MaxSAT Resolution and SubCube Sums

Meena Mahajan

The Institute of Mathematical Sciences,
Homi Bhabha National Institute,
Chennai, India.

Joint work with Yuval Filmus, Gaurav Sood, Marc Vinyals

Satisfiability: Theory, Practice, and Beyond — Reunion
14-17 June 2022

Resolution

- The rule: From $(C \vee x)$ and $(D \vee \neg x)$, infer $C \vee D$.
- A refutation of a propositional CNF formula F:
  A sequence $C_1, C_2, \ldots, C_t$ where
  $C_t = \Box$, and
  for each $i \in [t]$, either $C_i \in F$ or
  C_i is inferred from C_j, C_k for some $j, k < i$.
- Sequence $F_0, F_1, F_2, \ldots F_t$ where
  $F_0 = F$, and for each $i \in [t]$, $F_i = F_{i-1} \cup \{C_i\}$.
- The invariant: Every assignment satisfying F_{i-1} also satisfies F_i.
- $\Box \in F_t$, so no assignment satisfies F_t, so F_t unsat, so $F_0 = F$ unsat.

Certifying MaxSAT values

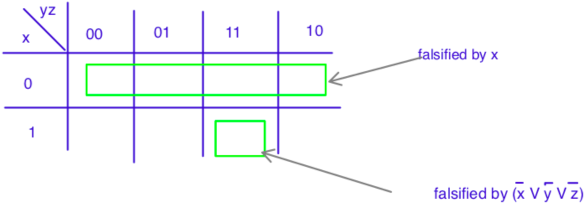
- For CNF formula F, number k,
  Goal: show that every assignment falsifes at least k clauses.
- Produce a sequence $F_0, F_1, \ldots, F_t$ of multisets of clauses.
- Desired invariant: For every assignment α, number of clauses falsified in F_{i-1} equals number of clauses falsified in F_i.

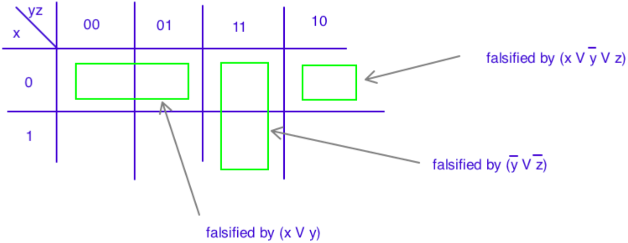
$$\text{viol}_{F_{i-1}}(\alpha) = \text{viol}_{F_i}(\alpha).$$

- Desired target: F_t has at least k copies of $\Box$.
- Resolution does not maintain this invariant.
  The MaxSAT resolution rule, [BonetLevyManyá 2007], does.

The MaxSat Resolution Rule

Rearrange cubes of falsifying assignments.





The MaxSat Resolution Rule

$$\frac{\begin{array}{ll} x \vee a_1 \vee \ldots \vee a_s & (x \vee A) \\ \overline{x} \vee b_1 \vee \ldots \vee b_t & (\overline{x} \vee B) \end{array}}{\begin{array}{ll} a_1 \vee \ldots \vee a_s \vee b_1 \vee \ldots \vee b_t & \text{(the ``standard resolvent'')} \end{array}}$$

$$\left.\begin{array}{l} x \vee A \vee \overline{b}_1 \\ x \vee A \vee b_1 \vee \overline{b}_2 \\ \vdots \\ x \vee A \vee b_1 \vee \ldots \vee b_{t-1} \vee \overline{b}_t \end{array}\right\} \quad \text{(weakenings of } x \vee A\text{)}$$

$$\left.\begin{array}{l} \overline{x} \vee B \vee \overline{a}_1 \\ \overline{x} \vee B \vee a_1 \vee \overline{a}_2 \\ \vdots \\ \overline{x} \vee B \vee a_1 \vee \ldots \vee a_{s-1} \vee \overline{a}_s \end{array}\right\} \quad \text{(weakenings of } \overline{x} \vee B\text{)}$$

A MaxSat Resolution derivation

- A sequence $F_0, F_1, \ldots, F_t$ of **multisets** of clauses.
- $F_0 = F$.
- For $i \in [t]$, F_i obtained from F_{i-1} by applying MaxSAT resolution rule, **replacing** the antecedents by the consequents.

