The interpolation technique in proof complexity

Pavel Hrubeš

University of Washington

1. Proof complexity and its goals

Mathematical logic:
given a proof system P, which formulas are *provable* in P?

Mathematical logic:
given a proof system P, which formulas are *provable* in P?

Proof complexity:
given a proof system P, which formulas have *short proofs* in P?

Propositional proof system:

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a “proof” is a proof in P.

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

Examples:

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

Examples:

1. Frege system.

Propositional proof system:

1. a formula *A* has a proof in *P* iff *A* is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in *P*.

Examples:

1. Frege system.

- The textbook axiomatization of propositional calculus.

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

Examples:

1. Frege system.

- The textbook axiomatization of propositional calculus.
- Axioms such as

$$A \to A \vee B\,,\ A \to (B \to A)\,,\ldots$$

from which we are supposed to derive a formula by means of the *modus ponens* rule

$$\frac{A\,,\ A \to B}{B}\,.$$

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

Examples:

1. Frege system.

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

Examples:

1. Frege system.
2. Resolution.

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

Examples:

1. Frege system.
2. Resolution.
3. Cutting Planes.

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

Examples:

1. Frege system.
2. Resolution.
3. Cutting Planes.
4. ZFC.

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a “proof” is a proof in P.

P is *polynomially bounded* if there exists a polynomial q s.t. every tautology A of size s has a proof in P of size $q(s)$.

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

P is *polynomially bounded* if there exists a polynomial q s.t. every tautology A of size s has a proof in P of size $q(s)$.

Proof complexity:
given a propositional proof system P, is P polynomially bounded?

Propositional proof system:

1. a formula A has a proof in P iff A is a propositional tautology;
2. it can be checked in polynomial time whether a "proof" is a proof in P.

P is *polynomially bounded* if there exists a polynomial q s.t. every tautology A of size s has a proof in P of size $q(s)$.

Proof complexity:
given a propositional proof system P, is P polynomially bounded?

Theorem (Cook-Reckov)

There exists a polynomially bounded propositional proof system iff $NP = coNP$.

- We know that Resolution, Cutting Planes and many other systems are not polynomially bound.

- We know that Resolution, Cutting Planes and many other systems are not polynomially bound.
- There are *many more* systems for which this is not known.

- We know that Resolution, Cutting Planes and many other systems are not polynomially bound.
- There are *many more* systems for which this is not known.

Main open problem: Is the Frege system polynomially bounded?

- We know that Resolution, Cutting Planes and many other systems are not polynomially bound.
- There are *many more* systems for which this is not known.

Main open problem: Is the Frege system polynomially bounded?

- Design techniques for proving lower bounds on sizes of proofs.

- We know that Resolution, Cutting Planes and many other systems are not polynomially bound.
- There are *many more* systems for which this is not known.

Main open problem: Is the Frege system polynomially bounded?

- Design techniques for proving lower bounds on sizes of proofs.
- Feasible interpolation is one such technique.

2. Some facts about Boolean functions

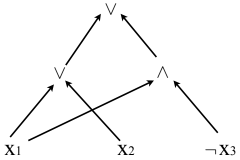
Boolean function - $f : \{0,1\}^n \to \{0,1\}$.

Boolean circuit - directed acyclic graph

- Inputs: in-degree zero

$$x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n.$$

- Operations: $\wedge, \vee$ with in-degree two.



Size - number of nodes/gates.

Circuit size of *f* - the size of a smallest circuit computing f.

Monotone Boolean function

- for $x, y \in \{0,1\}^n$, write $x \leq y$ if $x_i \leq y_i$ for every $i \in \{1, \ldots, n\}$.

Monotone Boolean function

- for $x, y \in \{0,1\}^n$, write $x \leq y$ if $x_i \leq y_i$ for every $i \in \{1, \ldots, n\}$.
- $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if for every $x, y \in \{0,1\}^n$ $x \leq y$ implies $f(x) \leq f(y)$.

Monotone Boolean function

- for $x, y \in \{0,1\}^n$, write $x \leq y$ if $x_i \leq y_i$ for every $i \in \{1, \ldots, n\}$.
- $f : \{0,1\}^n \rightarrow \{0,1\}$ is *monotone* if for every $x, y \in \{0,1\}^n$ $x \leq y$ implies $f(x) \leq f(y)$.

Examples:

Monotone Boolean function

- for $x, y \in \{0,1\}^n$, write $x \leq y$ if $x_i \leq y_i$ for every $i \in \{1, \ldots, n\}$.
- $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if for every $x, y \in \{0,1\}^n$ $x \leq y$ implies $f(x) \leq f(y)$.

Examples:

- conjunction, disjunction, the majority function.

Monotone Boolean function

- for $x, y \in \{0,1\}^n$, write $x \leq y$ if $x_i \leq y_i$ for every $i \in \{1, \ldots, n\}$.
- $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if for every $x, y \in \{0,1\}^n$ $x \leq y$ implies $f(x) \leq f(y)$.

Examples:

- conjunction, disjunction, the majority function.
- The k-clique function CL_n^k:
  n^2 inputs $\{x_{i,j}; i, j \in [n]\}$ represent edges of a graph on n vertices.

$$\mathrm{CL}_n^k = 1$$

iff the graph has a clique of size k.

