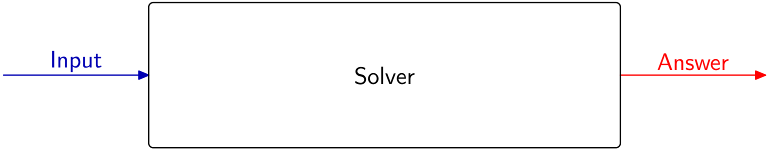
Circuit complexity

- "Moral parallels" between circuit complexity and proof complexity
- But proof complexity gets stuck earlier
- Intriguing interplay between proof complexity, circuit complexity, and communication complexity

Extension complexity

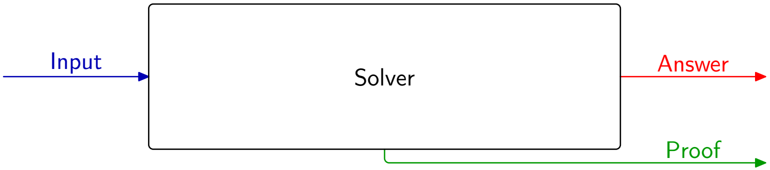
- Impossibility results for LP and SDP formulations
- Lower bounds for Sherali–Adams and sum-of-squares

Proof Complexity for Certified Combinatorial Solving



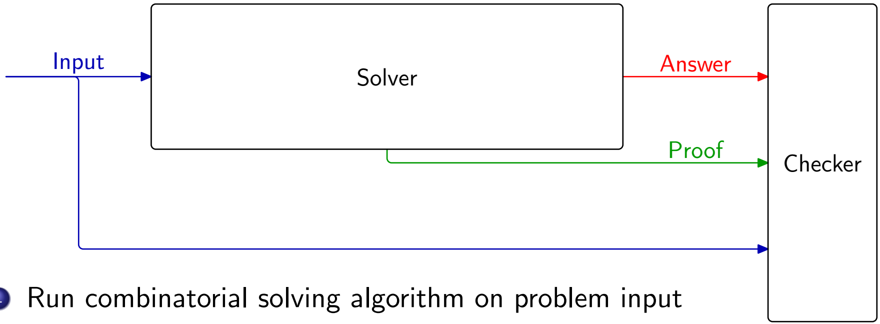
1. Run combinatorial solving algorithm on problem input

Proof Complexity for Certified Combinatorial Solving



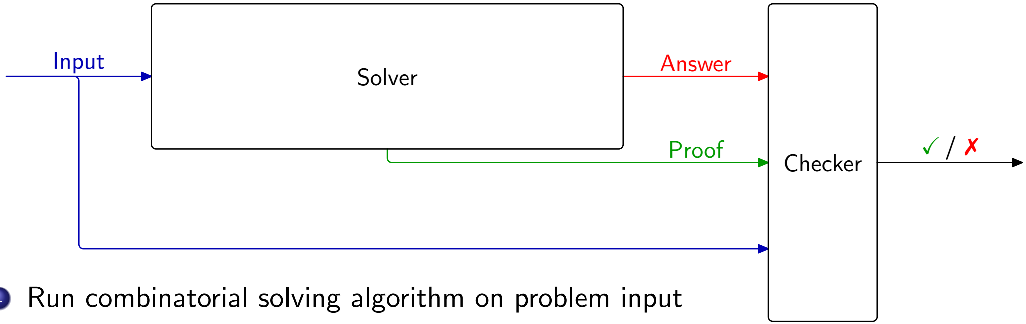
1. Run combinatorial solving algorithm on problem input
2. Get as output not only answer but also proof

Proof Complexity for Certified Combinatorial Solving



1. Run combinatorial solving algorithm on problem input
2. Get as output not only answer but also proof
3. Feed input + answer + proof to proof checker

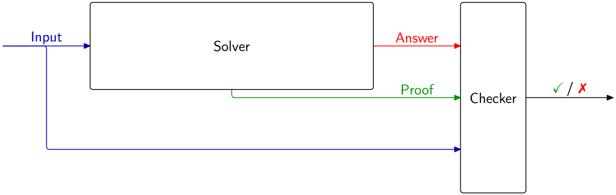
Proof Complexity for Certified Combinatorial Solving



