# Proof Complexity I: Introduction to propositional proof complexity

Sam Buss

Caleidoscope Research School
Institute Henri Poincaré, Paris
June 17-18, 2019

This talk discusses:

**Proof systems:**
Frege proofs, extended Frege proofs, abstract proof systems, resolution, cutting planes, nullstellensatz, the polynomial calculus.

**The extension rule:**
Frege versus extended resolution (equivalent to extended Frege).
Resolution versus extended resolution

**Interpolation and lower bounds:**
Resolution.
Cutting planes.

**Automatizability and conditional lower bounds.**

# Propositional logic, satisfiability, tautologies

**Propositional formulas:**

- Variables: Range over *True*/*False*.
- Literal: Variables and negated variables.
- Formulas: Formed from variables/literals, and propositional connectives, such as $\wedge$, $\vee$, $\rightarrow$, $\neg$.
- CNF, DNF - Conjunctive/Disjunctive Normal Form Formulas.

Satisfiability and Validity:

- A formula $\varphi$ is a **tautology** iff every truth assignment makes $\varphi$ true.
- A formula $\varphi$ is **satisfiable** iff some truth assignment makes $\varphi$ true.
- $\varphi$ is unsatisfiable iff $\neg\varphi$ is a tautology.
- It is NP-hard to determine satisfiability/validity of $\varphi$.
- One way to establish unsatisfiability is to give a **proof** of $\neg\varphi$.

The **Frege proof system** $\mathcal{F}$ is a "textbook-style" propositional proof system with Modus Ponens as its only rule of inference.

Variables: $x$, $y$, $z, \ldots$ range over *True*/*False*.
Connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$.

**Modus Ponens:** $$\frac{\varphi \qquad \varphi \rightarrow \psi}{\psi}.$$

**Axiom Schemes:**

$$\begin{array}{l}
\varphi \rightarrow \psi \rightarrow \varphi \\
(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi) \\
\varphi \;\rightarrow\; \psi \;\rightarrow\; \varphi \wedge \psi \\
\varphi \wedge \psi \;\rightarrow\; \varphi \\
\varphi \wedge \psi \;\rightarrow\; \psi
\end{array}$$

and 5 more axiom schemes.

**Defn:** The **size** of a Frege proof is the number of symbols in the proof. $\mathcal{F} \overset{m}{\vdash} \varphi$ means $\varphi$ has an $\mathcal{F}$ proof of size $m$.
The **size** of a formula is the number of symbols in the formula.

**Thm:** $\mathcal{F}$ is sound and complete.
In fact: $\mathcal{F}$ is implicationally sound and implicationally complete.

*Implicational Soundness and Completeness:*
There is an $\mathcal{F}$-proof of $\varphi$ from hypotheses $\Gamma$ iff $\Gamma \vDash \varphi$.
In particular, every tautology has an $\mathcal{F}$-proof.

**Pf idea:** Formalize the method of truth tables; i.e., try all truth assignments.

More generally, a **Frege system** is specified by any finite complete set of Boolean connectives and finite set of axiom schemes and rule schemes, provided it is implicationally sound and implicationally complete.

By completeness, every tautology has an $\mathcal{F}$proof.

**Open problem:** Is there a polynomial $p(n)$ such that every tautology $\varphi$ has an $\mathcal{F}$-proof of size $\leq p(n)$, where $n$ is the size of $\varphi$. That is, is $\mathcal{F}$ polynomially bounded?

The answer is the same for all Frege systems, in that any two Frege systems "p-simulate" each other.
[Reckhow'76; Cook-Reckhow'79]

**Defn:** An **abstract proof system** is a polynomial time function $f$ mapping $\{0,1\}^*$ *onto* the set of tautologies.
$w$ is an $f$-**proof** of $\varphi$ iff $f(w) = \varphi$.
The **size** of $w$ is $|w|$, i.e. the length of $w$.

*Example:* For the Frege system $\mathcal{F}$:

$$f_{\mathcal{F}}(w) = \begin{cases} \text{the last line of } w & \text{if } w \text{ is an } \mathcal{F}\text{-proof} \\ (x \vee \neg x) & \text{otherwise} \end{cases}$$

Similar constructions allow very strong systems, e.g. ZF set theory, to be abstract proof systems.

**Thm.** [CR'79] There ia polynomially bounded abstract proof system iff $\mathrm{NP} = \mathrm{coNP}$.

**Proof idea:** The set of tautologies is $\mathrm{coNP}$-complete.

**Defn:** [ess. Tseitin '68] Extension allows introduction of new variables for formulas; namely the **extension rule**:

$$z \leftrightarrow \varphi$$

where $z$ is a variable not appearing in earlier lines the proof, in $\varphi$, or in the last line of the proof.
The **extended Frege system (**$e\mathcal{F}$**)** is Frege ($\mathcal{F}$) plus the extension rule.