Monotone Boolean function

- for $x, y \in \{0, 1\}^n$ write $x \leq y$ if $x_i \leq y_i$ for every $i \in \{1, \ldots, n\}$.
- $f : \{0, 1\}^n \to \{0, 1\}$ is *monotone* if for every $x, y \in \{0, 1\}^n$ $x \leq y$ implies $f(x) \leq f(y)$.

Monotone Boolean function

- for $x, y \in \{0,1\}^n$ write $x \leq y$ if $x_i \leq y_i$ for every $i \in \{1, \ldots, n\}$.
- $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if for every $x, y \in \{0,1\}^n$ $x \leq y$ implies $f(x) \leq f(y)$.

Monotone boolean circuit is a boolean circuit which doesn't contain negations.

Monotone Boolean function

- for $x, y \in \{0,1\}^n$ write $x \leq y$ if $x_i \leq y_i$ for every $i \in \{1, \ldots, n\}$.
- $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if for every $x, y \in \{0,1\}^n$ $x \leq y$ implies $f(x) \leq f(y)$.

Monotone boolean circuit is a boolean circuit which doesn't contain negations.

- Every monotone function can be computed by a monotone circuit.

Monotone Boolean function

- for $x, y \in \{0,1\}^n$ write $x \leq y$ if $x_i \leq y_i$ for every $i \in \{1, \ldots, n\}$.
- $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if for every $x, y \in \{0,1\}^n$ $x \leq y$ implies $f(x) \leq f(y)$.

Monotone boolean circuit is a boolean circuit which doesn't contain negations.

- Every monotone function can be computed by a monotone circuit.

Theorem (R,AB)

Set $k := \lceil \sqrt{n} \rceil$. Every monotone circuit computing the clique function CL_n^k must have size at least $2^{\Omega(n^{1/4})}$.

3. Craig's interpolation theorem

x, y, z - disjoint sets of variables.

Craig's interpolation theorem

Assume that $A(x, y) \to B(x, z)$ is a tautology.
Then there exists a formula $C(x)$ such that both
$A(x, y) \to C(x)$, $C(x) \to B(x, y)$ are tautologies.

x, y, z - disjoint sets of variables.

Craig's interpolation theorem

Assume that $A(x, y) \rightarrow B(x, z)$ *is a tautology.*
Then there exists a formula $C(x)$ *such that both*
$A(x, y) \rightarrow C(x)\,, \;\; C(x) \rightarrow B(x, y)$ *are tautologies.*

C is an *interpolant* of A and B.

Craig's interpolation theorem

Assume that $A(x, y) \to B(x, z)$ *is a tautology.*
Then there exists a formula $C(x)$ *such that both*
$A(x, y) \to C(x)\,,\ \ C(x) \to B(x, z)$ *are tautologies.*

Craig's interpolation theorem

Assume that $A(x,y) \to B(x,z)$ *is a tautology.*
Then there exists a formula $C(x)$ *such that both*
$A(x,y) \to C(x)\,,\;\; C(x) \to B(x,z)$ *are tautologies.*

Proof.

Craig's interpolation theorem

Assume that $A(x, y) \to B(x, z)$ *is a tautology.*
Then there exists a formula $C(x)$ *such that both*
$A(x, y) \to C(x)$, $C(x) \to B(x, z)$ *are tautologies.*

Proof.

Define

$$C(x) := \bigvee_{\sigma \in \{0,1\}^k} A(x, \sigma), \text{ where } k = |y|.$$

□

Craig's interpolation theorem

Assume that $A(x, y) \rightarrow B(x, z)$ is a tautology.
Then there exists a formula $C(x)$ such that both
$A(x, y) \rightarrow C(x)$, $C(x) \rightarrow B(x, z)$ are tautologies.

Proof.

Define

$$C(x) := \bigvee_{\sigma \in \{0,1\}^k} A(x, \sigma), \text{ where } k = |y|.$$

That is $C(x) = 1$ iff there exists $\sigma \in \{0, 1\}^k$ with $A(x, \sigma) = 1$.

□

Craig's interpolation theorem
Assume that $A(x, y) \to B(x, z)$ is a tautology.
Then there exists a formula $C(x)$ such that both
$A(x, y) \to C(x)\,,\ \ C(x) \to B(x, z)$ are tautologies.

Proof.
Define

$$C(x) := \bigvee_{\sigma \in \{0,1\}^k} A(x, \sigma)\,, \text{ where } k = |y|.$$

That is $C(x) = 1$ iff there exists $\sigma \in \{0, 1\}^k$ with $A(x, \sigma) = 1$.
Or dually,

$$C'(x) := \bigwedge_{\sigma \in \{0,1\}^s} B(x, \sigma)\,, \text{ where } s = |z|.$$

□

Craig's interpolation theorem

Assume that $A(x, y) \to B(x, z)$ *is a tautology.*
Then there exists a formula $C(x)$ *such that both*
$A(x, y) \to C(x)\,,\;\; C(x) \to B(x, z)$ *are tautologies.*

A symmetric version: if $A'(x, y) \vee B'(x, z)$ is a tautology then there exists $C(x)$ such that

$$C(x) \to A'(x, y)\,,\; \neg C(x) \to B'(x, z)$$

are tautologies.