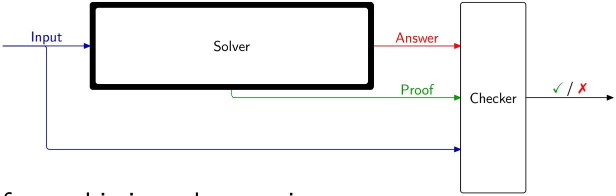
1. Run combinatorial solving algorithm on problem input
2. Get as output not only answer but also proof
3. Feed input + answer + proof to proof checker
4. Verify that proof checker says answer is correct

Proof System Desiderata

Proof format for certifying solver should be



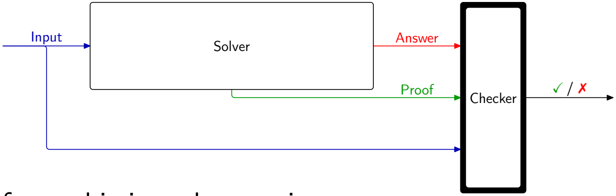
Proof System Desiderata



Proof format for certifying solver should be

- **very powerful:** minimal overhead for sophisticated reasoning

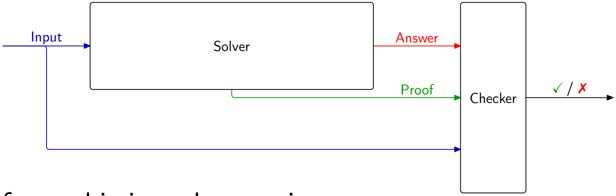
Proof System Desiderata



Proof format for certifying solver should be

- **very powerful:** minimal overhead for sophisticated reasoning
- **dead simple:** checking correctness of proofs should be (almost) trivial

Proof System Desiderata

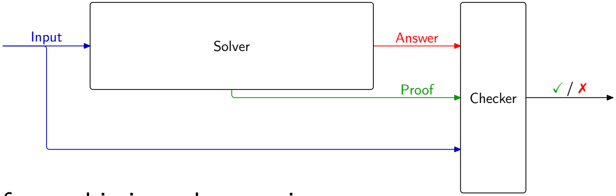


Proof format for certifying solver should be

- **very powerful:** minimal overhead for sophisticated reasoning
- **dead simple:** checking correctness of proofs should be (almost) trivial

Clear conflict expressivity vs. simplicity!

Proof System Desiderata



Proof format for certifying solver should be

- **very powerful:** minimal overhead for sophisticated reasoning
- **dead simple:** checking correctness of proofs should be (almost) trivial

Clear conflict expressivity vs. simplicity!

Interesting problem to try to design suitable proof systems
(also for optimization problems and beyond Boolean format)

Redundance-Based Strengthening

C is redundant with respect to $\mathcal{F}$ if $\mathcal{F}$ and $\mathcal{F} \cup \{C\}$ are equisatisfiable

Want to allow adding such "redundant" constraints

Redundance-Based Strengthening

C is redundant with respect to $\mathcal{F}$ if $\mathcal{F}$ and $\mathcal{F} \cup \{C\}$ are equisatisfiable

Want to allow adding such "redundant" constraints

Redundance-based strengthening ([BT19, GN21], inspired by [JHB12])

C is redundant with respect to $\mathcal{F}$ if and only if there is a substitution ω (mapping variables to truth values or literals), called a witness, for which

$$\mathcal{F} \cup \{\neg C\} \models (\mathcal{F} \cup \{C\})\restriction_{\omega}$$

Redundance-Based Strengthening

C is redundant with respect to $\mathcal{F}$ if $\mathcal{F}$ and $\mathcal{F} \cup \{C\}$ are equisatisfiable

Want to allow adding such "redundant" constraints

Redundance-based strengthening ([BT19, GN21], inspired by [JHB12])

C is redundant with respect to $\mathcal{F}$ if and only if there is a substitution ω (mapping variables to truth values or literals), called a witness, for which

$$\mathcal{F} \cup \{\neg C\} \models (\mathcal{F} \cup \{C\})\restriction_{\omega}$$

- Proof sketch for interesting direction: If α satisfies $\mathcal{F}$ but falsifies C, then $\alpha \circ \omega$ satisfies $\mathcal{F} \cup \{C\}$

Redundance-Based Strengthening

C is redundant with respect to $\mathcal{F}$ if $\mathcal{F}$ and $\mathcal{F} \cup \{C\}$ are equisatisfiable
Want to allow adding such "redundant" constraints

