HOW HARD IS IT TO CERTIFY UNSATISFIABILITY OF A RANDOM FORMULA ?

Toniann Pitassi
U Toronto, IAS,
Columbia U.

K-SAT

Input: KCNF formula f

$$f = (x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee x_4 \vee x_7)(\bar{x}_2 \vee x_3)\ldots(x_9 \vee \bar{x}_{10})$$

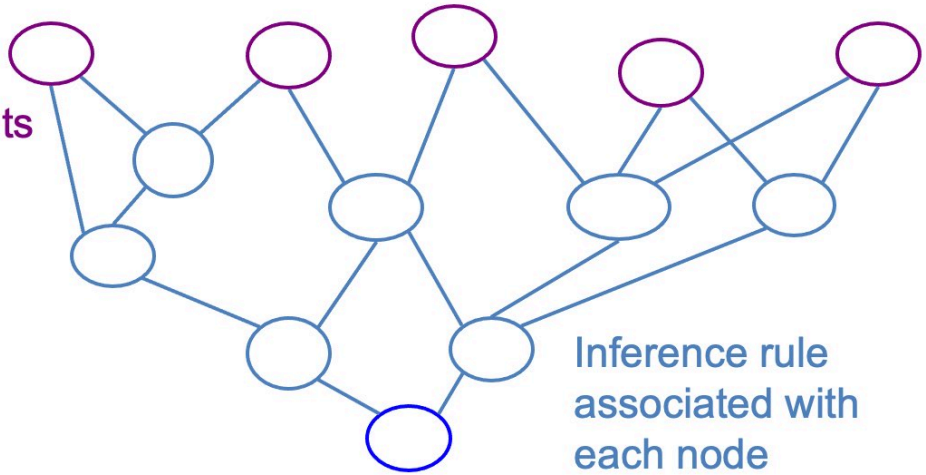
Output: SAT iff $\exists \alpha \quad f(\alpha) = 1$

UNSAT iff $\forall \alpha \quad f(\alpha) = 0$

K-SAT is NP-COMPLETE

HOW TO CERTIFY/PROVE f IS UNSAT ?

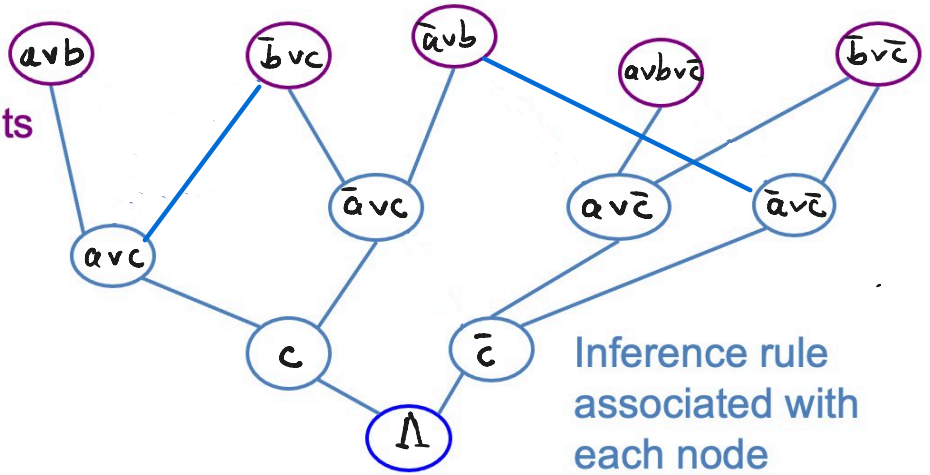
The graph of a proof



Sink labelled by tautology
(or Λ for refutation)

Example: Graph of a RESOLUTION Proof

$f = (a \vee b)(\bar{b} \vee c)(\bar{a} \vee b)(a \vee b \vee \bar{c})(\bar{b} \vee \bar{c})$



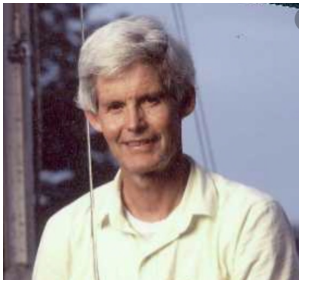
Axioms/inputs are sources

Inference rule associated with each node

Sink labelled by tautology (or Λ for refutation)

COOK'S PROGRAM FOR PROVING NP ≠ coNP

UNSAT = { f | f is an UNSAT CNF formula }



Def'n (Cook-Reckhow 1975)

An abstract proof system is a polynomial-time function A from $\{0,1\}^*$ onto the set of all UNSAT CNFs

(For proof system P, define $A(w)$ = UNSAT CNF that w refutes)

Theorem

There exists an abstract proof system in which all UNSAT CNFs have polynomial-size proofs iff NP = coNP

MOTIVATION : COOK'S PROGRAM FOR PROVING NP ≠ coNP

UNSAT = { f | f is an UNSAT CNF formula }

Def'n (Cook-Reckhow 1975)

An abstract proof system is a polynomial-time function A from $\{0,1\}^*$ onto the set of all UNSAT CNFs