**Thm.** [Statman'77] If $\mathcal{F} \overset{m \text{ steps}}{\vdash} \varphi$, then $\varphi$ has a $e\mathcal{F}$-proof of size $O(m + |\varphi|^2)$, that is $e\mathcal{F} \overset{O(m+|\varphi|^2)}{\vdash} \varphi$.

Thus the *size* of extended Frege proofs is essentially the same as the *number of lines* in Frege proofs.

**Proof idea:** Introduce extension variables for the formulas in the Frege proof; thereby reduce all lines to constant size with only a linear increase in the number of lines in the proof. □

Using extension allows succinct representation of Boolean circuits $C$, by introducing an extension variable for each gate in $C$. Thus, in effect:

- A Frege proof is a proof in which each line is a Boolean *formula*.
- An extended Frege proof is a proof in which each line is a Boolean *circuit*.

It is conjectured that circuits cannot be converted into polynomial size equivalent formulas; the corresponding conjecture is that $\mathcal{F}$ does not (p-)simulate $e\mathcal{F}$. [Cook-Reckhow'79]

There is no known direct connection between these conjectures:

- Formulas might polynomially represent circuits, yet this might not be provable with $\mathcal{F}$ proofs.
- Conversely, $\mathcal{F}$ might simulate $e\mathcal{F}$ by some other means.

# The pigeonhole principle as a propositional tautology

Let $[n] = \{0, \ldots, n-1\}$.
Let $i$'s range over members of $[n{+}1]$ and $j$'s range over $[n]$.

$$\mathbf{Tot}_i^n \ := \ \bigvee_{j\in[n]} x_{i,j}. \quad \text{“Total at } i\text{”}$$

$$\mathbf{Inj}_j^n \ := \ \bigwedge_{0\le i_1<i_2\le n} \neg(x_{i_1,j} \wedge x_{i_2,j}). \quad \text{“Injective at } j\text{”}$$

$$\mathbf{PHP}_n^{n+1} \ := \ \neg\Big(\bigwedge_{i\in[n+1]} \mathrm{Tot}_i^n \wedge \bigwedge_{j\in[n]} \mathrm{Inj}_j^n\Big).$$

$\mathrm{PHP}_n^{n+1}$ is a tautology.

**Thm:** $\mathrm{PHP}_n^{n+1}$ has polynomial size $e\mathcal{F}$ proofs. [Cook-Reckhow'79]

# Cook-Reckhow's $e\mathcal{F}$ proof of $\mathrm{PHP}^{n+1}_n$

Code the graph of $f : [n+1] \to [n]$ with variables $x_{i,j}$ indicating that $f(i) = j$.

$\mathrm{PHP}^{n+1}_n(\vec{x})$: "$f$ is not both total and injective"

Use **extension** to introduce new variables

$$x^{\ell-1}_{i,j} \leftrightarrow x^{\ell}_{i,j} \vee (x^{\ell}_{i,\ell-1} \wedge x^{\ell}_{\ell,j}).$$

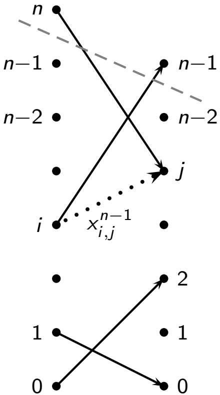
for $i \le \ell$, $j < \ell$; where $x^n_{i,j} \leftrightarrow x_{i,j}$.

Let $\mathrm{PHP}^{\ell+1}_{\ell}$ be over variables $x^{\ell}_{i,j}$.

Prove, for each $\ell$ that

$$\neg\mathrm{PHP}^{\ell+1}_{\ell}(\vec{x}^{\ell}) \to \neg\mathrm{PHP}^{\ell}_{\ell-1}(\vec{x}^{\ell-1}).$$

Finally derive $\mathrm{PHP}^{n+1}_n(\vec{x})$ from $\mathrm{PHP}^2_1(\vec{x}^1)$. □



Theorem (Cook-Reckhow '79)

$\mathrm{PHP}_n^{n+1}$ *has polynomial size extended Frege proofs.*

**Theorem (Cook-Reckhow '79)**

$\mathrm{PHP}_n^{n+1}$ *has polynomial size extended Frege proofs.*

**Theorem (B '87)**

$\mathrm{PHP}_n^{n+1}$ *has polynomial size Frege proofs.*

**Theorem (Cook-Reckhow '79)**

$\mathrm{PHP}_n^{n+1}$ *has polynomial size extended Frege proofs.*

**Theorem (B '87)**

$\mathrm{PHP}_n^{n+1}$ *has polynomial size Frege proofs.*

**Theorem (B '15)**

$\mathrm{PHP}_n^{n+1}$ *has quasipolynomial size Frege proofs.*

# Polynomial size $\mathcal{F}$ proofs of $\mathrm{PHP}_n^{n+1}$ [B'86]

*Proof is based on counting.*

- There are polynomial-size formulas for **vector addition.** For $m, n \in \mathbb{N}$, input variables define the $n$ bits of $m$ integers. The $n + \log m$ formulas $\mathrm{CSA}_{m,n}$ define the bits of their sum. Based on carry-save-addition circuits.
- $\mathcal{F}$ can prove elementary facts about sums of vectors of integers as computed with $\mathrm{CSA}$ formulas and "2-3" adder trees