Redundance-based strengthening ([BT19, GN21], inspired by [JHB12])

C is redundant with respect to $\mathcal{F}$ if and only if there is a substitution ω (mapping variables to truth values or literals), called a witness, for which

$$\mathcal{F} \cup \{\neg C\} \models (\mathcal{F} \cup \{C\})\restriction_\omega$$

- Proof sketch for interesting direction: If α satisfies $\mathcal{F}$ but falsifies C, then $\alpha \circ \omega$ satisfies $\mathcal{F} \cup \{C\}$
- In a proof, the implication needs to be efficiently verifiable — every $D \in (\mathcal{F} \cup \{C\})\restriction_\omega$ should follow from $\mathcal{F} \cup \{\neg C\}$ either
  1. "obviously" or
  2. by explicitly presented derivation

Example: Deriving $a \leftrightarrow (x \wedge y)$ Using the Redundance Rule

Want to derive

$$2\overline{a} + x + y \geq 2 \qquad a + \overline{x} + \overline{y} \geq 1$$

using condition $\mathcal{F} \cup \{\neg C\} \models (\mathcal{F} \cup \{C\})\restriction_{\omega}$

Example: Deriving $a \leftrightarrow (x \land y)$ Using the Redundance Rule

Want to derive

$$2\overline{a} + x + y \geq 2 \qquad a + \overline{x} + \overline{y} \geq 1$$

using condition $\mathcal{F} \cup \{\neg C\} \models (\mathcal{F} \cup \{C\})\restriction_{\omega}$

1. $\mathcal{F} \cup \{\neg(2\overline{a} + x + y \geq 2)\} \models (\mathcal{F} \cup \{2\overline{a} + x + y \geq 2\})\restriction_{\omega}$

Example: Deriving $a \leftrightarrow (x \wedge y)$ Using the Redundance Rule

Want to derive

$$2\overline{a} + x + y \geq 2 \qquad a + \overline{x} + \overline{y} \geq 1$$

using condition $\mathcal{F} \cup \{\neg C\} \models (\mathcal{F} \cup \{C\})\restriction_{\omega}$

1. $\mathcal{F} \cup \{\neg(2\overline{a} + x + y \geq 2)\} \models (\mathcal{F} \cup \{2\overline{a} + x + y \geq 2\})\restriction_{\omega}$
   Choose $\omega = \{a \mapsto 0\}$ — $\mathcal{F}$ untouched; new constraint satisfied

Example: Deriving $a \leftrightarrow (x \wedge y)$ Using the Redundance Rule

Want to derive

$$2\overline{a} + x + y \geq 2 \qquad a + \overline{x} + \overline{y} \geq 1$$

using condition $\mathcal{F} \cup \{\neg C\} \models (\mathcal{F} \cup \{C\})\restriction_{\omega}$

1. $\mathcal{F} \cup \{\neg(2\overline{a} + x + y \geq 2)\} \models (\mathcal{F} \cup \{2\overline{a} + x + y \geq 2\})\restriction_{\omega}$
   Choose $\omega = \{a \mapsto 0\}$ — $\mathcal{F}$ untouched; new constraint satisfied

2. $\mathcal{F} \cup \{2\overline{a} + x + y \geq 2, \neg(a + \overline{x} + \overline{y} \geq 1)\} \models (\mathcal{F} \cup \{2\overline{a} + x + y \geq 2, a + \overline{x} + \overline{y} \geq 1\})\restriction_{\omega}$

Example: Deriving $a \leftrightarrow (x \wedge y)$ Using the Redundance Rule

Want to derive

$$2\overline{a} + x + y \geq 2 \qquad a + \overline{x} + \overline{y} \geq 1$$

using condition $\mathcal{F} \cup \{\neg C\} \models (\mathcal{F} \cup \{C\})\restriction_\omega$

1. $\mathcal{F} \cup \{\neg(2\overline{a} + x + y \geq 2)\} \models (\mathcal{F} \cup \{2\overline{a} + x + y \geq 2\})\restriction_\omega$
   Choose $\omega = \{a \mapsto 0\}$ — $\mathcal{F}$ untouched; new constraint satisfied

2. $\mathcal{F} \cup \{2\overline{a} + x + y \geq 2, \neg(a + \overline{x} + \overline{y} \geq 1)\} \models (\mathcal{F} \cup \{2\overline{a} + x + y \geq 2, a + \overline{x} + \overline{y} \geq 1\})\restriction_\omega$
   Choose $\omega = \{a \mapsto 1\}$ — $\mathcal{F}$ untouched; new constraint satisfied
   $\neg(a + \overline{x} + \overline{y} \geq 1)$ forces $x \mapsto 1$ and $y \mapsto 1$, hence $2\overline{a} + x + y \geq 2$ remains satisfied after forcing a to be true

Open Problems: Strength of Restricted Redundance Rules?