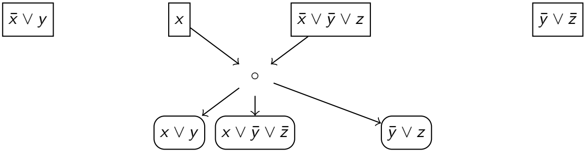
A MaxSat Resolution derivation

- A sequence $F_0, F_1, \ldots, F_t$ of **multisets** of clauses.
- $F_0 = F$.
- For $i \in [t]$, F_i obtained from F_{i-1} by applying MaxSAT resolution rule, **replacing** the antecedents by the consequents.

$\boxed{\bar{x} \vee y}$ $\boxed{x}$ $\boxed{\bar{x} \vee \bar{y} \vee z}$ $\boxed{\bar{y} \vee \bar{z}}$

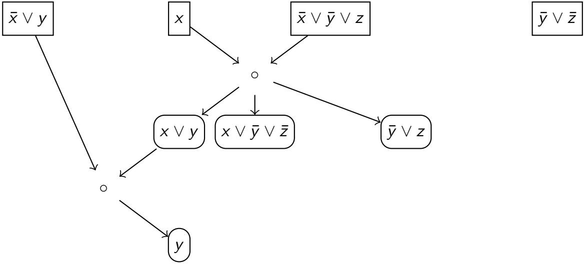
A MaxSat Resolution derivation

- A sequence $F_0, F_1, \ldots, F_t$ of **multisets** of clauses.
- $F_0 = F$.
- For $i \in [t]$, F_i obtained from F_{i-1} by applying MaxSAT resolution rule, **replacing** the antecedents by the consequents.



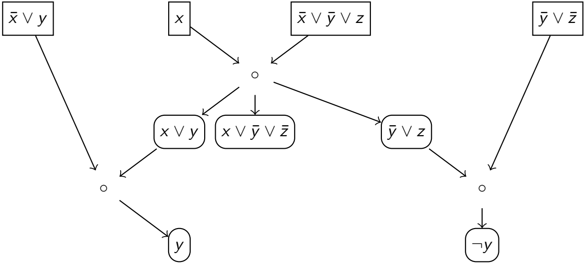
A MaxSat Resolution derivation

- A sequence $F_0, F_1, \ldots, F_t$ of **multisets** of clauses.
- $F_0 = F$.
- For $i \in [t]$, F_i obtained from F_{i-1} by applying MaxSAT resolution rule, **replacing** the antecedents by the consequents.



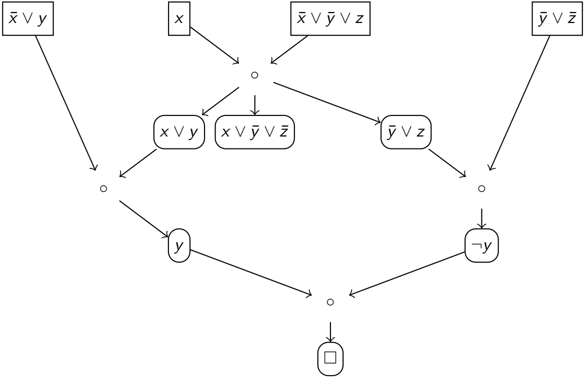
A MaxSat Resolution derivation

- A sequence $F_0, F_1, \ldots, F_t$ of **multisets** of clauses.
- $F_0 = F$.
- For $i \in [t]$, F_i obtained from F_{i-1} by applying MaxSAT resolution rule, **replacing** the antecedents by the consequents.



A MaxSat Resolution derivation

- A sequence $F_0, F_1, \ldots, F_t$ of **multisets** of clauses.
- $F_0 = F$.
- For $i \in [t]$, F_i obtained from F_{i-1} by applying MaxSAT resolution rule, **replacing** the antecedents by the consequents.



The MaxSat Weakening Rule

- Recall invariant to be maintained:
  number of falsified clauses preserved.
- Usual weakening not sound. Instead:
- Replace a clause A by the two clauses $A \vee x$ and $A \vee \overline{x}$.

The MaxSat Weakening Rule

- Recall invariant to be maintained:
  number of falsified clauses preserved.
- Usual weakening not sound. Instead:
- Replace a clause A by the two clauses $A \vee x$ and $A \vee \overline{x}$.
- Note: derivations are **reversible**.
  From F_i we can obtain F_{i-1} through a sequence of MaxSAT resolution and MaxSAT weakening rules.