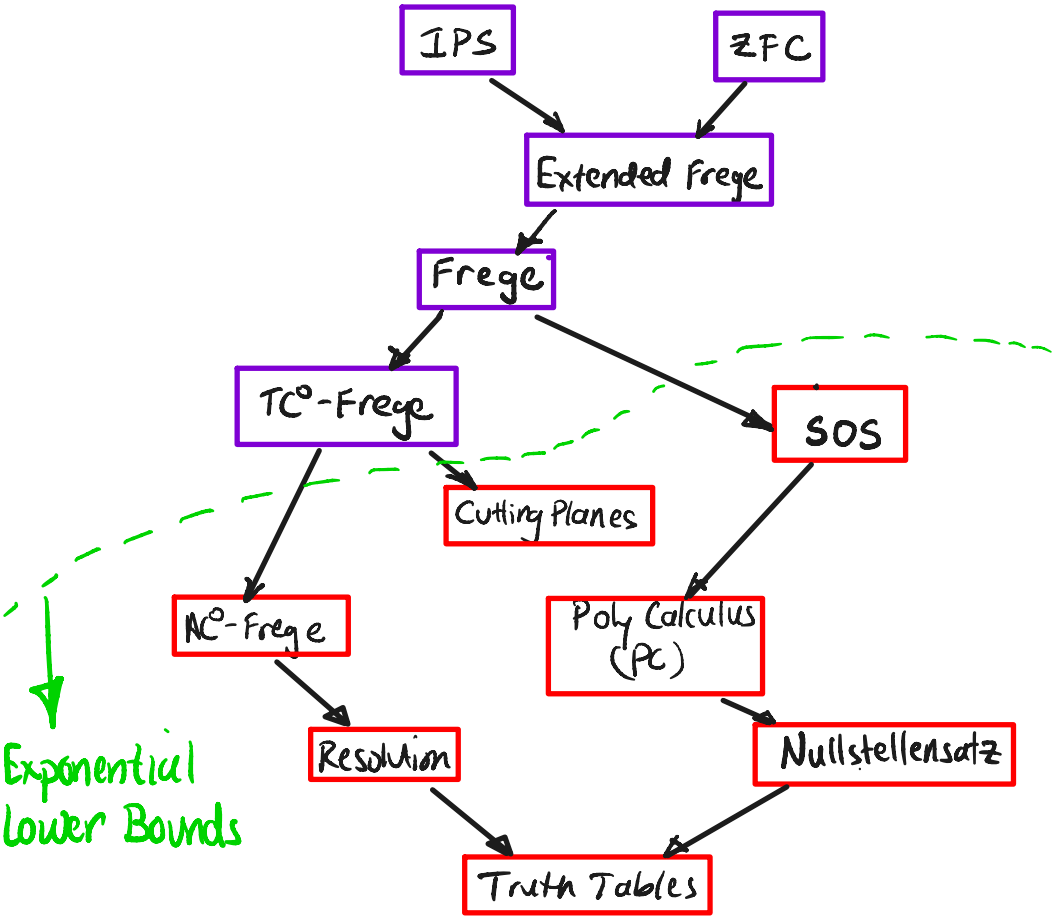
(For proof system P, define $A(w)$ = UNSAT CNF that w refutes)

Theorem

There exists an abstract proof system in which all UNSAT CNFs have polynomial-size proofs iff NP = coNP

∴ Superpolynomial Lower bounds for every proof system ⇒ P ≠ NP !

Proof Systems



MOTIVATION II : Particular proof systems closely connected to a Natural family of SAT algorithms

★ The transcript of a run of a (complete) SAT algorithm on an UNSAT formula is a proof of its UNSATISFIABILITY

Even runs of (some) SAT algorithms on satisfiable formulas yield proofs of UNSATISFIABILITY ON related formulas

Proof Complexity and Satisfiability Algorithms

- Resolution Lower Bounds proves that any Resolution-based SAT algorithm requires exponential time (worst case)
  - CDCL
- Cutting Planes Lower bounds rules out subexponential time CP-based SAT algorithms
  - branch and cut
- SOS Lower bounds rules out large family of subexponential time SAT exact and approximate algorithms
  - Large class of SDP algs
  - Extension complexity / extended formulations

MAIN QUESTIONS IN PROOF COMPLEXITY

given a particular proof system P:

- Characterize which formulas have poly size refutations
  - unconditional superpolynomial lower bounds
  - even conditional lower bounds open
- Automatizability: how hard is it to find P-refutations
- Relate P to a natural class of algorithms $\mathcal{A}(P)$
  - use lower bounds to prove limitations on exact + approximate $\mathcal{A}(P)$ algorithms for natural problems
- Compare proof strength of P to other proof systems

HARD FORMULAS ?

"It is awfully difficult to come up with even candidate hard tautologies-- there is NO such thing as tons of NP-complete problems at our disposal!"

Nearly all statements that can be expressed propositionally are either:

(1) Not true (not a tautology)

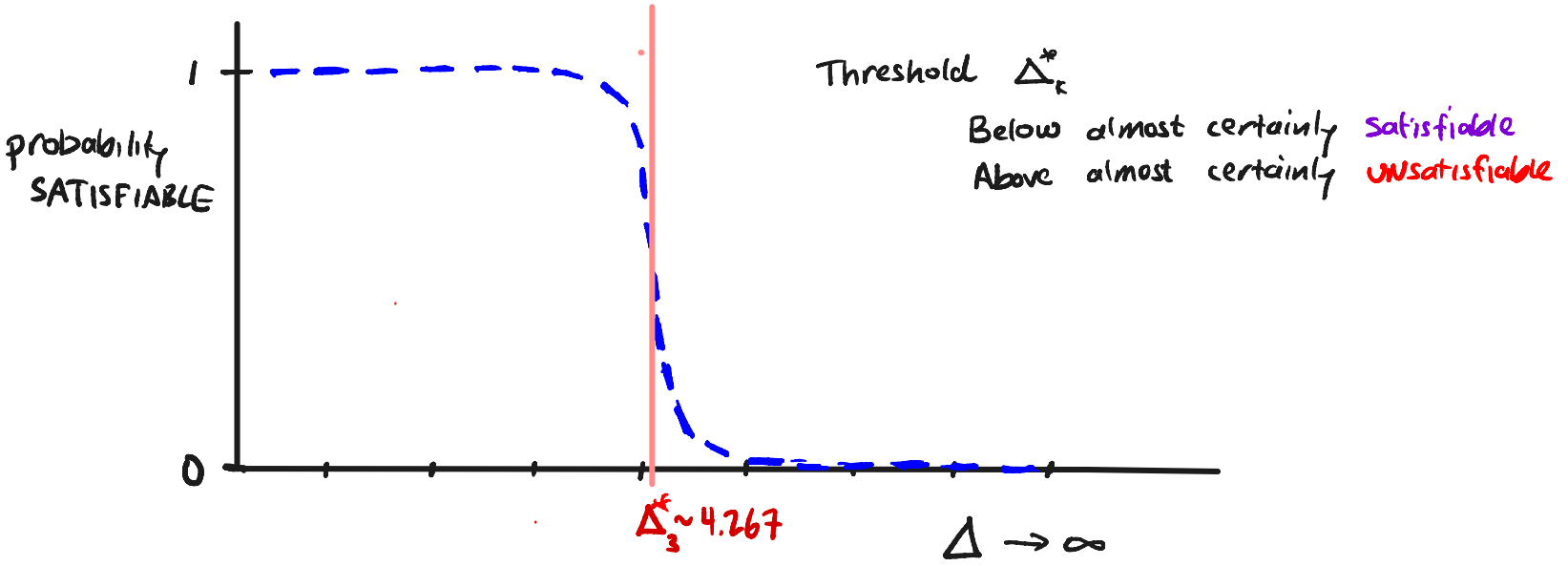
(2) Not known to be true or false

(3) Provably true (and with short Frege proof)