Adding redundance rule $\Rightarrow$ proof system polynomially equivalent to extended Frege

Open Problems: Strength of Restricted Redundance Rules?

Adding redundance rule $\Rightarrow$ proof system polynomially equivalent to extended Frege

1. What is the power of the redundance rule if we forbid new variables?
   For resolution + redundance known that:
   - Pigeonhole principle formulas easy
   - Tseitin formulas easy

Open Problems: Strength of Restricted Redundance Rules?

Adding redundance rule $\Rightarrow$ proof system polynomially equivalent to extended Frege

1. What is the power of the redundance rule if we forbid new variables?
   For resolution + redundance known that:
   - Pigeonhole principle formulas easy
   - Tseitin formulas easy
2. What is the power of resolution with redundance if we only allow new variables $z \leftrightarrow C$ for previously derived clauses C?
   - Corresponds (kind of) to reasoning in core-guided MaxSAT solvers

Redundance and Dominance Rules in VeriPB (Slightly Simplified)

Redundance-based strengthening, optimization version with objective f [BGMN23]

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models (\mathcal{F} \cup \mathcal{D} \cup \{C\})\restriction_\omega \cup \{f\restriction_\omega \leq f\}$$

Redundance and Dominance Rules in VeriPB (Slightly Simplified)

Redundance-based strengthening, optimization version with objective f [BGMN23]

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models (\mathcal{F} \cup \mathcal{D} \cup \{C\})\restriction_\omega \cup \{f\restriction_\omega \leq f\}$$

Can be more aggressive if witness ω strictly improves solution

Redundance and Dominance Rules in VeriPB (Slightly Simplified)

Redundance-based strengthening, optimization version with objective f [BGMN23]

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models (\mathcal{F} \cup \mathcal{D} \cup \{C\})\restriction_\omega \cup \{f\restriction_\omega \leq f\}$$

Can be more aggressive if witness ω strictly improves solution

Dominance-based strengthening with objective f [BGMN23]

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Redundance and Dominance Rules in VERIPB (Slightly Simplified)

Redundance-based strengthening, optimization version with objective f [BGMN23]

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models (\mathcal{F} \cup \mathcal{D} \cup \{C\})\restriction_\omega \cup \{f\restriction_\omega \leq f\}$$

Can be more aggressive if witness ω strictly improves solution

Dominance-based strengthening with objective f [BGMN23]

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

- Applying ω should strictly decrease f
- If so, don't need to show that $(\mathcal{D} \cup \{C\})\restriction_\omega$ implied!

Soundness of Dominance Rule

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}{\restriction_\omega} \cup \{f{\restriction_\omega} < f\}$$

Why is this sound? Let $\mathcal{D} = \emptyset$ for simplicity

Soundness of Dominance Rule

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Why is this sound? Let $\mathcal{D} = \emptyset$ for simplicity

1. Suppose α satisfies $\mathcal{F}$ but falsifies C (i.e., satisfies $\neg C$)

Soundness of Dominance Rule

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Why is this sound? Let $\mathcal{D} = \emptyset$ for simplicity

1. Suppose α satisfies $\mathcal{F}$ but falsifies C (i.e., satisfies $\neg C$)
2. Then $\alpha \circ \omega$ satisfies $\mathcal{F}$ and $f(\alpha \circ \omega) < f(\alpha)$

Soundness of Dominance Rule

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Why is this sound? Let $\mathcal{D} = \emptyset$ for simplicity

1. Suppose α satisfies $\mathcal{F}$ but falsifies C (i.e., satisfies $\neg C$)
2. Then $\alpha \circ \omega$ satisfies $\mathcal{F}$ and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies C, we're done

Soundness of Dominance Rule

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Why is this sound? Let $\mathcal{D} = \emptyset$ for simplicity