**Proof sketch: ($\mathcal{F}$)** Assume $\mathrm{PHP}_n^{n+1}$ is false. Proceed by "brute force induction" on $i' \le n+1$ to prove that

- The number of $j \le n$ such that $\bigvee_{i \le i'} p_{i,j}$ is greater than or equal to $i'$.
- The number of $j \le n$ such that $\bigvee_{i \le i'} p_{i,j}$ is less than or equal to $i'$.

Conclude by obtaining a contradiction $n + 1 \le n$. □

# Cook-Reckhow's proof of $\mathrm{PHP}_n^{n+1}$ as a Frege proof [B'15]

Let $G^\ell$ be the directed graph with:
edges $(\langle i,0\rangle,\langle j,1\rangle)$ such that $x_{i,j}$ holds, and
edges $(\langle i,1\rangle,\langle i{+}1,0\rangle)$ such that $i{\ge}\ell$ (blue edges).

For $i\le\ell$, $j<\ell$, let $\varphi^\ell_{i,j}$ express

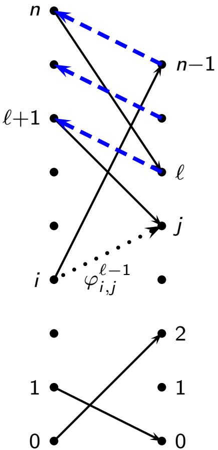
"Range node $\langle j,1\rangle$ is reachable
from domain node $\langle i,0\rangle$ in $G^\ell$".

$\varphi^\ell_{i,j}$ is a quasi-polynomial size formula via an $\mathrm{NC}^2$ definition of reachability.

For each $\ell$, prove that

$$\neg\mathrm{PHP}_\ell^{\ell+1}(\vec\varphi^\ell)\to\neg\mathrm{PHP}_{\ell-1}^\ell(\vec\varphi^{\ell-1}).$$

Finally derive $\mathrm{PHP}_n^{n+1}(\vec x)$ from $\mathrm{PHP}_1^2(\vec\varphi^1)$. □



Thus, $\mathrm{PHP}_n^{n+1}$ no longer provides evidence for Frege not quasipolynomially simulating $e\mathcal{F}$.

[Bonet-B-Pitassi'94] "Are there hard examples for Frege?": examined candidates for separating Frege and $e\mathcal{F}$. Very few were found:

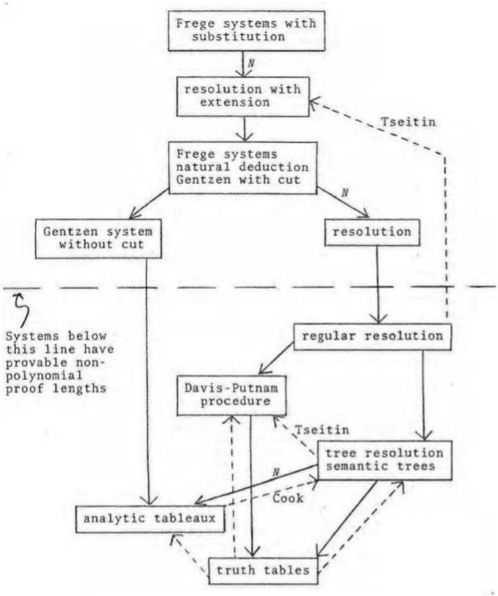
- Cook's $AB = I \Rightarrow BA = I$, Odd-town theorem, etc.
  Now known to have quasipolynomial size $\mathcal{F}$-proofs, by proving matrix determinant properties with $\mathrm{NC}^2$ formulas.
  [Hrubes-Tzameret'15; Tzameret-Cook'??]
- Frankl's Theorem
  Also quasi-polynomial size $\mathcal{F}$ proofs. [Aisenberg-B-Bonet'15]

[Kołodziejczyk-Nguyen-Thapen'11]: Local improvement principles, mostly settled by [Beckmann-B'14], $\mathrm{RLI}_2$ still open.