Craig's interpolation theorem

Assume that $A(x, y) \rightarrow B(x, z)$ *is a tautology.*
Then there exists a formula $C(x)$ *such that both*
$A(x, y) \rightarrow C(x)\,,\;\; C(x) \rightarrow B(x, z)$ *are tautologies.*

Craig's interpolation theorem

Assume that $A(x, y) \to B(x, z)$ *is a tautology.*
Then there exists a formula $C(x)$ *such that both*
$A(x, y) \to C(x)\,,\;\; C(x) \to B(x, z)$ *are tautologies.*

A formula in connectives $\wedge, \vee, \neg$ is *monotone in* x if the variables in x do not occur in the scope of any negation.

Craig's interpolation theorem
Assume that $A(x, y) \to B(x, z)$ is a tautology. Then there exists a formula $C(x)$ such that both $A(x, y) \to C(x)\,,\ \ C(x) \to B(x, z)$ are tautologies.

A formula in connectives $\wedge, \vee, \neg$ is *monotone in x* if the variables in x do not occur in the scope of any negation.

Monotone interpolation theorem *If $A(x, y)$ is monotone in x then there exists an interpolant $C(x)$ which is monotone.*

Example: clique versus coloring

$x = x_{i_1,i_2}, i_1, i_2 \in [n]$ - represent a graph G on vertices $[n]$.

Example: clique versus coloring

$x = x_{i_1,i_2}, i_1, i_2 \in [n]$ - represent a graph G on vertices $[n]$.

$\text{Clique}_n^k(x, y)$ - a CNF formula asserting that y defines a clique of size k in G.

Example: clique versus coloring

$x = x_{i_1,i_2}, i_1, i_2 \in [n]$ - represent a graph G on vertices $[n]$.

$\text{Clique}_n^k(x, y)$ - a CNF formula asserting that y defines a clique of size k in G.

$y = y_{j,i}, j \in [k], i \in [n]$.
$\text{Clique}_n^k(x, y)$ is the conjunction of the following:

1. $\bigvee_{i \in [n]} y_{j,i}$, for every $j \in [k]$,
2. $\neg y_{j_1,i} \vee \neg y_{j_2,i}$, for every $j_1 \neq j_2 \in [k]$, $i \in [n]$,
3. $\neg y_{j_1,i_1} \vee \neg y_{j_2,i_2} \vee x_{i_1,i_2}$, for every $j_1, j_2 \in [k], i_1, i_2 \in [n]$.

Example: clique versus coloring

$x = x_{i_1,i_2}, i_1, i_2 \in [n]$ - represent a graph G on vertices $[n]$.

$\text{Clique}_n^k(x, y)$ - a CNF formula asserting that y defines a clique of size k in G.

Example: clique versus coloring

$x = x_{i_1,i_2}, i_1, i_2 \in [n]$ - represent a graph G on vertices $[n]$.

$\text{Clique}_n^k(x, y)$ - a CNF formula asserting that y defines a clique of size k in G.

$\text{Color}_n^k(x, z)$ - a CNF formula asserting that z defines a k-coloring of G.

Example: clique versus coloring

$x = x_{i_1,i_2}, i_1, i_2 \in [n]$ - represent a graph G on vertices $[n]$.

$\text{Clique}_n^k(x, y)$ - a CNF formula asserting that y defines a clique of size k in G.

$\text{Color}_n^k(x, z)$ - a CNF formula asserting that z defines a k-coloring of G.

Then:

$$\text{Clique}_n^{k+1}(x, y) \rightarrow \neg\text{Color}_n^k(x, z)$$

is a tautology.

Example: clique versus coloring

$$\text{Clique}_n^{k+1}(x, y) \rightarrow \neg\text{Color}_n^k(x, z) \, .$$

An interpolant $C(x)$: tautologies

$$\text{Clique}_n^{k+1}(x, y) \rightarrow C(x) \, , \quad \text{and } C(x) \rightarrow \neg\text{Color}_n^k(x, z)$$

Example: clique versus coloring

$$\text{Clique}_n^{k+1}(x,y) \to \neg\text{Color}_n^k(x,z)\,.$$

An interpolant $C(x)$: tautologies

$$\text{Clique}_n^{k+1}(x,y) \to C(x)\,, \quad \text{and } C(x) \to \neg\text{Color}_n^k(x,z)$$

I.e., C distinguishes between k-colorable graphs and graphs with $k+1$-clique:

Example: clique versus coloring

$$\text{Clique}_n^{k+1}(x, y) \to \neg\text{Color}_n^k(x, z)\,.$$

An interpolant $C(x)$: tautologies

$$\text{Clique}_n^{k+1}(x, y) \to C(x)\,, \quad \text{and } C(x) \to \neg\text{Color}_n^k(x, z)$$

I.e., C distinguishes between k-colorable graphs and graphs with $k+1$-clique:

1. if x has $k+1$-clique then $C(x) = 1$,
2. if x is k-colorable then $C(x) = 0$.

Example: clique versus coloring

$$\mathrm{Clique}_n^{k+1}(x, y) \to \neg\mathrm{Color}_n^k(x, z)\,.$$

An interpolant $C(x)$: tautologies

$$\mathrm{Clique}_n^{k+1}(x, y) \to C(x)\,, \quad \text{and } C(x) \to \neg\mathrm{Color}_n^k(x, z)$$

Example: clique versus coloring

$$\text{Clique}_n^{k+1}(x,y) \to \neg\text{Color}_n^k(x,z)\,.$$

An interpolant $C(x)$: tautologies

$$\text{Clique}_n^{k+1}(x,y) \to C(x)\,, \quad \text{and } C(x) \to \neg\text{Color}_n^k(x,z)$$

Theorem (R, AB)

Let $k := \lceil\sqrt{n}\rceil$. Every monotone interpolant of $\text{Clique}_n^{k+1}(x,y)$ and $\neg\text{Color}_n^k(x,z)$ requires monotone circuit of size at least $2^{\Omega(n^{1/4})}$.