MaxSAT Resolution

- MaxSAT Resolution sound and complete for certifying MaxSAT value. [BonetLevyManyá 2007].
- In practice, MaxSAT solvers don't really use this rule directly.
- So why is it interesting?

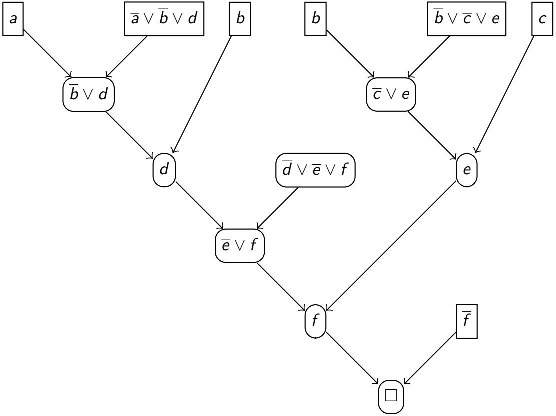
Certifying Unsatisfiability

- Resolution can certify unsatisfiability.
- Using MaxSAT Resolution – overkill?
- Interesting things can happen if we do preprocessing.
  - Encode F into dualRailHorn F'; then $\mathrm{maxSAT}(F') \geq n$; and F sat iff $\mathrm{maxSAT}(F') \leq n$.
  - weighted DualRailMaxSAT p-simulates general Resolution. [Bonet,Buss,Ignatiev,Marques-Silvao 2018]:
- Interesting things can happen if we allow arbitrary positive weights: It simulates Resolution and is equivalent to Circular Resolution. [BonetLevy 2020].
- Interesting things can happen if we allow negative weights and virtual creation – add (A, w) and $(A, -w)$ to the current multiset. It simulates Resolution and is equivalent to Circular Resolution. [LarrossaRollon 2020].
- Understanding unweighted MaxSAT resolution and weakening better can help lead to more such extensions.

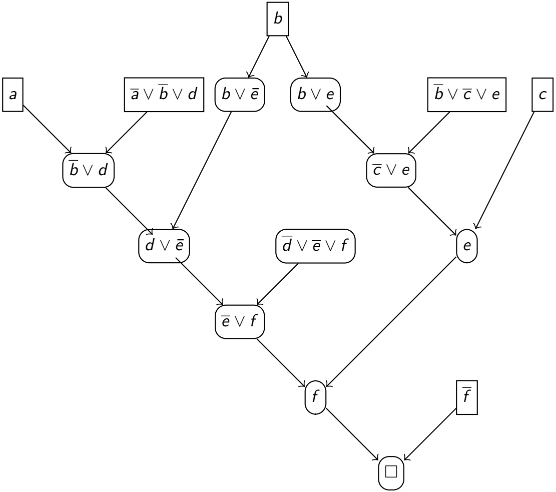
MaxRes, MaxResW

- We consider two proof systems for certifying unsatisfiability:
  MaxRes: only MaxSAT Resolution, and
  MaxResW: MaxSAT Resolution and MaxSAT Weakening.
  (All rules unweighted)
- Sound – invariant maintained at each stage
- Complete – because complete even for certifying MaxSAT value
- p-simulated by Resolution: add instead of replace clauses.

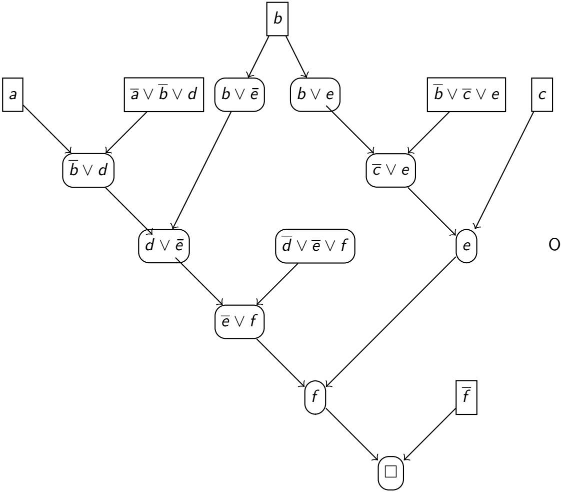
MaxResW p-simulates TreeRes



MaxResW p-simulates TreeRes



MaxResW p-simulates TreeRes



Open: Is the weakening rule really necessary?