RANDOM K-CNFs

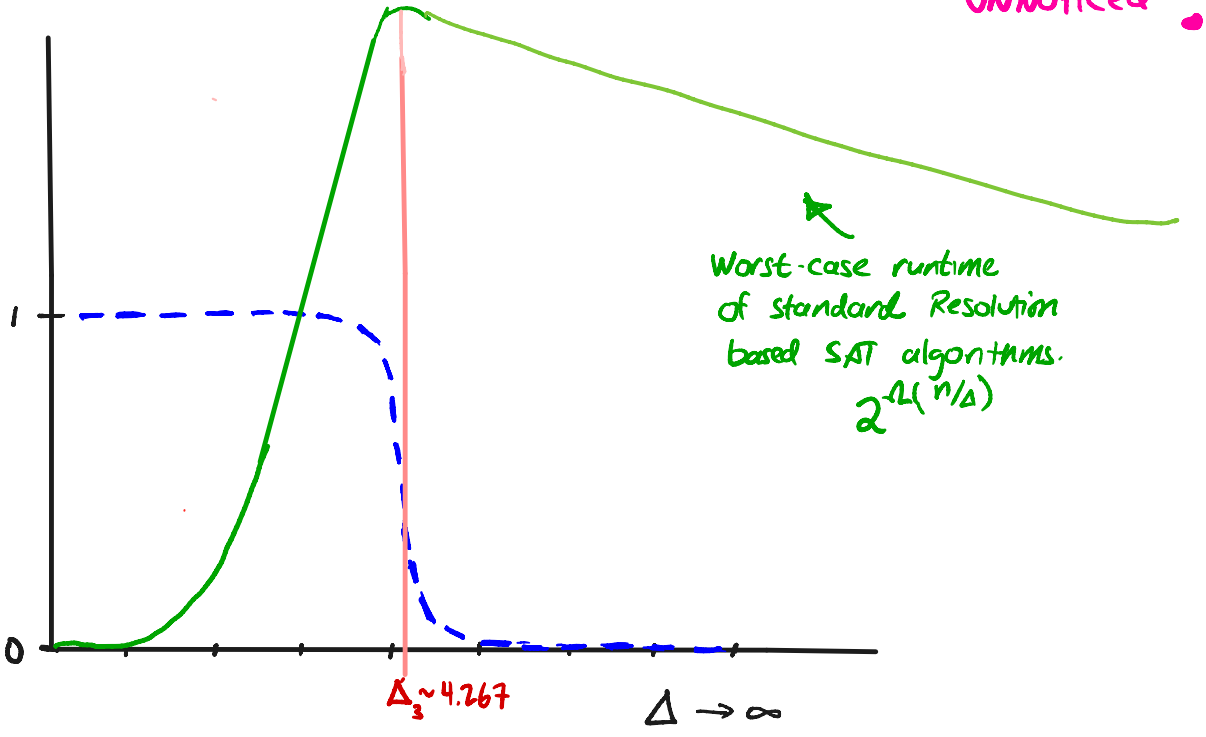
$f \sim \mathcal{F}(\Delta, n, k)$: pick $m = \Delta n$ clauses of width k

for $\Delta > 0$ suff large $f \sim \mathcal{F}(\Delta, n, k)$ UNSAT w.h.p.



Resolution-Based Algorithms for random KCNF

passing transition point Δ_k^* goes absolutely unnoticed !



OTHER RANDOM CNF FAMILIES

1. Random $kXOR_1$, random kCSP

2. $Clique_g(k)$, $g \sim \mathcal{G}(n,p)$ $p \sim n^{-2/k-1}$

   $p = 1/2$ Clique of size $\log n$

3. $Hard_f(s)$ $f \sim$ all boolean functions on n variables

   Says f computed by a size s circuit

MOTIVATION

1. Structural properties relate to our understanding

2. Natural distributions as benchmark for SAT algorithms

3. Lower bounds for particular proof systems (RES, SOS) give unconditional inapproximability for large family of algorithms

WHY IS IT SO HARD TO CERTIFY UNSAT OF RANDOM f?

① "Counting orguments don't count anymore" - Razborov

Circuit complexity:

$2^{poly(n)}$ circuits of poly size $\ll$ 2^{2^n} Boolean functions

Proof complexity

# of proofs of size s $\approx$ # UNSAT formulas

FEIGE'S HYPOTHESIS

Defn (Refutation algorithm)

Algorithm A is a refutation algorithm for random KSAT, $f \sim \mathcal{F}(\Delta, n, k)$:

A outputs YES with probability $> \frac{1}{2}$

A outputs NO if $\mathcal{Q}$ is satisfiable

Feige's Hypothesis: For $\Delta > 0$ sufficiently large, $f \sim \mathcal{F}(\Delta, n, k)$:

I. there is NO polytime refutation algorithm of f

II. NO proof system can efficiently refute f

The incredible usefulness of Feige's conjecture

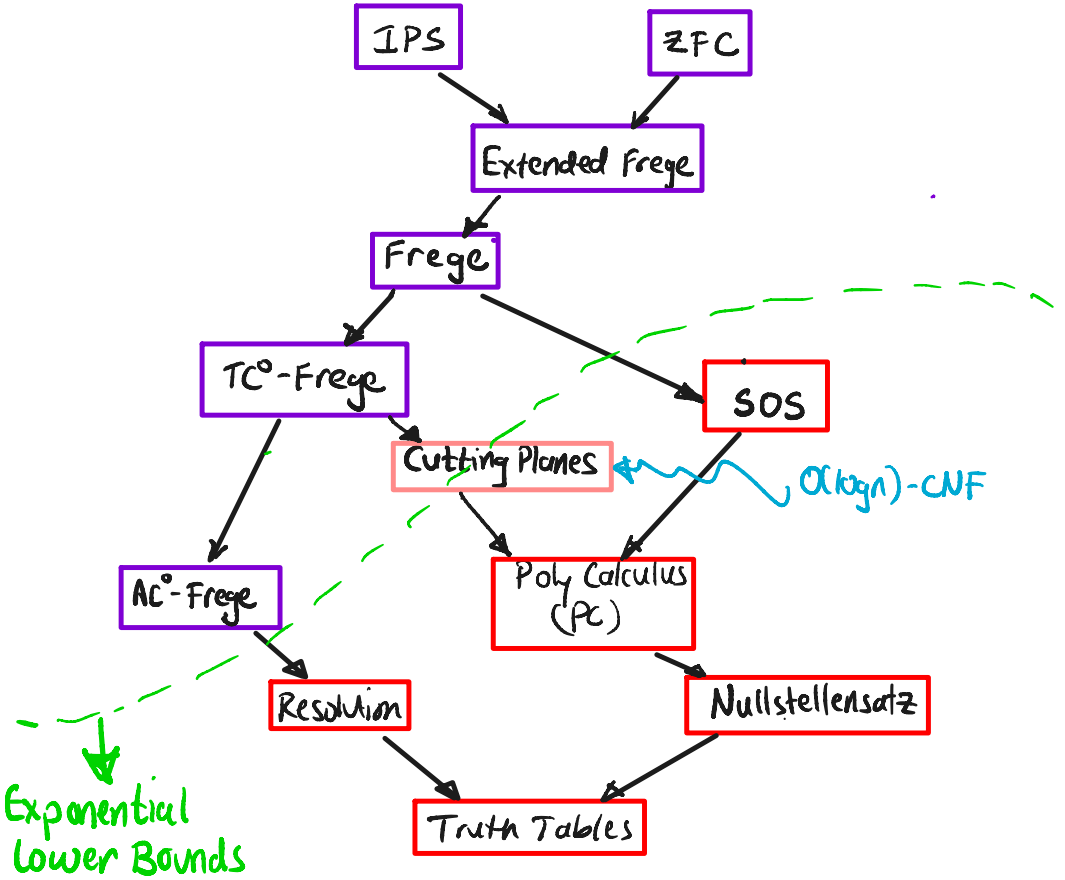
Many problems are hard under Feige's conjecture:

- Approximating vertex cover
- Avg case MCSP
- PAC learning DNF

UPPER BOUNDS FOR RANDOM SAT

	Poly-size UB
Resolution	$m > n^2/\log n$ [Beame, Kautz, Pi, Saks] ($n^{k-1}/\log n$)
TC^0 Frege	$m \sim n^{1.4}$ [Feige, Kim, Ofek] [Müller, Tzameret]

LOWER BOUNDS FOR RANDOM SAT



LOWER BOUNDS FOR RANDOM SAT

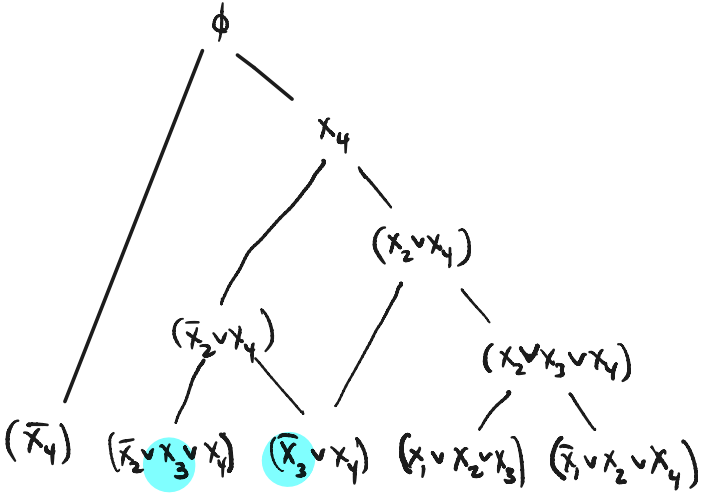
	Poly-size UB	Expontial LB
Resolution	$m > n^2/\log n$ [Beame, Karp, P, Saks]	$m < n^{1.5}$ [Chvatal, Szemeredi] [Beame, Karp, P, Saks] [Ben-Sasson, Wigderson]
Nullsatz		$m = O(n)$ [Grigoriev]
Poly Calculus		$m = O(n)$ [Buss, Grigoriev, Impagliazzo, P]
SOS		$m = O(n)$ [Gresoriev, Schoenebeck]
Cutting Planes		$k = \Theta(\log n)$ $m = \text{poly}(n)$ [Fleming, Pankratov, P, Robere/Hrubes, Pudlak]
TC^0 Frege	$m \sim n^{1.4}$ [Feige, Kim, Ofek] [Müller, Tzameret]	?

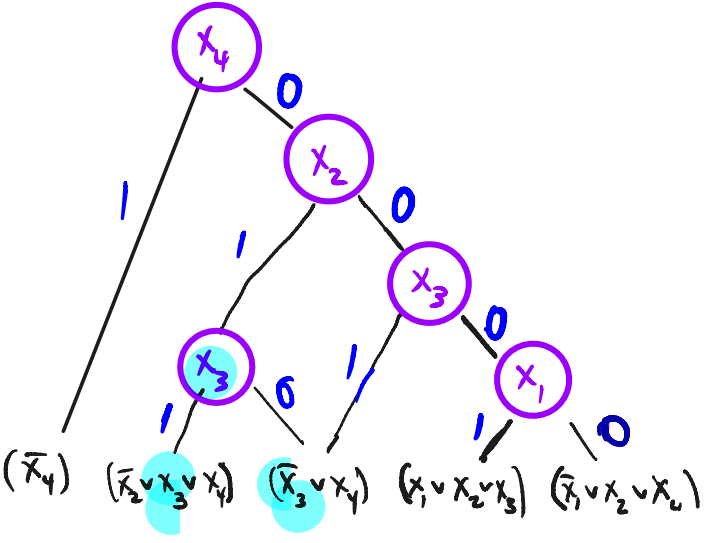
LOWER BOUNDS FOR RANDOM SAT

	Poly-size UB	Expontial LB
Resolution	$m > n^2/\log n$ [Beame, Karp, P, Saks]	$m < n^{1.5}$ [Chvatal Szemeredi] [Beame, Karp, P, Saks] [Ben-Sasson, Wigderson]
Nullsatz		$m = O(n)$ [Grigoriev]
Poly Calculus		$m = O(n)$ [Buss, Grigoriev, Impagliazzo, P]
SOS		$m = O(n)$ [Gresoriev, Schoenebeck]
Cutting Planes		$k = \Theta(\log n)$ $m = \text{poly}(n)$ [Fleming, Pankratov, P, Robere/Hrubes, Pudlak]
TC^0 Frege	$m \sim n^{1.4}$ [Feige, Kim, Ofek] [Müller, Tzameret]	?