Example: clique versus coloring

$$\text{Clique}_n^{k+1}(x,y) \to \neg\text{Color}_n^k(x,z)\,.$$

An interpolant $C(x)$: tautologies

$$\text{Clique}_n^{k+1}(x,y) \to C(x)\,, \quad \text{and } C(x) \to \neg\text{Color}_n^k(x,z)$$

Theorem (R, AB)

Let $k := \lceil\sqrt{n}\rceil$. Every monotone interpolant of $\text{Clique}_n^{k+1}(x,y)$ and $\neg\text{Color}_n^k(x,z)$ requires monotone circuit of size at least $2^{\Omega(n^{1/4})}$.

- In contrast, there exists an interpolant which can be computed a polynomial size *non-monotone* circuit [L].

4. Feasible interpolation

A propositional proof system P has feasible interpolation, if there exists a polynomial q such that: for every implication $A(x, y) \rightarrow B(x, z)$, if it has a proof in P of size s then $A(x, y)$ and $B(x, z)$ have an interpolant of circuit size $\leq q(s)$.

A propositional proof system P has feasible interpolation, if there exists a polynomial q such that: for every implication $A(x, y) \rightarrow B(x, z)$, if it has a proof in P of size s then $A(x, y)$ and $B(x, z)$ have an interpolant of circuit size $\leq q(s)$.

P has monotone feasible interpolation, if whenever $A(x, y)$ is *monotone* in x then $A(x, y)$ and $B(x, z)$ have an interpolant of *monotone* circuit size $\leq q(s)$.

A propositional proof system P has feasible interpolation, if there exists a polynomial q such that: for every implication $A(x,y) \rightarrow B(x,z)$, if it has a proof in P of size s then $A(x,y)$ and $B(x,z)$ have an interpolant of circuit size $\leq q(s)$.

P has monotone feasible interpolation, if whenever $A(x,y)$ is *monotone* in x then $A(x,y)$ and $B(x,z)$ have an interpolant of *monotone* circuit size $\leq q(s)$.

Theorem

1. *If P has monotone feasible interpolation then the clique vs. coloring tautologies require exponential size proof in P. Hence P is not polynomially bounded.*

A propositional proof system P has feasible interpolation, if there exists a polynomial q such that: for every implication $A(x, y) \rightarrow B(x, z)$, if it has a proof in P of size s then $A(x, y)$ and $B(x, z)$ have an interpolant of circuit size $\leq q(s)$.

P has monotone feasible interpolation, if whenever $A(x, y)$ is *monotone* in x then $A(x, y)$ and $B(x, z)$ have an interpolant of *monotone* circuit size $\leq q(s)$.

Theorem

1. *If P has monotone feasible interpolation then the clique vs. coloring tautologies require exponential size proof in P. Hence P is not polynomially bounded.*
2. *If P has feasible interpolation, then it is not polynomially bounded assuming a conjecture in cryptography.*

5. Feasible interpolation for Resolution

Resolution

Resolution

- a *clause* is a disjunction of variables or their negations.
  $x_1 \vee x_2 \vee \neg x_3$ – identified with the set $\{x_1, x_2, \neg x_3\}$.

Resolution

- a *clause* is a disjunction of variables or their negations. $x_1 \vee x_2 \vee \neg x_3$ – identified with the set $\{x_1, x_2, \neg x_3\}$.
- a *resolution rule* is the rule

$$\frac{C \cup \{x\},\ D \cup \{\neg x\}}{C \cup D}.$$

Resolution

- a *clause* is a disjunction of variables or their negations. $x_1 \vee x_2 \vee \neg x_3$ – identified with the set $\{x_1, x_2, \neg x_3\}$.
- a *resolution rule* is the rule

$$\frac{C \cup \{x\}, \;\; D \cup \{\neg x\}}{C \cup D}.$$

A resolution refutation of a set of clauses $\mathcal{C}$ is a sequence of clauses $D_1, \ldots, D_s$ such that

Resolution

- a *clause* is a disjunction of variables or their negations. $x_1 \vee x_2 \vee \neg x_3$ – identified with the set $\{x_1, x_2, \neg x_3\}$.
- a *resolution rule* is the rule

$$\frac{C \cup \{x\},\ \ D \cup \{\neg x\}}{C \cup D}\ .$$

A resolution refutation of a set of clauses $\mathcal{C}$ is a sequence of clauses $D_1, \ldots, D_s$ such that

1. every D_i is a clause in $\mathcal{C}$, or it has been obtained from some D_{i_1}, D_{i_2}, $i_1, i_2 < i$, by the resolution rule.

Resolution

- a *clause* is a disjunction of variables or their negations. $x_1 \vee x_2 \vee \neg x_3$ – identified with the set $\{x_1, x_2, \neg x_3\}$.
- a *resolution rule* is the rule

$$\frac{C \cup \{x\},\ \ D \cup \{\neg x\}}{C \cup D}\ .$$

A resolution refutation of a set of clauses $\mathcal{C}$ is a sequence of clauses $D_1, \ldots, D_s$ such that

1. every D_i is a clause in $\mathcal{C}$, or it has been obtained from some D_{i_1}, D_{i_2}, $i_1, i_2 < i$, by the resolution rule.
2. D_s is the empty clause $\emptyset$.