MaxResW better than TreeRes

Theorem

TreeRes does not simulate MaxRes

- Pebbling formulas on single-sink DAGs: easy in TreeRes
- Compose with OR_2: hard for TreeRes on Pyramid Graphs [Ben-SassonImpagliazzoWigderson 2004].

MaxResW better than TreeRes

Theorem

TreeRes does not simulate MaxRes

- Pebbling formulas on single-sink DAGs: easy in TreeRes
- Compose with OR_2: hard for TreeRes on Pyramid Graphs [Ben-SassonImpagliazzoWigderson 2004].
- Composed formula easy in MaxRes/MaxResW?? We don't know.

MaxResW better than TreeRes

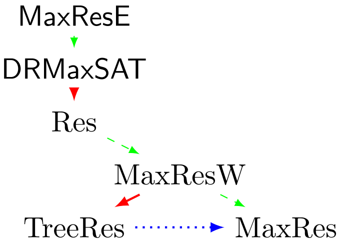
Theorem

TreeRes does not simulate MaxRes

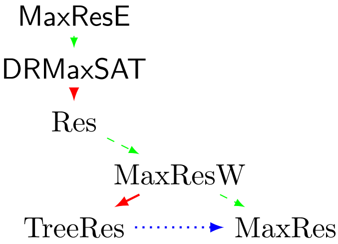
- Pebbling formulas on single-sink DAGs: easy in TreeRes
- Compose with OR_2: hard for TreeRes on Pyramid Graphs [Ben-SassonImpagliazzoWigderson 2004].
- Composed formula easy in MaxRes/MaxResW?? We don't know.
- Tweak the composed formulas — add some *hint* clauses.
- Show: now short MaxRes refutation.
- Show: still hard for TreeRes. use game, 1-query complexity, pebbling

$$\begin{aligned}
\text{TreeResSz}(F \circ OR) &\geq 2^{\text{DelayerScore on } F \circ OR} && \text{[PudImp 2000]} \\
&\geq 2^{DT_1(\text{SearchF})} \\
&\geq 2^{\text{peb}(G)} && (\text{if } F = \text{PebHint}(G)) \\
&\geq 2^{\text{PyramidHeight}} && (\text{for } G{=}\text{Pyr, [Cook 1974]})
\end{aligned}$$

Relating to Resolution

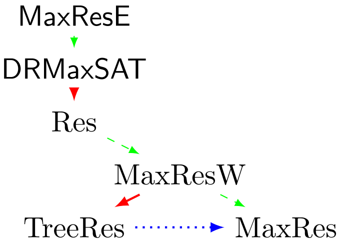


Relating to Resolution



- Does MaxRes, or even MaxResW, simulate Res?
  We wouldn't expect this, but it seemed hard to prove.
- Need a lower bound technique that is specific to MaxResW, not inherited from Res.

Relating to Resolution



- Does MaxRes, or even MaxResW, simulate Res?
  We wouldn't expect this, but it seemed hard to prove.
- Need a lower bound technique that is specific to MaxResW, not inherited from Res.
- Observation: MaxResW refutation $F_0, F_1, \ldots, F_t$ where $\Box \in F_t$.
  Let $G = F_t$ minus one copy of $\Box$.
  - For every assignment α, $\mathrm{viol}_{F_0}(\alpha) = \mathrm{viol}_{F_t}(\alpha) = 1 + \mathrm{viol}_G(\alpha)$.
  - $|G|$ is polynomial in $|F|, t$.

  Showing that every such G is large gives a MaxResW lower bound.

The SubCube Sums Proof System

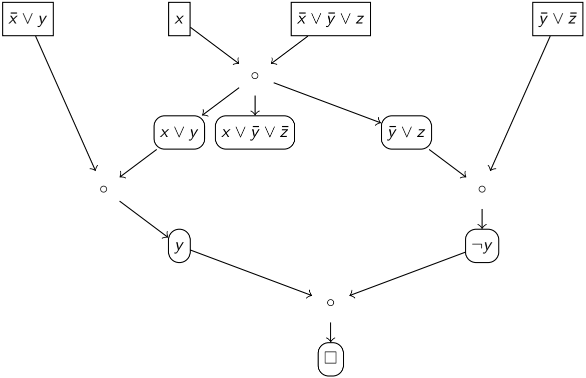
A proof that F is unsatisfiable: a multiset G of clauses such that