# Abstract Proof Systems

**Defn:** Let $f$, $g$ be abstract proof systems.
$f$ **simulates** $g$ if there is a polynomial $q(n)$ s.t., whenever $g(w) = \varphi$, there is a $v$, $|v| \le q(|w|)$ such that $f(v) = \varphi$.
$f$ **p-simulates** $g$ if there is a polynomial-time computable $h(w)$, such that, whenever $g(w) = \varphi$, we have $f(h(w)) = \varphi$.
$f$ is **polynomially bounded** if, for some polynomial $q(n)$, every tautology $\varphi$ has an $f$-proof $w$ of size $\le q(|\varphi|)$.

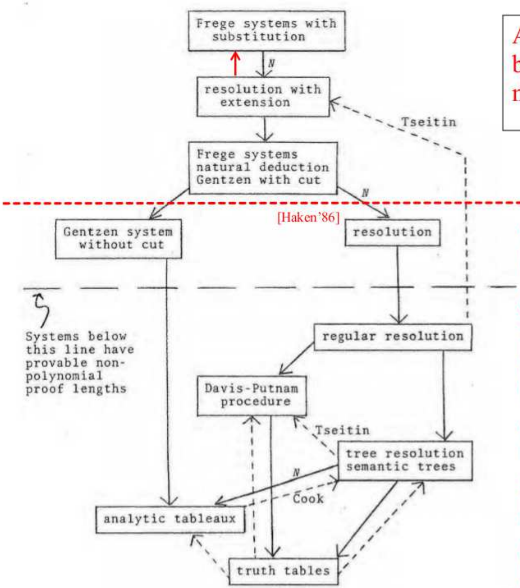
**1.** Any two Frege systems p-simulate each other.
**2.** Any two $e\mathcal{F}$ systems p-simulate each other.
**3.** Extended Frege systems p-simulate Frege systems.
**4.** It is open whether $\mathcal{F}$ simulates $e\mathcal{F}$.
**5.** It is open whether $\mathcal{F}$ of $e\mathcal{F}$ is polynomially bounded.
**6.** It is open whether there is an abstract proof system which p-simulates all abstract proof systems.


Frege systems with substitution
N
resolution with extension
Tseitin
Frege systems natural deduction Gentzen with cut
N
Gentzen system without cut
resolution
Systems below this line have provable non-polynomial proof lengths
regular resolution
Davis-Putnam procedure
Tseitin
tree resolution semantic trees
N
Cook
analytic tableaux
truth tables

R.A. Reckhow, PhD thesis, 1975

**Cook's Program:** Prove NP≠coNP by proving there is no polynomially bounded propositional proof system.

As of 1975: Systems above the line were not known to not be polynomially bounded.


Frege systems with substitution
N
resolution with extension
Tseitin
Frege systems natural deduction Gentzen with cut
N
Gentzen system without cut
[Haken'86]
resolution
Systems below this line have provable non-polynomial proof lengths
regular resolution
Davis-Putnam procedure
Tseitin
tree resolution semantic trees
N
Cook
analytic tableaux
truth tables

R.A. Reckhow, PhD thesis, 1975

As of 2015, proof systems below the line are known to not be polynomially bounded:

Constant-depth ($AC^0$) Frege
[Ajtai'88; Pitassi-Beame-Impagliazzo'93; Krajicek-Pudlak-Woods'95]

Constant-depth Frege with counting mod *m* axioms
[Ajtai'94; Beame-Impagliazzo-Krajicek-Pitassi-Pudlak'96; B-Impagliazzo-Krajicek-Pudlak-Razborov-Sgall'96; Grigoriev'98]

Cutting Planes
[Pudlak'97]

Nullstellensatz
[B-Impagliazzo-Krajicek-Pudlak-Razborov-Sgall'96; Grigoriev'98]

Polynomial calculus
[Razborov'98; Impagliazzo-Pudlak-Sgall'99; Ben-Sasson-Impagliazzo'99; B-Grigoriev-Impagliazzo-Pitassi'96; B-Impagliazzo-Krajicek-Pudlak-Razborov-Sgall'96; Alekhnovich-Razborov'01]

# Resolution

**Resolution** is a refutation system, refuting sets of clauses. Thus, resolution is a system for refuting CNF formulas, equivalently, a system for proving DNF formulas are tautologies.

- A literal is a variable $x$ or a negated variable $\overline{x}$.
- A *clause* is a set of literals, interpreted as their disjunction
- A set $\Gamma$ of clauses is a CNF formula
- Resolution rule: $$\frac{x, C \qquad \overline{x}, D}{C \cup D}$$
- A **resolution refutation** of $\Gamma$ is a derivation of the empty clause from clauses in $\Gamma$.
- This allows resolution to be a proof system for DNF formulas.