1. Suppose α satisfies $\mathcal{F}$ but falsifies C (i.e., satisfies $\neg C$)
2. Then $\alpha \circ \omega$ satisfies $\mathcal{F}$ and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies C, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies $\mathcal{F}$ and $f\big((\alpha \circ \omega) \circ \omega\big) < f\big(\alpha \circ \omega\big)$

Soundness of Dominance Rule

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Why is this sound? Let $\mathcal{D} = \emptyset$ for simplicity

1. Suppose α satisfies $\mathcal{F}$ but falsifies C (i.e., satisfies $\neg C$)
2. Then $\alpha \circ \omega$ satisfies $\mathcal{F}$ and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies C, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies $\mathcal{F}$ and $f\big((\alpha \circ \omega) \circ \omega\big) < f\big(\alpha \circ \omega\big)$
5. If $(\alpha \circ \omega) \circ \omega$ satisfies C, we're done

Soundness of Dominance Rule

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}{\restriction_\omega} \cup \{f{\restriction_\omega} < f\}$$

Why is this sound? Let $\mathcal{D} = \emptyset$ for simplicity

1. Suppose α satisfies $\mathcal{F}$ but falsifies C (i.e., satisfies $\neg C$)
2. Then $\alpha \circ \omega$ satisfies $\mathcal{F}$ and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies C, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies $\mathcal{F}$ and $f((\alpha \circ \omega) \circ \omega) < f(\alpha \circ \omega)$
5. If $(\alpha \circ \omega) \circ \omega$ satisfies C, we're done
6. Otherwise $((\alpha \circ \omega) \circ \omega) \circ \omega$ satisfies $\mathcal{F}$ and $f(((\alpha \circ \omega) \circ \omega) \circ \omega) < f((\alpha \circ \omega) \circ \omega)$

Soundness of Dominance Rule

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Why is this sound? Let $\mathcal{D} = \emptyset$ for simplicity

1. Suppose α satisfies $\mathcal{F}$ but falsifies C (i.e., satisfies $\neg C$)
2. Then $\alpha \circ \omega$ satisfies $\mathcal{F}$ and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies C, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies $\mathcal{F}$ and $f((\alpha \circ \omega) \circ \omega) < f(\alpha \circ \omega)$
5. If $(\alpha \circ \omega) \circ \omega$ satisfies C, we're done
6. Otherwise $((\alpha \circ \omega) \circ \omega) \circ \omega$ satisfies $\mathcal{F}$ and $f(((\alpha \circ \omega) \circ \omega) \circ \omega) < f((\alpha \circ \omega) \circ \omega)$
7. ...

Soundness of Dominance Rule

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Why is this sound? Let $\mathcal{D} = \emptyset$ for simplicity

1. Suppose α satisfies $\mathcal{F}$ but falsifies C (i.e., satisfies $\neg C$)
2. Then $\alpha \circ \omega$ satisfies $\mathcal{F}$ and $f(\alpha \circ \omega) < f(\alpha)$
3. If $\alpha \circ \omega$ satisfies C, we're done
4. Otherwise $(\alpha \circ \omega) \circ \omega$ satisfies $\mathcal{F}$ and $f\big((\alpha \circ \omega) \circ \omega\big) < f\big(\alpha \circ \omega\big)$
5. If $(\alpha \circ \omega) \circ \omega$ satisfies C, we're done
6. Otherwise $\big((\alpha \circ \omega) \circ \omega\big) \circ \omega$ satisfies $\mathcal{F}$ and $f\big(((\alpha \circ \omega) \circ \omega) \circ \omega\big) < f\big((\alpha \circ \omega) \circ \omega\big)$
7. ...
8. Can't go on forever, so finally reach α' satisfying $\mathcal{F} \cup \{C\}$

Soundness of Dominance Rule (Continued)

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}{\restriction_\omega} \cup \{f{\restriction_\omega} < f\}$$

Soundness of Dominance Rule (Continued)

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Suppose now that $\mathcal{D} \neq \emptyset$

- Same inductive proof as before, but also nested forward induction over derivation
- Or pick α satisfying $\mathcal{F} \cup \mathcal{D}$ and minimizing f and argue by contradiction

Soundness of Dominance Rule (Continued)

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Suppose now that $\mathcal{D} \neq \emptyset$