Resolution

- a *clause* is a disjunction of variables or their negations. $x_1 \vee x_2 \vee \neg x_3$ – identified with the set $\{x_1, x_2, \neg x_3\}$.
- a *resolution rule* is the rule

$$\frac{C \cup \{x\}\,,\ \ D \cup \{\neg x\}}{C \cup D}\,.$$

A resolution refutation of a set of clauses $\mathcal{C}$ is a sequence of clauses $D_1, \ldots, D_s$ such that

1. every D_i is a clause in $\mathcal{C}$, or it has been obtained from some D_{i_1}, D_{i_2}, $i_1, i_2 < i$, by the resolution rule.
2. D_s is the empty clause $\emptyset$.

If $A = C_1 \wedge \cdots \wedge C_m$ is a CNF formula, a resolution refutation of A is a refutation of $C_1, \ldots, C_m$.

Resolution

Proposition

Let A be CNF formula. The following are equivalent:

1. *A has a resolution refutation.*
2. *A is not satisfiable (= $\neg A$ is a tautology).*

Theorem (Krajíček)

Resolution has both feasible interpolation and monotone feasible interpolation.

Theorem (Krajíček)

Resolution has both feasible interpolation and monotone feasible interpolation.

In words, assume that $A(x, y) \wedge B(x, z)$ has a resolution refutation of size s.

Then there exists a circuit $C(x)$ of size $O(s)$ such that for every assignment σ to the variables x

- if $C(\sigma) = 0$ then $A(\sigma, y)$ is unsatisfiable,
- if $C(\sigma) = 1$ then $B(\sigma, z)$ is unsatisfiable.

Moreover, if A is monotone in x then C can be taken monotone.

Theorem (Krajíček)

Resolution has both feasible interpolation and monotone feasible interpolation.

Corollary

For $k = \lceil \sqrt{n} \rceil$*, every resolution refutation of* $Clique_n^{k+1} \wedge Color_n^k$ *has size at least* $2^{\Omega(n^{1/4})}$*.*

Proof outline

Proof outline

- Let R be a resolution refutation of $A(x, y) \wedge B(x, z)$ of size s.

Proof outline

- Let R be a resolution refutation of $A(x, y) \wedge B(x, z)$ of size s.
- We want a circuit s.t. $(C(\sigma) = 0) \rightarrow A(\sigma, y)$ unsat. $(C(\sigma) = 1) \rightarrow B(\sigma, z)$ unsat.

Proof outline

- Let R be a resolution refutation of $A(x, y) \wedge B(x, z)$ of size s.
- We want a circuit s.t. $(C(\sigma) = 0) \rightarrow A(\sigma, y)$ unsat. $(C(\sigma) = 1) \rightarrow B(\sigma, z)$ unsat.
- Instead, construct such an *algorithm*.

Proof outline

- Let R be a resolution refutation of $A(x, y) \wedge B(x, z)$ of size s.
- We want a circuit s.t. $(C(\sigma) = 0) \rightarrow A(\sigma, y)$ unsat. $(C(\sigma) = 1) \rightarrow B(\sigma, z)$ unsat.
- Instead, construct such an *algorithm*.

Claim 1. For every σ, $A(\sigma, y) \wedge B(\sigma, z)$ has a refutation R_σ of size $\leq s$. R_σ can be constructed from R in polynomial time.

Proof outline

- Let R be a resolution refutation of $A(x, y) \wedge B(x, z)$ of size s.
- We want a circuit s.t. $(C(\sigma) = 0) \rightarrow A(\sigma, y)$ unsat. $(C(\sigma) = 1) \rightarrow B(\sigma, z)$ unsat.
- Instead, construct such an *algorithm*.

Claim 1. For every σ, $A(\sigma, y) \wedge B(\sigma, z)$ has a refutation R_σ of size $\leq s$. R_σ can be constructed from R in polynomial time.

Claim 2. If A' and B' have disjoint variables then every refutation of $A' \wedge B'$ contains a refutation of A' or a refutation of B'.

6. Feasible interpolation for Cutting Planes

Cutting Planes

Cutting Planes

- Manipulates linear inequalities with integere coefficients, $a_1x_1 + \ldots a_nx_n \geq b$, with $a_1, \ldots, a_n, b \in \mathbb{Z}$

Cutting Planes

- Manipulates linear inequalities with integere coefficients, $a_1x_1 + \ldots a_nx_n \geq b$, with $a_1, \ldots, a_n, b \in \mathbb{Z}$
- A clause can be represented by a linear inequality, $x \vee y \vee \neg z$ as $x + y + (1 - z) \leq 1$.

Cutting Planes

- Manipulates linear inequalities with integere coefficients, $a_1 x_1 + \ldots a_n x_n \geq b$, with $a_1, \ldots, a_n, b \in \mathbb{Z}$
- A clause can be represented by a linear inequality, $x \vee y \vee \neg z$ as $x + y + (1 - z) \leq 1$.