**Thm:** Resolution is sound and complete (for CNF refutations)

**Defn: Extension rule** for resolution: For $x$ and $y$ literals, and letting $z$ be a *new* variable, introduce $z \leftrightarrow (x \wedge y)$ by adding the clauses:

$$\{\overline{x}, \overline{y}, z\} \qquad \{\overline{z}, x\} \qquad \{\overline{z}, y\}.$$

**Resolution as an abstract proof system:** Given $\varphi$, introduce clauses $\Gamma$ for the extension variables $z_\psi$ for all subformulas $\psi$ of $\varphi$. A resolution proof of $\varphi$ is a resolution refutation of $\overline{z}_\varphi, \Gamma$.
**Extended resolution** is resolution augmented with unrestricted use of the extension rule.

**Thm:** Extended resolution and extended Frege p-simulate each other.

# Can the extension rule help resoution?

**Yes, extension helps resolution;** Since $\mathrm{PHP}_n^{n+1}$ has polynomial size $e\mathcal{F}$ proofs and since:

**Thm:** [Haken'86, Raz'02, Razborov'03, many others]
The pigeonhole principle ($\mathrm{PHP}$) requires resolution proofs of size $2^{n^\epsilon}$ (even $\mathrm{PHP}_n^m$ for $m \gg n$).

For $\mathrm{PHP}_n^{n+1}$, a similar bound can be proved for constant-depth Frege proofs.

**Thm:** [BIKPPW'92]
Depth $d$ Frege proofs of $\mathrm{PHP}_n^{n+1}$ require size $2^{n^\epsilon}$ where $\epsilon = \epsilon(d)$.

**Def'n** Constant depth Frege proofs are formulated using the sequent calculus, with only connectives $\wedge$, $\vee$ applied to literals.

The **depth** of a Boolean formula is the number of alternations of $\wedge$'s and $\vee$'s.

The **depth** of a proof is the max depth of its formulas.

# Proof method for $\mathrm{PHP}_n^{n+1}$ resolution lower bound

Proof uses two ingredients.

**Def'n.** Let Γ be an unsatisfiable set of clauses.
$\textsc{ResLen}(\Gamma)$ is the minimum number of steps in a resolution refutation of Γ.
$\textsc{ResWidth}(\Gamma)$ is the minimum width of a resolution refutation of Γ, where "width" is the maximum number of literals in any clause.

**Theorem (Ben-Sasson, Wigderson'01)**

*If Γ is a k-CNF over n variables, then*

$$ResLen(\Gamma) \geq exp\left(\Omega\left(\frac{(\textsc{ResWidth}(\Gamma)-k)^2}{n}\right)\right)$$

The $\textsc{ResLen}$ - $\textsc{ResWidth}$ tradeoff cannot be used directly with $\mathrm{PHP}_n^{n+1}$ since the $\mathrm{Tot}_i^n$ clauses are large and thus force $k$ to be large.

But, **sparse PHP** can be used instead.
For $G$ a bipartite graph on $[n+1] \oplus [n]$, replace $\mathrm{Tot}_i^n$ with

$$G\text{-}\mathbf{Tot}_i^n \;:=\; \bigvee_{(i,j)\in G} x_{i,j}. \quad \text{"Total at } i\text{"}$$

For $G$ a constant degree graph with suitable expansion properties, we have $\textsc{ResWidth}(G\text{-}\mathrm{PHP}_n^{n+1})$ is $\omega(n)$.
[Ben-Sasson,Wigderson] Hence

**Theorem (Haken'86, Ben-Sasson,Wigderson'01)**

*$G$-$\mathrm{PHP}_n^{n+1}$ and hence $\mathrm{PHP}_n^{n+1}$ requires resolution proofs of length $exp(\Omega(n))$.*

# Craig Interpolation

[Bonet-Pitassi-Raz'95, Razborov'95, Krajíček'97, Pudlák'97]

**Defn:** Suppose $A(\vec{p},\vec{r}) \wedge B(\vec{q},\vec{r})$ is unsatisfiable, where $A$ and $B$ depend only on the variables indicated.
A **Craig interpolant** for this formula is a predicate $C(\vec{r})$ such that

- If $\neg C(\vec{r})$, then $A(\vec{p},\vec{r})$ is unsatisfiable.
- If $C(\vec{r})$, then $B(\vec{q},\vec{r})$ is unsatisfiable.
- Equivalently, $A(\vec{p},\vec{r}) \to C(\vec{r})$ and $C(\vec{r}) \to \neg B(\vec{q},\vec{r})$ are both tautologies.

**Thm:** A Craig interpolant always exists when $A(\vec{p},\vec{r}) \wedge B(\vec{q},\vec{r})$ is unsatisfiable.

**Pf:** Take $C(r)$ to be either

$$(\exists\vec{p})A(\vec{p},\vec{r}) \qquad \text{or} \qquad (\forall\vec{q})\neg B(\vec{q},\vec{r}).$$

However, the Craig interpolant may not be a feasible predicate of $\vec{r}$.

Craig interpolant: $(\exists \vec{p})A(\vec{p},\vec{r})$ or $(\forall \vec{q})\neg B(\vec{q},\vec{r})$.