TREE RESOLUTION ≡ DECISION TREE FOR SEARCH

Search(f): given assignment $\alpha \in \{0,1\}^n$, output some $C_i \in f$ falsified by α





TOP-DOWN VIEW OF (DAG-LIKE) RESOLUTION?

First try:

Tree-like Resolution proof of f $\approx$ Decision tree for Search(f)

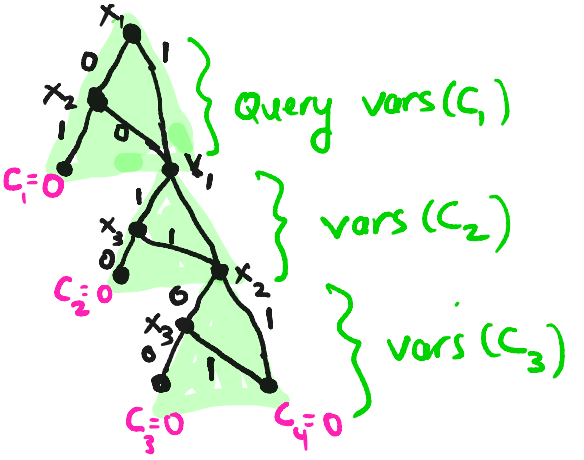
Dag-like Resolution proof of f $\overset{?}{\approx}$ Branching program for Search(f)

Resolution proof of f	$\neq$	Branching program for Search(f)

Claim: For every UNSAT f, Search(f) has a linear-size branching program

Example:

$$f = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_3)(x_2 \vee x_3)(\bar{x}_3)$$



TOP-DOWN VIEW OF RESOLUTION

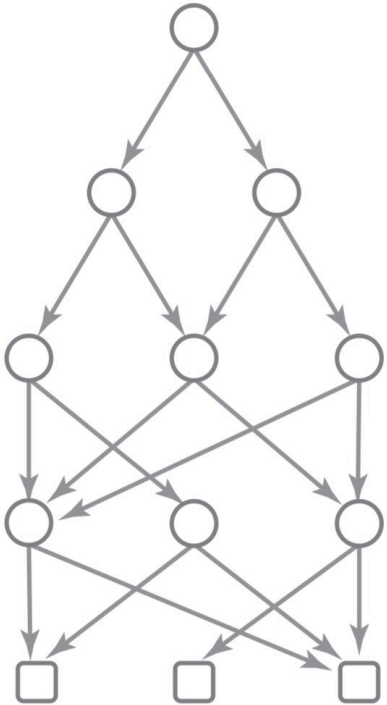
[Raz95]

second try:

Each dag node v is labeled with a conjunction $C_v : \{0,1\}^n \to \{0,1\}$

- *root r:* $C_r \equiv 1$ (constant 1)
- *node v with children u, u':*

  $C_v^{-1}(1) \subseteq C_u^{-1}(1) \cup C_{u'}^{-1}(1)$
- *leaf v:* Labeled with solution to Search(f)



RESOLUTION LOWER BOUNDS

1. For $f \sim \mathcal{H}(\Delta, n, k)$ any Resolution dag requires Linear width.

2. Ben-Sasson, Wigderson:

   Small size $\Rightarrow$ small width

   s $\qquad$ $\sqrt{n \log s}$

RESOLUTION LOWER BOUNDS

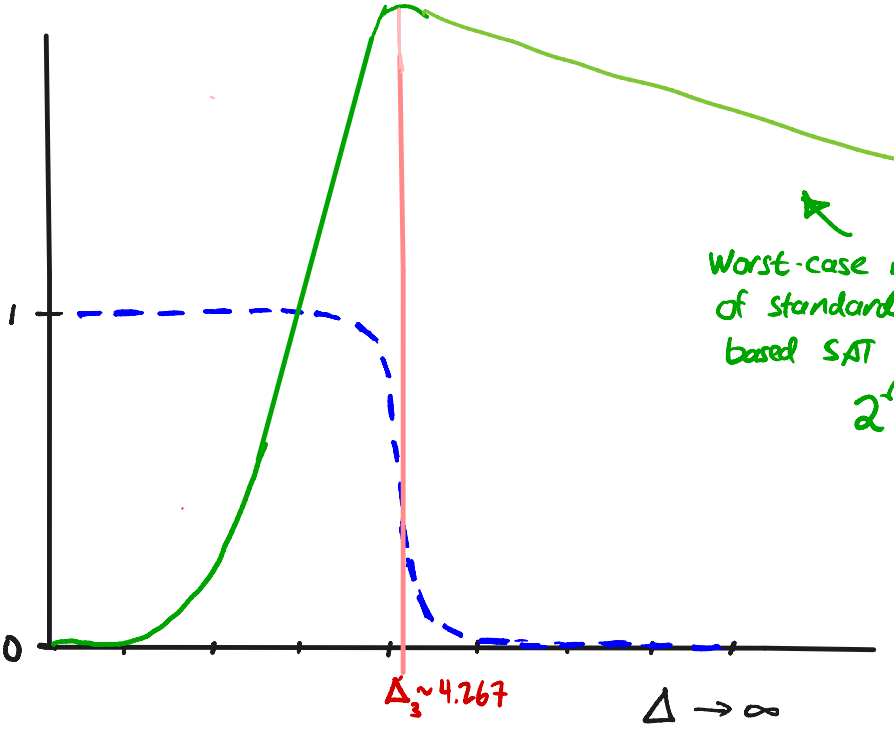
Lemma For $f \sim \mathcal{G}(\Delta, n, k)$ any Resolution dag requires **Linear width**.