The rules are:

$$\frac{L \geq b}{cL \geq cb}\,,\ \text{if } c \geq 0\,,\ \frac{L_1 \geq b_1\,,\ L_2 \geq b_2}{L_1 + L_2 \geq b_1 + b_2}\,,$$

$$\frac{a_1 x_1 + \ldots a_n x_n \geq b}{(a_1/c)x_1 + \ldots (a_n/c)x_n \geq \lceil b/c \rceil}\,,\ \text{provided } c \text{ divides every } a_i\,.$$

A Cutting Planes refutation of a set of inequalities $\mathcal{L}$ is a sequence of inequalities $L_1, \ldots, L_s$ such that

A Cutting Planes refutation of a set of inequalities $\mathcal{L}$ is a sequence of inequalities $L_1, \ldots, L_s$ such that

1. every L_i is in $\mathcal{L}$, or it has been obtained from some L_{i_1}, L_{i_2}, $i_1, i_2 < i$, by one of the three rules.

A Cutting Planes refutation of a set of inequalities $\mathcal{L}$ is a sequence of inequalities $L_1, \ldots, L_s$ such that

1. every L_i is in $\mathcal{L}$, or it has been obtained from some L_{i_1}, L_{i_2}, $i_1, i_2 < i$, by one of the three rules.
2. L_s is the inequality $0 \geq 1$.

A Cutting Planes refutation of a set of inequalities $\mathcal{L}$ is a sequence of inequalities $L_1, \ldots, L_s$ such that

1. every L_i is in $\mathcal{L}$, or it has been obtained from some L_{i_1}, L_{i_2}, $i_1, i_2 < i$, by one of the three rules.
2. L_s is the inequality $0 \geq 1$.

If $A = C_1 \wedge \ldots C_m$ is a CNF formula, a Cutting Planes refutation of A is a refutation of the inequalities

$$\text{“}C_1\text{”}, \ldots, \text{“}C_m\text{”},\ x_1 \geq 0\,,\ 1 - x_1 \geq 0\,,\ x_2 \geq 0\,,\ 1 - x_2 \geq 0\,, \ldots$$

A Cutting Planes refutation of a set of inequalities $\mathcal{L}$ is a sequence of inequalities $L_1, \ldots, L_s$ such that

1. every L_i is in $\mathcal{L}$, or it has been obtained from some L_{i_1}, L_{i_2}, $i_1, i_2 < i$, by one of the three rules.
2. L_s is the inequality $0 \geq 1$.

If $A = C_1 \wedge \ldots C_m$ is a CNF formula, a Cutting Planes refutation of A is a refutation of the inequalities

$$\text{“}C_1\text{”}, \ldots, \text{“}C_m\text{”}, \; x_1 \geq 0\,, \; 1 - x_1 \geq 0\,, \; x_2 \geq 0\,, \; 1 - x_2 \geq 0\,, \ldots$$

Proposition

Let A be a CNF formula. The following are equivalent:

1. *A has a Cutting Planes refutation.*
2. *A is unsatisfiable.*

Monotone real circuit

- computes a boolean function $f : \{0,1\}^n \to \{0,1\}$.
- the inputs as well as the output are in $\{0,1\}$, but
- the intermediary gates can compute an *arbitrary* monotone real function (in two variables).

Theorem (Pudlák)

Cutting planes has monotone interpolation via real monotone circuits.

Theorem (Pudlák)

Cutting planes has monotone interpolation via real monotone circuits.

In words, assume that $A(x, y) \wedge B(x, z)$ has CP refutation of size s and A is monotone in x.
Then there exists a monotone real circuit $C(x)$ of size $O(s)$ such that for every assignment σ to the variables x

- if $C(\sigma) = 0$ then $A(\sigma, y)$ is unsatisfiable,
- if $C(\sigma) = 1$ then $B(\sigma, z)$ is unsatisfiable.

Monotone real circuit

- computes a boolean function $f : \{0,1\}^n \to \{0,1\}$.
- the inputs as well as the output are in $\{0,1\}$.
- But the intermediary gates can compute an *arbitrary* monotone real function (in two variables).

Monotone real circuit

- computes a boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$.
- the inputs as well as the output are in $\{0,1\}$.
- But the intermediary gates can compute an *arbitrary* monotone real function (in two variables).

Theorem (Pudlák)

For a suitable k, every monotone real circuit which interpolates $Clique_n^{k+1}$ and $Color_n^k$ has exponential size.

Monotone real circuit

- computes a boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$.
- the inputs as well as the output are in $\{0,1\}$.
- But the intermediary gates can compute an *arbitrary* monotone real function (in two variables).

Theorem (Pudlák)

For a suitable k, every monotone real circuit which interpolates $Clique_n^{k+1}$ and $Color_n^k$ has exponential size.

Corollary

$Clique_n^{k+1} \wedge Color_n^k$ requires exponential size CP refutations.

7. No feasible interpolation for Frege system

The Frege system

The Frege system

- the usual textbook axiomatization of propositional calculus.

The Frege system

- the usual textbook axiomatization of propositional calculus.

The rule:

$$\frac{A \to B\,,\ \ A}{B}\,.$$

The Frege system

- the usual textbook axiomatization of propositional calculus.

The rule:

$$\frac{A \to B,\ A}{B}\,.$$

Axioms:

$$A \to (B \to A)\,,$$
$$(A \to (B \to C)) \to ((A \to B) \to (A \to C))\,,$$
$$(\neg A \to \neg B) \to ((\neg A \to B) \to C)\,,$$
$$A \wedge B \to A,\ A \wedge B \to B\,,$$
$$A \to (B \to A \wedge B)\,,$$
$$A \to A \vee B,\ B \to A \vee B\,,$$
$$(A \to C) \to ((B \to C) \to (A \vee B \to C))\,.$$

The Frege system

- the usual textbook axiomatization of propositional calculus.