### Theorem (Krajíček'97)

*Suppose a set of clauses $A(\vec{p},\vec{r}), B(\vec{q},\vec{r})$ has a resolution refutation of size $m$, and that variables $\vec{r}$ all appear only positively in the clauses in $A(\vec{p},\vec{r})$ or only negatively in the clauses in $B(\vec{q},\vec{r})$. Then, there is a Craig interpolant which is computed by a* monotone *Boolean circuit of size $O(m)$.*

A **monotone** circuit is constructed from literals $r_i$, $\overline{r_i}$ and $\wedge$ and $\vee$. If the refutation is tree-like, the interpolant is a monotone Boolean *formula*.

Application: The clique-coloring principles require exponential size resolution refutations.

CLIQUE-COLORING PRINCIPLE: A graph cannot have both a size $m$ clique and a size $m-1$ coloring.

Proof of the Craig interpolation property:

A resolution refutation $R$ transforms directly to a monotone circuit.

Each clause $C$ in $R$ corresponds to a gate $g_C$.

| | |
|---|---|
| $C$ is an $A(\vec{p},\vec{r})$ clause | $g_C := \bot$ |
| $C$ is a $B(\vec{q},\vec{r})$ clause | $g_C := \top$ |
| $C, p_i \quad D, \overline{p_i} \ / \ C, D$ | $g_{CD} := g_{Cp_i} \vee g_{D\overline{p_i}}$ |
| $C, q_i \quad D, \overline{q_i} \ / \ C, D$ | $g_{CD} := g_{Cp_i} \wedge g_{D\overline{p_i}}$ |
| $C, r_i \quad D, \overline{r_i} \ / \ C, D$ | $g_{CD} := (r_i \vee g_{Cr_i}) \wedge g_{D\overline{r_i}}$, or<br>$g_{CD} := g_{Cr_i} \vee (r_i \wedge g_{D\overline{r_i}})$,<br>depending on whether $\vec{r}$ is monotone in $A$ or in $B$. |

Invariant: $g_C$ computes an interpolant that is correct for any assignment falsifying $C$. □

# Clique Coloring clauses

**Variables:** $a \in [m]$, $c \in [m-1]$, and $i < j \in n$.

- $p_{a,i}$ - node $i$ is the $a$-th member of a clique.
- $q_{i,c}$ - node $i$ has color $c$.
- $r_{i,j}$ - there is a edge joining edges $i$ and $j$.

$\boldsymbol{A(\vec{p},\vec{r})}$ **clauses:** (Clique of size $m$)

- $\bigvee_i p_{a,i}$ - for each $a \in [m]$
- $\overline{p_{a,i}} \vee \overline{p_{a',i}}$ - for $a < a' \in [m]$, $i \in [n]$.
- $\overline{p_{a,i}} \vee \overline{p_{a',j}} \vee r_{i,j}$ - for distinct $a, a' \in [m]$, distinct $i, j \in [n]$.

$\boldsymbol{B(\vec{q},\vec{r})}$ **clauses:** (Coloring of size $m-1$)

- $\bigvee_c q_{i,c}$ - for $i \in [n]$.
- $\overline{q_{i,c}} \vee \overline{q_{i,c'}}$ - for $a < c' \in [m-1]$, $i \in [n]$.
- $\overline{q_{i,c}} \vee \overline{q_{j,c}} \vee \overline{r_{i,j}}$ - for distinct $c \in [m-1]$, distinct $i < j \in [n]$.

The monotonicity hypothesis holds for both $A(\vec{p},\vec{r})$ and $B(\vec{q},\vec{r})$.

**Theorem** [Krajíček'97] Any resolution refutation of the clique-coloring tautology for $m = n^{1/2}$ requires size $2^{\omega(n^{3/4})}$.

Proof: This is a corollary to the Craig interpolation theorem just proved, and the known exponential lower bounds on the size of monotone Boolean circuits that distinguish between graphs with large cliques and graphs with large colorings.
[Razborov'85, Alon Boppana'87]

The variables $\vec{r}$ encode a graph $G$.

$C(\vec{r})$ is false means: $A(\vec{p}, \vec{r})$ is unsatisfiable, i.e., $G$ does not have a clique of size $m$.

$C(\vec{r})$ is true means: $B(\vec{q}, \vec{r})$ is unsatisfiable, i.e., $G$ does not have an $m{-}1$ coloring.

Any monotone Boolean circuit for $C$ must be large. □

# Cutting planes proofs

Variables $x_1, x_2, \ldots$ are 0/1 valued (0="False", 1="True").

Lines in a cutting planes proof are linear inequalities with integer coefficients:

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq a_0.$$

Clauses become inequalities: for example

$x \vee y \vee z$ becomes $x + y + z \geq 1$, and

$x \vee \overline{y} \vee z$ becomes $x - y + z \geq 0$.