Proof sketch

- With high probability, the clause-variable graph has high (boundary) expansion
- Let π be a Resolution refutation of f.
  Find clause C^* derived from between $\frac{n}{3}$ and $\frac{2n}{3}$ initial clauses
  By boundary expansion, C^* must contain Ωn variables

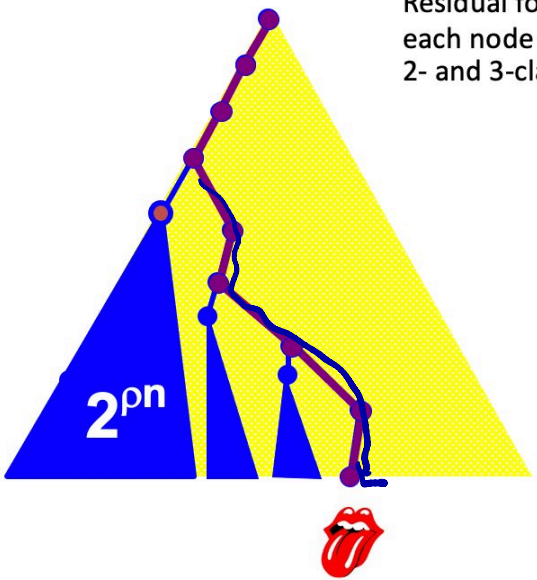
Resolution-Based Algorithms for random KCNF

passing transition point Δ_k^* goes absolutely unnoticed !



Worst-case runtime of standard Resolution based SAT algorithms.
$2^{\Omega(n/\Delta)}$

Long-running DPLL Executions and Proof Complexity



Residual formula at each node is a mix of 2- and 3-clauses

Residual formula at ● is UNSAT but requires exponential size Resolution proofs

[Achlioptas, Beame, Molloy]

LOWER BOUNDS FOR RANDOM SAT

	Poly-size UB	Expontial LB
Resolution	$m > n^2/\log n$ [Beame, Karp, P, Saks]	$m < n^{1.5}$ [Chvatal, Szemeredi] [Beame, Karp, P, Saks] [Ben-Sasson, Wigderson]
Nullsatz		$m = O(n)$ [Grigoriev]
Poly Calculus		$m = O(n)$ [Buss, Grigoriev, Impagliazzo, P]
SOS		$m = O(n)$ [Gresoriev, Schoenebeck]
Cutting Planes		$k = \Theta(\log n)$ $m = \text{poly}(n)$ [Fleming, Pankratov, P, Robere/Hrubes, Pudlak]
TC^0 Frege	$m \sim n^{1.4}$ [Feige, Kim, Ofek] [Müller, Tzameret]	?

CUTTING PLANES

- Lines in proof are linear inequalities $\sum_{i=1}^{n} a_i x_i \geq b$ (weights)
- Weights wlog have length $O(n^2)$
- CP* : weights have length polylogn
- For today we will focus on CP* lower bounds

Cutting Planes rules

- addition:

$$\frac{\begin{array}{l} a_1x_1 + \ldots + a_nx_n \geq A \\ b_1x_1 + \ldots + b_nx_n \geq B \end{array}}{(a_1+b_1)x_1+\ldots+(a_n+b_n)x_n \geq A+B}$$

- multiplication by positive integer:

$$\frac{a_1x_1 + \ldots + a_nx_n \geq A}{ca_1x_1 + \ldots + ca_nx_n \geq cA}$$

- **Division** by positive integer:

$$\frac{ca_1x + \ldots + ca_nx_n \geq B}{a_1x_1 + \ldots + a_nx_n \geq \lceil B/c \rceil}$$

Cutting Planes Lower Bounds via Interpolation

Theorem [Bonet-P-Ruiz, Razborov, Pudlak; Krajicek]

Exponential lower bounds for $\text{Clique}(x,z) \wedge \text{Color}(y,z)$

- x-vars correspond to k-cliques
- y-vars correspond to $(k-1)$-colorings

Pf uses feasible interpolation property of CP: any size s CP refutation of $\text{Clique} \wedge \text{Color} \Rightarrow$ size s monotone circuit for clique/coclique

CUTTING PLANES AND RANDOM FORMULAS

- Cannot directly use feasible interpolation
  (random formulas not of correct form)

- Still we can use ideas indirectly -- since random formulas have the same combinatorial properties when we look under the hood

CUTTING PLANES AND RANDOM FORMULAS





Theorem [Fleming, P, Pankratov, Robere; Hrubes, Pudlak]

Random $\Theta(\log n)$ CNFs above threshold require $2^{\tilde{\Omega}(n)}$ size Cutting Planes refutations

PROOF OVERVIEW

Let $f \sim \mathcal{F}(\Delta, 2n, k)$ $\quad k = O(\log n)$, Δ suff large

Randomly partition the $2n$ variables of f into 2 equal-sized parts $\vec{x}$ and $\vec{y}$

Today: we'll assume all clauses of f are roughly balanced.

$$f = (\bar{x}_1 \vee \bar{x}_3 \vee x_5 \vee \bar{y}_2 \vee y_7 \vee y_8)(\bar{x}_1 \vee x_2 \vee x_5 \vee y_1 \vee \bar{y}_3 \vee y_5)(x_2 \vee x_3 \vee \bar{x}_4 \vee \bar{y}_6 \vee \bar{y}_7 \vee \bar{y}_k) \cdots$$

PROOF OVERVIEW

Let $f \sim \mathcal{F}(\Delta, an, k)$ $\quad k = O(\log n)$, Δ suff large

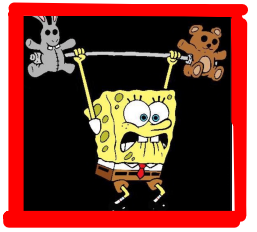
Today: we'll assume all clauses of f are roughly balanced.

$$f = (\bar{x}_1 \vee \bar{x}_3 \vee x_5 \vee \bar{y}_2 \vee y_7 \vee y_8)(\bar{x}_1 \vee x_2 \vee x_5 \vee y_1 \vee \bar{y}_3 \vee y_5)(x_2 \vee x_3 \vee \bar{x}_4 \vee \bar{y}_6 \vee \bar{y}_7 \vee \bar{y}_k) \cdots$$

Search(f):

$\alpha = 11001011$ assignment to $\vec{x}$ vars

output: clause of f falsified



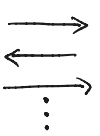
$\beta = 11000110$ assignment to $\vec{y}$ variables