The Frege system

- the usual textbook axiomatization of propositional calculus.
- different textbook axiomatization are equivalent in terms of proof length.

The Frege system

- the usual textbook axiomatization of propositional calculus.
- different textbook axiomatization are equivalent in terms of proof length.
- Frege system is a robust and powerful proof system.

Theorem (B, BPR)

1. *The implication* $Clique_n^{k+1} \rightarrow \neg Color_n^k$ *has a polynomial size Frege proof. Hence Frege system doesn't have monotone feasible interpolation.*

Theorem (B, BPR)

1. *The implication* $Clique_n^{k+1} \rightarrow \neg Color_n^k$ *has a polynomial size Frege proof. Hence Frege system doesn't have monotone feasible interpolation.*
2. *Assuming that "factoring of integers is hard", Frege system does not have feasible interpolation.*

Theorem (B, BPR)

1. *The implication* $Clique_n^{k+1} \rightarrow \neg Color_n^k$ *has a polynomial size Frege proof. Hence Frege system doesn't have monotone feasible interpolation.*
2. *Assuming that "factoring of integers is hard", Frege system does not have feasible interpolation.*

Sketch of 1.
Frege system can effectively prove the Pigeonhole principle:

"if $f : [k+1] \rightarrow [k]$ *is a total function then there exist* $i \neq j \in [k+1]$ *with* $f(i) = f(j)$*".*

Sketch of 2.

- A one-to-one function f is called *one-way*, if f is easy to compute but the inverse f^{-1} is hard.

Sketch of 2.

- A one-to-one function f is called *one-way*, if f is easy to compute but the inverse f^{-1} is hard.
  - $f(p_1, p_2) := p_1 p_2$ where $p_1 \leq p_2$ are primes.

Sketch of 2.

- A one-to-one function f is called *one-way*, if f is easy to compute but the inverse f^{-1} is hard.
  - $f(p_1, p_2) := p_1 p_2$ where $p_1 \leq p_2$ are primes.
  - $f(n) = 2^n \mod p$, where p is a prime and $0 \leq n \leq p - 1$.

Sketch of 2.

- A one-to-one function f is called *one-way*, if f is easy to compute but the inverse f^{-1} is hard.
  - $f(p_1, p_2) := p_1 p_2$ where $p_1 \leq p_2$ are primes.
  - $f(n) = 2^n \mod p$, where p is a prime and $0 \leq n \leq p - 1$.
- It is not known whether such functions exist, but this assumption is central to theoretical cryptography.

Sketch of 2.

- A one-to-one function f is called *one-way*, if f is easy to compute but the inverse f^{-1} is hard.
  - $f(p_1, p_2) := p_1 p_2$ where $p_1 \leq p_2$ are primes.
  - $f(n) = 2^n \mod p$, where p is a prime and $0 \leq n \leq p - 1$.
- It is not known whether such functions exist, but this assumption is central to theoretical cryptography.
- Security of encryption schemes such as RSA relies on the existence of one-way functions.

Sketch of 2.

- A one-to-one function f is *one-way* if f is easy to compute but the inverse f^{-1} is hard.

Sketch of 2.

- A one-to-one function f is *one-way* if f is easy to compute but the inverse f^{-1} is hard.

$$A(x,y) := \text{“}f(y) = x \text{ and the } i\text{-th bit of } y \text{ is } 1\text{”}$$
$$B(x,z) := \text{“}f(z) = x \text{ and the } i\text{-th bit of } z \text{ is } 0\text{”}$$

Sketch of 2.

- A one-to-one function f is *one-way* if f is easy to compute but the inverse f^{-1} is hard.

$$\begin{aligned} A(x, y) &:= \text{“}f(y) = x \text{ and the } i\text{-th bit of } y \text{ is } 1\text{”} \\ B(x, z) &:= \text{“}f(z) = x \text{ and the } i\text{-th bit of } z \text{ is } 0\text{”} \end{aligned}$$

- $A(x, y) \to \neg B(x, z)$ is a tautology.

Sketch of 2.

- A one-to-one function f is *one-way* if f is easy to compute but the inverse f^{-1} is hard.

$$\begin{aligned} A(x, y) &:= \text{“}f(y) = x \text{ and the } i\text{-th bit of } y \text{ is } 1\text{”} \\ B(x, z) &:= \text{“}f(z) = x \text{ and the } i\text{-th bit of } z \text{ is } 0\text{”} \end{aligned}$$

- $A(x, y) \rightarrow \neg B(x, z)$ is a tautology.
- An interpolant: $C(x) = 1$ iff the i-th bit of $f^{-1}(x)$ is 1.

Sketch of 2.

- A one-to-one function f is *one-way* if f is easy to compute but the inverse f^{-1} is hard.

$$\begin{aligned} A(x,y) &:= \text{“}f(y) = x \text{ and the } i\text{-th bit of } y \text{ is } 1\text{”} \\ B(x,z) &:= \text{“}f(z) = x \text{ and the } i\text{-th bit of } z \text{ is } 0\text{”} \end{aligned}$$

- $A(x,y) \rightarrow \neg B(x,z)$ is a tautology.
- An interpolant: $C(x) = 1$ iff the i-th bit of $f^{-1}(x)$ is 1.
- If f^{-1} is hard, C must have large circuit (at least for some i).