Note that $\overline{y}$ is replaced with $1 - y$.

# Cutting planes refutations

Cutting planes is a *refutation system*:
Initial lines are logical axioms,
or encode hypotheses (often obtained from clauses).

**Logical axioms:** $x_i \geq 0$ and $-x_1 \geq -1$.

Valid inferences are *Addition* and *Division*.

**Addition rule:**

$$\frac{\sum a_i x_i \geq a_0 \qquad \sum b_i x_i \geq b_0}{\sum (a_i + b_i) x_i \geq a_0 + b_0}$$

**Division rule:** If $c > 0$ and $c | a_i$ for all $i > 0$,

$$\frac{\sum a_i x_i \geq a_0}{\sum (a_i / c) x_i \geq \lceil a_0 / c \rceil}$$

The final line of a refutation must be $0 \geq 1$.

# Alternate formulation of Division

Sometime *Division* is reformulated:

**Division$'$ rule:** If $c|a_i$ for all $i > 0$,

$$\frac{\sum a_i x_i \geq a_0}{\sum \lceil a_i/c \rceil x_i \geq \lceil a_0/c \rceil}$$

The two formulations are equivalent in the presence of Addition.

**Example:** Let $\Gamma$ contain the clauses

$$\overline{x} \lor \overline{y} \quad \text{and} \quad \overline{x} \lor \overline{z} \quad \text{and} \quad \overline{y} \lor \overline{z}.$$

This expresses "No two of $x$, $y$, $z$ are true".

Cutting planes expresses these clauses as three inequalities:

$$-x - y \geq -1 \quad \text{and} \quad -x - z \geq -1 \quad \text{and} \quad -y - z \geq -1.$$

**Addition** gives: $\quad -2x + -2y - 2z \geq -3.$

**Division** by $c = 2$ gives: $\quad -x - y - z \geq -1.$

I.e., $x + y + z \leq 1$. This is a more succinct way of expressing that no two of $x$, $y$, $z$ are true.

**Theorem:** Cutting planes has polynomial size refutations of the $\mathrm{PHP}_n^{n+1}$ principle.

**Proof idea:** Use the totality axioms to derive $\sum_{i,j} p_{i,j} \ge n+1$. Use the injectivity axioms to prove $\sum_{i,j} p_{i,j} \le n$, similar to the argument in the example.

Conclude $0 \ge 1$. □

**Hence:** Resolution does not p-simulate cutting planes.

**Also:** Constant-depth Frege does not p-simulate cutting planes.

# Two simulations

**Theorem:** Cutting planes p-simulates resolution.

**Proof idea:** It is straightforward to simulate a single resolution step with addition and division.

**Thm:** [Goerdt'92] Cutting planes is p-simulated by Frege systems.

**Proof idea:** Use carry-save-addition iterated integer division formulas to express the lines in a cutting planes refutation.

# A cutting planes bound for Clique-Coloring

**Thm:** [Pudlák'97] Suppose a set of clauses $A(\vec{p},\vec{r}), B(\vec{q},\vec{r})$ has a cutting planes refutation of $m$ steps, and that the variables $\vec{r}$ appear only positively in the clauses in $A(\vec{p},\vec{r})$ or negatively in the clauses in $B(\vec{q},\vec{r})$. Then, there is Craig interpolant which is computed by a monotone real circuit of size $m^{O(1)}$.

**Corollary:** (Using [Razborov'85, Alon-Boppana'87.) Cutting Planes does not have polynomial size refutations of the Clique-Coloring clauses expressing that a graph both is $k$-colorable and has a $k+1$ clique.

Open: Find methods other Craig interpolation for giving lower bounds to cutting planes proofs.

# Saturation

Let $x_i^0$ denote $x_i$ and $x_i^1$ denote $\overline{x_i}$. Thus $x_i^1 = 1 - x_i^0$.

Sometimes cutting planes lines are expressed as linear combinations of **literals** with non-negative integer coefficients:

$$a_1^0 x_1 + a_1^1 \overline{x_1} + a_2^0 x_2 + a_2^1 \overline{x_2} + \cdots + a_n^0 x_n + a_n^1 \overline{x_n} \ \geq \ a_0$$

where $\min\{a_i^0, a_i, 1\} = 0$.

In **CP-Saturation**, the *Division* rule is replaced with *Saturation*.

**Saturation rule:** If $c | a_i$ for all $i > 0$,

$$\frac{\sum_{i,\sigma} a_i^\sigma \cdot x_i^\sigma \geq a_0}{\sum_{i,\sigma} \min\{a_i^\sigma, a_0\} \cdot x_i^\sigma \geq a_0}$$

CP-Saturation is particularly used in several SAT solvers based on cutting planes [Chai-Kuehlmann'05]

# Saturation versus Division

**Theorem (Gocht-Nordström-Yehudayoff'19)**

- *CP-Saturation does not simulate Cutting Planes.*
- *A single Saturation inference can require superpolynomially many steps to simulate in Cutting Planes.*

In spite of the second part, it is open whether Cutting Planes p-simulates CP-Saturation.