PROOF OVERVIEW

$f = (\bar{x}_1 \vee \bar{x}_3 \vee x_5 \vee \bar{y}_2 \vee y_7 \vee y_8)(\bar{x}_1 \vee x_2 \vee x_5 \vee y_1 \vee \bar{y}_3 \vee y_5)(x_2 \vee x_3 \vee \bar{x}_4 \vee \bar{y}_6 \vee \bar{y}_7 \vee \bar{y}_8) \cdots$

Search(f):

$\alpha = 110000$



$\beta = 001110$

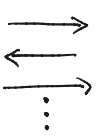
① CP refutation of f $\Rightarrow$ dag-like communication protocol for Search(f)

PROOF OVERVIEW

$f = (\bar{x}_1 \vee \bar{x}_3 \vee x_5 \vee \bar{y}_2 \vee y_7 \vee y_8)(\bar{x}_1 \vee x_2 \vee x_5 \vee y_1 \vee \bar{y}_3 \vee y_5)(x_2 \vee x_3 \vee \bar{x}_4 \vee \bar{y}_6 \vee \bar{y}_7 \vee \bar{y}_8) \cdots$

Search(f):

$\alpha = 110000$



$\beta = 001110$

① CP refutation of f ⟹ dag-like communication protocol for Search(f)

② Dag-like cc of Search(f) ➡ Dag-like cc of mKW(CSP_f)
➡ monotone circuit complexity of CSP_f

③ Show monotone circuit complexity of CSP_f is $2^{\Omega(n)}$

① CP refutation ⇒ DAG-LIKE CC Protocol for Search(f)

[Raz95]

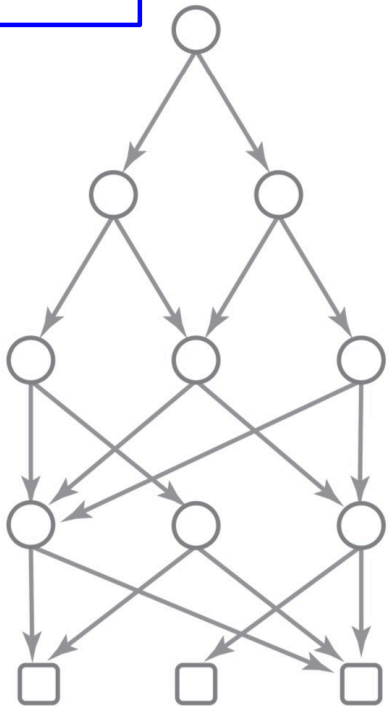
Each dag node v is labeled with a Rectangle $R_v \subseteq \{0,1\}^n \times \{0,1\}^n$

- *root* r: $R_r \equiv \{0,1\}^n \times \{0,1\}^n$
- *node* v *with children* u, u':

$$R_v \subseteq R_u \cup R_{u'}$$

- *leaf* v: R_v is monochromatic

(∃ clause $C_i \in f$ such that all $(\alpha, \beta) \in R_v$ falsify C_i)



① CP refutation ⇒ DAG-LIKE CC Protocol for Search(f)

Theorem

size S (semantic) CP^* $\equiv$ size $poly(S)$ CC-DAG for Search(f)

Pf sketch ($\Rightarrow$)

Fix a CP^* refutation Π.

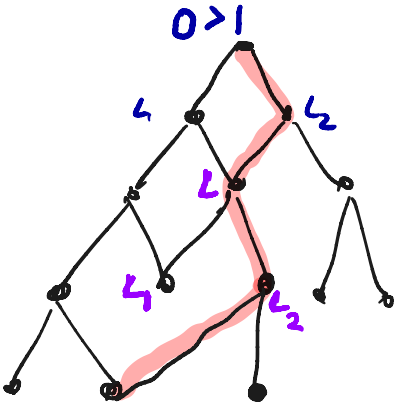
Alice and Bob on inputs α, β take a walk down Π from root to a leaf

Invariant: (α, β) falsifies every line in path

By Soundness if $L_1, L_2 \Rightarrow L$ then

(α, β) falsify $L \Rightarrow (\alpha, \beta)$ falsify either L_1 or L_2

Players can evaluate each line by low cc protocol



② DAG-like CC for Search(f) $\equiv$ DAG-Like CC for mKW(CSP_f)
$\equiv$ monotone circuit for CSP_f

CNF search problem Search(f): f a CNF over $X_1 \cdots X_n, Y_1 \cdots Y_n$

INPUT: $\alpha \in \{0,1\}^n$ $\beta \in \{0,1\}^n$

OUTPUT: a clause C_i of f falsified by (α, β)

mKW(F) Search Problem: $F : \{0,1\}^n \to \{0,1\}$ a monotone function

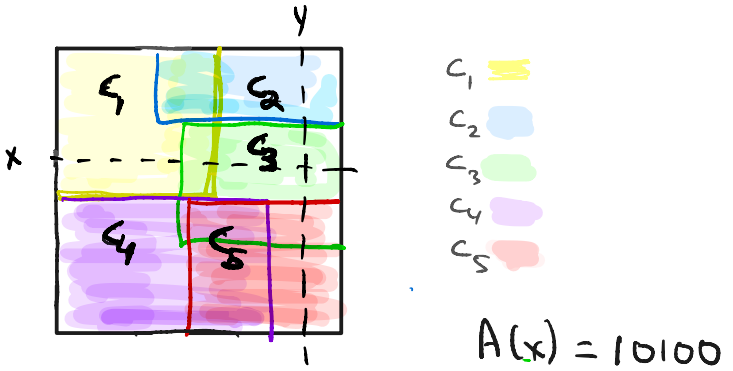
INPUT: $x \in F^{-1}(1)$, $y \in F^{-1}(0)$

OUTPUT: a coordinate $i \in [n]$ such that $x_i > y_i$

② DAG-like CC for Search(f) ≡ DAG-Like CC for mKW(CSP_f)
≡ monotone circuit for CSP_f

goal: Given $f(\vec{x},\vec{y})$, construct a monotone function CSP_f such that Search(f) = mKW(CSP_f)

Search(f) induces a rectangle cover of CC matrix



A(x) = 10100
B(y) = 10010

Define CSP_f by sets:

$\mathcal{A} = \{A(\alpha) \in \{0,1\}^m \mid \alpha \in \{0,1\}^n\}$ of accepting inputs

$\mathcal{B} = \{B(\beta) \in \{0,1\}^m \mid \beta \in \{0,1\}^n\}$ of rejecting inputs

For $\alpha \in \{0,1\}^n$ $A(\alpha)_i \stackrel{d}{=} 1 \iff \alpha \in R_i$

For $\beta \in \{0,1\}^n$ $B(\beta)_i \stackrel{d}{=} 0 \iff \beta \in R_i$

③ Monotone circuit Lower Bounds for CSP_f

Method of Approximations [Jukna, Berg-Ulfberg, Razborov]

Any monotone circuit separating A from B such that

① $|A|$, $|B|$ are exponentially large

② A and B are sufficiently spread

must be of exponential size

Spread: Set system A is δ-spread if $\forall I \subseteq [m]$

$$|\{a \in A \mid a_i = 1 \ \forall i \in I\}| \leq \delta^{|I|} |A|$$

B is δ-spread if $\forall I \subseteq [m]$ $|\{b \in B \mid b_i = 0 \ \forall i \in I\}| \leq \delta^{|I|} |B|$

$f = (C_1^x \vee C_1^y)(C_2^x \vee C_2^y) \cdots (C_m^x \vee C_m^y)$

f balanced $\Rightarrow$ each clause has same number of $\vec{x}$ vars + $\vec{y}$ vars

① $|A|, |B|$ large:

recall $A = \{ A(\alpha) \mid \alpha \in \{0,1\}^n \}$
$B = \{ B(\beta) \mid \beta \in \{0,1\}^n \}$

A, B are nearly 1-1 so $|A|, |B| \approx 2^n$

② show A, B are well-spread

follows since $F_x = (C_1^x)(C_2^x) \cdots (C_m^x)$
$F_y = (C_1^y)(C_2^y) \cdots (C_m^y)$

are expanding.

Open Problems

1. CP lower bounds for random kCNF $k = O(1)$

2. Frege lower bounds for random kCNF
   conditional lower bounds?

3. Reverse mathematics: completeness of random CNF
   AC^0-Frege + random-UNSAT p-simulates Frege?

4. Other hard random distributions?
   - PRG formulas
   - Circuit lower bound formulas


Thanks!

A Related Conjecture about Random UNSAT formulas

$f \sim \mathcal{H}(n,s)$: Choose a random Boolean function $\alpha \in \{0,1\}^{2^n}$ on n inputs.

Variables of CNF formula f:
$\vec{c} = c_1 \dots c_{poly(s)}$ describe encoding of a polysize circuit on n inputs
f states : Circuit computed by $\vec{c}$ computes α

Rudich Conjecture There are NO polysized refutations of $f \sim \mathcal{H}(n,s)$ in any proof system. (for $s = poly(n)$)

f not balanced

Need to find a partition of variables (X, Y) so we can carry out previous argument

For any partition some clauses will be highly imbalanced 

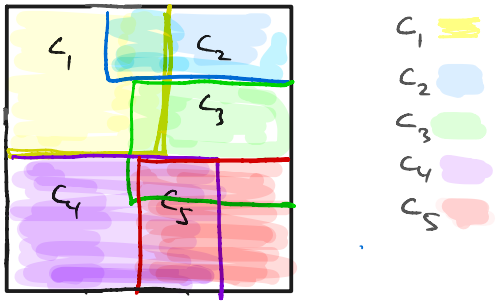
Solution Show whp a random partition satisfies

- Most clauses are nearly balanced
- There is a large collection of assignments $X' \times Y' \subseteq \{0,1\}^n \times \{0,1\}^n$ that satisfy all imbalanced clauses
- Repeat previous argument restricted to X', Y'

② DAG-like CC for Search(f) $\equiv$ DAG-like CC for mKW(CSP_f)
$\equiv$ monotone circuit for CSP_f

Goal: Given $f(\vec{x}, \vec{y})$, construct a monotone function CSP_f such that Search(f) = mKW(CSP_f)

Search(f) induces a rectangle cover of CC matrix



Define CSP_f by sets:

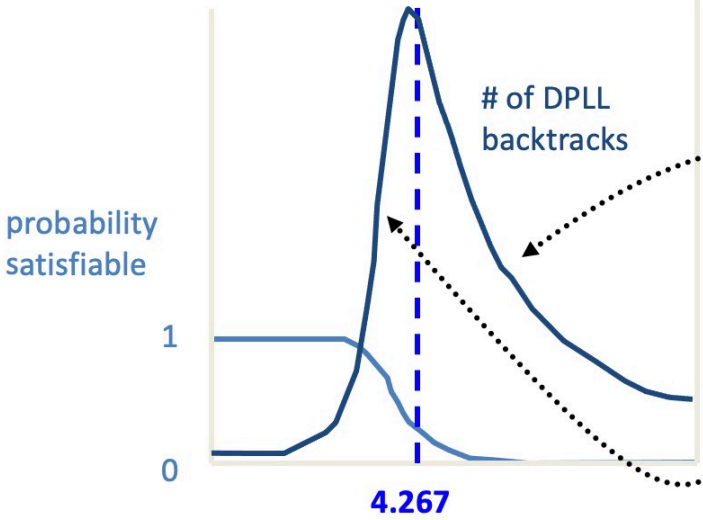
$\mathcal{A} = \{A(\alpha) \in \{0,1\}^m \mid \alpha \in \{0,1\}^n\}$ of accepting inputs

$\mathcal{B} = \{B(\beta) \in \{0,1\}^m \mid \beta \in \{0,1\}^n\}$ of rejecting inputs

For $\alpha \in \{0,1\}^n$ $A(\alpha)_i \stackrel{d}{=} 1 \iff \alpha \in R_i$

For $\beta \in \{0,1\}^n$ $B(\beta)_i \stackrel{d}{=} 1 \iff \beta \in R_i$

DPLL on random 3-CNF*



Proof complexity shows $2^{\Theta(n/r)}$ time is required for unsatisfiable formulas for $r > r_3$*

[B,Karp,Saks,Pitassi 98]
[Ben-Sasson 02]

What about satisfiable formulas below threshold?

[Mitchell,Selman,Levesque 92]

* n = 50 variables