Sketch of 2.

- A one-to-one function f is *one-way* if f is easy to compute but the inverse f^{-1} is hard.

$$
\begin{aligned}
A(x, y) &:= \text{“}f(y) = x \text{ and the } i\text{-th bit of } y \text{ is } 1\text{”} \\
B(x, z) &:= \text{“}f(z) = x \text{ and the } i\text{-th bit of } z \text{ is } 0\text{”}
\end{aligned}
$$

- $A(x, y) \to \neg B(x, z)$ is a tautology.
- An interpolant: $C(x) = 1$ iff the i-th bit of $f^{-1}(x)$ is 1.
- If f^{-1} is hard, C must have large circuit (at least for some i).
- [BPR] construct an f which is believed to be one-way (assuming that factoring is hard), but $A(x, y) \to \neg B(x, z)$ has short Frege proof.

Main open problem: Is the Frege system polynomially bounded?

Main open problem: Is the Frege system polynomially bounded?

- Feasible interpolation in its strict sense cannot be applied to Frege.

Main open problem: Is the Frege system polynomially bounded?

- Feasible interpolation in its strict sense cannot be applied to Frege.
- But maybe some modification can be.

Main open problem: Is the Frege system polynomially bounded?

- Feasible interpolation in its strict sense cannot be applied to Frege.
- But maybe some modification can be.

Question: Does Frege system have at least some form of feasible interpolation?

Main open problem: Is the Frege system polynomially bounded?

- Feasible interpolation in its strict sense cannot be applied to Frege.
- But maybe some modification can be.

Question: Does Frege system have at least some form of feasible interpolation?
Do Frege proofs have a computational content?

8. Non-clasical logics

Non-classical logics

Non-classical logics

Intuitionistic logic

- Does not have the excluded-middle principle $A \vee \neg A$, and does not allow reasoning by contradiction, $\neg\neg A \rightarrow A$.
- Intended to capture constructive reasoning.

Non-classical logics

Intuitionistic logic

- Does not have the excluded-middle principle $A \vee \neg A$, and does not allow reasoning by contradiction, $\neg\neg A \rightarrow A$.
- Intended to capture constructive reasoning.

Fuzzy logic

- Formulas take truth values from the real interval $[0, 1]$
- Intended to capture vague reasoning.

Non-classical logics

Intuitionistic logic

- Does not have the excluded-middle principle $A \vee \neg A$, and does not allow reasoning by contradiction, $\neg\neg A \to A$.
- Intended to capture constructive reasoning.

Fuzzy logic

- Formulas take truth values from the real interval $[0, 1]$
- Intended to capture vague reasoning.

Modal logic

- Obtained from propositional logic by adding the symbol $\Box$, where $\Box A$ is intended to mean "A is necessarily true".
- $\Box$ can be understood as "provable in a theory T" (such as Peano arithmetic), leading to provability logic of T.

The modal logic K

The modal logic *K*

- extend propositional logic with a new symbol $\Box$, understood as "necessarily true".

The modal logic K

- extend propositional logic with a new symbol $\Box$, understood as "necessarily true".
- In addition to propositional rules and axioms, we have

$$\text{the rule } \frac{A}{\Box A}, \quad \text{and the axiom } \Box(A \to B) \to (\Box A \to \Box B).$$

The modal logic *K*

- extend propositional logic with a new symbol $\Box$, understood as "necessarily true".
- In addition to propositional rules and axioms, we have

$$\text{the rule } \frac{A}{\Box A}, \quad \text{and the axiom } \Box(A \to B) \to (\Box A \to \Box B).$$

- Other systems of modal logic are obtained by adding axioms such as $\Box A \to A$ or $\Box A \to \Box\Box A$.

The modal logic K

- extend propositional logic with a new symbol $\Box$, understood as "necessarily true".
- In addition to propositional rules and axioms, we have

$$\text{the rule } \frac{A}{\Box A}, \quad \text{and the axiom } \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B).$$

The modal logic *K*

- extend propositional logic with a new symbol $\Box$, understood as "necessarily true".
- In addition to propositional rules and axioms, we have

$$\text{the rule } \frac{A}{\Box A}, \quad \text{and the axiom } \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B).$$

Theorem (H)

Let $A(x, y) \rightarrow B(x, z)$ be a propositional tautology with A monotone in x. Then

The modal logic *K*

- extend propositional logic with a new symbol $\Box$, understood as "necessarily true".
- In addition to propositional rules and axioms, we have

$$\text{the rule } \frac{A}{\Box A}, \quad \text{and the axiom } \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B).$$

Theorem (H)

Let $A(x, y) \rightarrow B(x, z)$ be a propositional tautology with A monotone in x. Then

1. $A(\Box x, y) \rightarrow \Box(B(x, z))$ *is K-tautology.*

The modal logic *K*

- extend propositional logic with a new symbol $\Box$, understood as "necessarily true".
- In addition to propositional rules and axioms, we have

$$\text{the rule } \frac{A}{\Box A}, \quad \text{and the axiom } \Box(A \to B) \to (\Box A \to \Box B).$$

Theorem (H)

Let $A(x, y) \to B(x, z)$ be a propositional tautology with A monotone in x. Then

1. *$A(\Box x, y) \to \Box(B(x, z))$ is K-tautology.*
2. *If the tautology has a K-proof of size s then there exists a monotone interpolant of A and B of size $O(s^2)$.*

Thank you