# Nullstellensatz and polynomial calculus

Work over a finite field, characteristic $p$.
Variables $x_1, x_2, \ldots$ are $0/1$ valued.

A polynomial $f$ is identified with the assertion $f = 0$.

A set of *initial* polynomials $\{f_j\}_j$ is **refuted** in the **Nullstellensatz system** by polynomials $g_j$, $h_i$ such that

$$\sum f_j \cdot g_j + \sum (x_i^2 - x_i) \cdot h_i = 1,$$

where equality indicates equality as polynomials.

A **polynomial calculus** refutation uses the inferences of addition and multiplication:

$$\frac{f \quad g}{f+g} \qquad\qquad \frac{f}{f \cdot g}$$

A **polynomial calculus refutation** of a set of polynomials $\{f_j\}_j$ is a derivation of 1 from the $f_j$'s and the polynomials $(x_i^2 - x_i)$.

It is more common to work with the *degree* of nullstellensatz or polynomial calculus proofs, rather than their size. These systems are known to not be simulated by resolution or bounded depth Frege; conversely, several lower bounds are known.

One sample result:

**Thm:** [Razborov'98] Any polynomial calculus proof of $PHP_n^{n+1}$ must have degree $\Omega(n)$.

# Automatizability

**Defn:** A proof system $T$ is **automatizable** (in polynomial time) if there is a procedure, which given a formula $\varphi$, produces a $T$-proof of $\varphi$ in time bounded by a polynomial of the size of the shortest $T$-proof of $\varphi$ (if any).

**Defn:** A proof system $T$ has **feasible interpolation** if there is polynomial time procedure $C(-,-)$ so that if $P$ is a $T$-proof of $\neg(A(\vec{p},\vec{r}) \wedge B(\vec{q},\vec{r}))$, then $C(P,\vec{r})$ is a Craig interpolant for $A(\vec{p},\vec{r}) \wedge B(\vec{q},\vec{r})$.

**Thm:** [Bonet-Pitassi-Raz'00] If $T$ is automatizable, then $T$ has feasible interpolation.

**Thm:** [Krajíček-Pudlák'95, also B'97] The extended Frege system $e\mathcal{F}$ does not have feasible interpolation and thus is not automatizable, unless the RSA encryption function, the discrete logarithm encryption function, and the Rabin encryption function can be inverted in polynomial time.

**Thm:** [Bonet-Pitassi-Raz'00] The Frege system $\mathcal{F}$ does not have feasible interpolation and thus is not automatizable, unless Blum integers can be factored in polynomial time.

**Defn: Blum integers** are products of two primes, each congruent to 3 mod 4.

A related theorem holds for bounded depth Frege systems under a stronger hardness assumption about Blum integers. [BDGMP'03].

**Thm:** [Alekhnovich-Razborov'03] Resolution and tree-like resolution are not automatizable unless the parameterized hierarchy class $W[P]$ is fixed-parameter tractable via randomized algorithms with one-sided error.

On the other hand:

**Thm:** [Beame-Pitassi'96; building on CEI'96]
Tree-like resolution is automatizable in time $n^{\log S}$ where $n$ is the number of variables, and $S$ is the size of the shortest tree-like resolution refutation. (This is quasipolynomial time.)

Resolution is automatizable in time $n^{\sqrt{n \log S}}$.

**Open:** Is resolution automatizable in quasi-polynomial time?
RECENT RESULT: It is $\mathrm{NP}$-hard. [Atserias-Müller'19]

End of first part!

## Some survey articles:

- S. Buss, "Towards NP-P via Proof Complexity and Search", Annals of Pure and Applied Logic 163, 7 (2012) 906-917.
- S. Buss, "Propositional Proof Complexity: An Introduction", In Computational Logic, edited by U. Berger and H. Schwichtenberg, Springer-Verlag, Berlin, 1999, pp. 127-178.
- P. Beame and T. Pitassi, "Propositional Proof Complexity: Past, Present and Future", In Current Trends in Theoretical Computer Science Entering the 21st Century, World Scientific, 2001, pp. 42-70.
- P. Beame with A. Sabharwal, "Propositional Proof Complexity", In Computational Complexity Theory, IAS/Park City Clay Mathematics Series 10, 2000, pp. 199-246.
- P. Pudlák, "Twelve problems in proof complexity", In Proc. 3rd International Computer Science Symposium in Russia, CSR 2008, pp.13-27
- N.Segerlind, "The Complexity of Propositional Proofs", Bulletin of Symbolic Logic 13 (2007) 417-481.