- Same inductive proof as before, but also nested forward induction over derivation
- Or pick α satisfying $\mathcal{F} \cup \mathcal{D}$ and minimizing f and argue by contradiction

Further extensions:

- Define dominance rule with respect to order $\mathcal{O}$ independent of objective function
- Switch between different orders in same proof

Soundness of Dominance Rule (Continued)

Dominance-based strengthening

Add constraint C to derived set $\mathcal{D}$ if exists witness substitution ω such that

$$\mathcal{F} \cup \mathcal{D} \cup \{\neg C\} \models \mathcal{F}\restriction_\omega \cup \{f\restriction_\omega < f\}$$

Suppose now that $\mathcal{D} \neq \emptyset$

- Same inductive proof as before, but also nested forward induction over derivation
- Or pick α satisfying $\mathcal{F} \cup \mathcal{D}$ and minimizing f and argue by contradiction

Further extensions:

- Define dominance rule with respect to order $\mathcal{O}$ independent of objective function
- Switch between different orders in same proof

Yields proof system that is probably stronger than extended Frege [KT24]

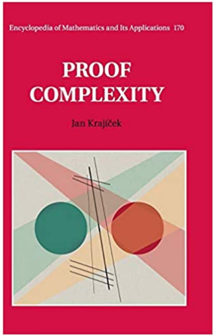
Symmetry-Aware Proof Systems

- With dominance rule, can support fully general symmetry breaking
  - Invent "objective function" that minimizes lexicographic order of satisfying assignment
  - Allows adding lex order constraints forbidding other solutions
  - Other approaches also possible (but beyond the scope of this discussion)
- Modern symmetry handling tools can solve many symmetric hard proof complexity formulas even during preprocessing
- But non-symmetric formulas are presumably still hard?
- Desirable to have lower bounds that remain valid also in the presence of state-of-the-art symmetry handling tools
  - Define "symmetry-aware" versions of resolution, polynomial calculus, cutting planes, ...
  - Develop techniques to prove lower bounds

Handbook of Satisfiability
(Especially chapter 7 ☺)

[BHvMW21]

Proof Complexity
by Jan Krajíček



[Kra19]

Summing up This Course

We focused on some proof systems corresponding to combinatorial solving algorithms:

- Resolution $\longleftrightarrow$ conflict-driven clause learning (CDCL)
- Nullstellensatz and polynomial calculus $\longleftrightarrow$ Gröbner bases
- Cutting planes $\longleftrightarrow$ pseudo-Boolean solving

Summing up This Course

We focused on some proof systems corresponding to combinatorial solving algorithms:

- Resolution $\longleftrightarrow$ conflict-driven clause learning (CDCL)
- Nullstellensatz and polynomial calculus $\longleftrightarrow$ Gröbner bases
- Cutting planes $\longleftrightarrow$ pseudo-Boolean solving

We mentioned but didn't really go into any details about:

- Sherali–Adams and sums of squares $\longleftrightarrow$ LP and SDP hierarchies
- Stabbing planes $\longleftrightarrow$ integer linear programming
- Extended resolution $\longleftrightarrow$ SAT pre- and inprocessing
- ... and other stronger proof systems

Summing up This Course

We focused on some proof systems corresponding to combinatorial solving algorithms:

- Resolution $\longleftrightarrow$ conflict-driven clause learning (CDCL)
- Nullstellensatz and polynomial calculus $\longleftrightarrow$ Gröbner bases
- Cutting planes $\longleftrightarrow$ pseudo-Boolean solving

We mentioned but didn't really go into any details about:

- Sherali–Adams and sums of squares $\longleftrightarrow$ LP and SDP hierarchies
- Stabbing planes $\longleftrightarrow$ integer linear programming
- Extended resolution $\longleftrightarrow$ SAT pre- and inprocessing
- ... and other stronger proof systems

Main motivation for proof complexity as a computational lens:

- Analyse state-of-the-art algorithms (and provide methods for certifying correctness!)
- Give ideas for new approaches
- Provide a fun playground for theory-practice interaction!

Summing up This Course

We focused on some proof systems corresponding to combinatorial solving algorithms:

- Resolution $\longleftrightarrow$ conflict-driven clause learning (CDCL)
- Nullstellensatz and polynomial calculus $\longleftrightarrow$ Gröbner bases
- Cutting planes $\longleftrightarrow$ pseudo-Boolean solving

We mentioned but didn't really go into any details about:

- Sherali–Adams and sums of squares $\longleftrightarrow$ LP and SDP hierarchies
- Stabbing planes $\longleftrightarrow$ integer linear programming
- Extended resolution $\longleftrightarrow$ SAT pre- and inprocessing
- ... and other stronger proof systems

Main motivation for proof complexity as a computational lens:

- Analyse state-of-the-art algorithms (and provide methods for certifying correctness!)
- Give ideas for new approaches
- Provide a fun playground for theory-practice interaction!

Thank you for attending this course!

References I

[AH19] Albert Atserias and Tuomas Hakoniemi. Size-degree trade-offs for Sums-of-Squares and Positivstellensatz proofs. In *Proceedings of the 34th Annual Computational Complexity Conference (CCC '19)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:20, July 2019.

[Ber18] Christoph Berkholz. The relation between polynomial calculus, Sherali-Adams, and sum-of-squares proofs. In *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science (STACS '18)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:14, February 2018.

[BFI+18] Paul Beame, Noah Fleming, Russell Impagliazzo, Antonina Kolokolova, Denis Pankratov, Toniann Pitassi, and Robert Robere. Stabbing planes. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference (ITCS '18)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:20, January 2018.

[BGMN23] Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström. Certified dominance and symmetry breaking for combinatorial optimisation. *Journal of Artificial Intelligence Research*, 77:1539–1589, August 2023. Preliminary version in *AAAI '22*.

References II

[BHvMW21] Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2nd edition, February 2021.

[BN21] Samuel R. Buss and Jakob Nordström. Proof complexity and SAT solving. In Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, chapter 7, pages 233–350. IOS Press, 2nd edition, February 2021. Available at `http://www.jakobnordstrom.se/publications/`.

[BT19] Samuel R. Buss and Neil Thapen. DRAT proofs, propagation redundancy, and extended resolution. In *Proceedings of the 22nd International Conference on Theory and Applications of Satisfiability Testing (SAT '19)*, volume 11628 of *Lecture Notes in Computer Science*, pages 71–89. Springer, July 2019.

[BW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.

[dRPR23] Susanna F. de Rezende, Aaron Potechin, and Kilian Risse. Clique is hard on average for unary Sherali–Adams. In *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS '23)*, pages 12–25, November 2023.

References III

[DT20] Daniel Dadush and Samarth Tiwari. On the complexity of branching proofs. In *Proceedings of the 35th Annual Computational Complexity Conference (CCC '20)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 34:1–34:35, July 2020.

[FGI⁺21] Noah Fleming, Mika Göös, Russell Impagliazzo, Toniann Pitassi, Robert Robere, Li-Yang Tan, and Avi Wigderson. On the power and limitations of branch and cut. In *Proceedings of the 36th Annual Computational Complexity Conference (CCC '21)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:30, July 2021.

[FKP19] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends in Theoretical Computer Science*, 14(1–2):1–221, December 2019.

[GHJ⁺24] Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Separations in proof complexity and tfnp. *Journal of the ACM*, 71(4):26:1–26:45, August 2024. Preliminary version in *FOCS '02*.

[GN21] Stephan Gocht and Jakob Nordström. Certifying parity reasoning efficiently using pseudo-Boolean proofs. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI '21)*, pages 3768–3777, February 2021.

References IV

[GP24] Max Gläser and Marc E. Pfetsch. Sub-exponential lower bounds for branch-and-bound with general disjunctions via interpolation. In *Proceedings of the 35th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '24)*, pages 3747–3764, January 2024.

[IPS99] Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[JHB12] Matti Järvisalo, Marijn J. H. Heule, and Armin Biere. Inprocessing rules. In *Proceedings of the 6th International Joint Conference on Automated Reasoning (IJCAR '12)*, volume 7364 of *Lecture Notes in Computer Science*, pages 355–370. Springer, June 2012.

[Kra19] Jan Krajíček. *Proof Complexity*, volume 170 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, March 2019.

[KT24] Leszek Kołodziejczyk and Neil Thapen. The strength of the dominance rule. In *Proceedings of the 27th International Conference on Theory and Applications of Satisfiability Testing (SAT '24)*, volume 305 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:22, August 2024.

[PX25] Aaron Potechin and Jeff Xu. Sum-of-squares lower bounds for coloring random graphs. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC '25)*, pages 84–95, June 2025.