$$\forall \alpha : \mathrm{viol}_F(\alpha) = 1 + \mathrm{viol}_G(\alpha).$$

The SubCube Sums Proof System

A proof that F is unsatisfiable: a multiset G of clauses such that

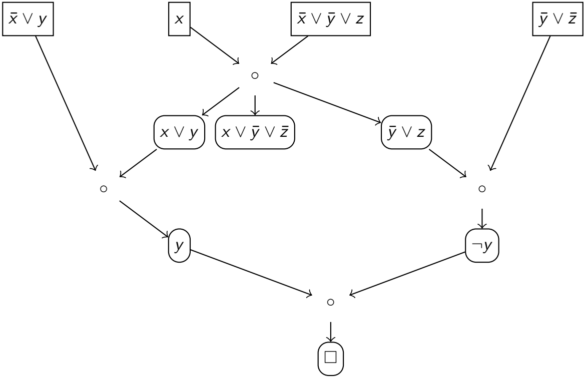
$$\forall \alpha : \mathrm{viol}_F(\alpha) = 1 + \mathrm{viol}_G(\alpha).$$



The SubCube Sums Proof System

A proof that F is unsatisfiable: a multiset G of clauses such that

$$\forall \alpha : \text{viol}_F(\alpha) = 1 + \text{viol}_G(\alpha).$$



Final multiset:
$\{x \vee \bar{y} \vee \bar{z}, \quad \square\}$

The SubCube Sums Proof System (cont'd)

- A proof that F is unsatisfiable: a multiset G of clauses such that

$$\forall\alpha : \mathrm{viol}_{F_0}(\alpha) = \mathrm{viol}_{F_t}(\alpha) = 1 + \mathrm{viol}_G(\alpha).$$

The SubCube Sums Proof System (cont'd)

- A proof that F is unsatisfiable: a multiset G of clauses such that

$$\forall \alpha : \mathrm{viol}_{F_0}(\alpha) = \mathrm{viol}_{F_t}(\alpha) = 1 + \mathrm{viol}_G(\alpha).$$

- Not easy to verify the proof.
  (Possible in randomized polynomial time.)
- Not important if we only use such proofs for MaxResW lower bounds.

The SubCube Sums Proof System (cont'd)

- A proof that F is unsatisfiable: a multiset G of clauses such that

$$\forall \alpha : \mathrm{viol}_{F_0}(\alpha) = \mathrm{viol}_{F_t}(\alpha) = 1 + \mathrm{viol}_G(\alpha).$$

- Not easy to verify the proof.
  (Possible in randomized polynomial time.)
- Not important if we only use such proofs for MaxResW lower bounds.
- Sound: almost by definition
- Complete: every MaxRes refutation gives a SubCubeSums proof.
- SubCubeSums lower bound $\implies$ MaxResW lower bound,
  not Res lower bound.

What's easy for SubCubeSums

- Everything easy for MaxResW, in particular for TreeRes.

What's easy for SubCubeSums

- Everything easy for MaxResW, in particular for TreeRes.
- The SubsetCardinality Formulas.
  A bipartite graph has a subgraph where the left-degrees are bounded above, the right-degrees are bounded below, and the sums are not the same – obviously Unsat.
  On expander graphs, hard for Res and MaxResW [MiksaNordstrom 2014].
  Easy for SubCubeSums.

What's easy for SubCubeSums

- Everything easy for MaxResW, in particular for TreeRes.
- The SubsetCardinality Formulas.
  A bipartite graph has a subgraph where the left-degrees are bounded above, the right-degrees are bounded below, and the sums are not the same – obviously Unsat.
  On expander graphs, hard for Res and MaxResW [MiksaNordstrom 2014].
  Easy for SubCubeSums.
- The PigeonHole Principle Formulas PHP_n^{n+1}.
  Upper bound implicit in [LarrosaRollon 2020], but proved via negative weighted MaxSAT resolution.
  We give direct combinatorial proof.

What's hard for SubCubeSums

The Tseitin Contradictions on expander graphs.
SubCubeSums proofs must have exponential size.

These formulas are also hard for Res.
So not useful for separating Res from MaxResW.

A Lower Bound Technique for SubCubeSums

Theorem (Lifting)

If d is the minimum width of any SubCubeSums refutation of F, then any SubCubeSums refutation of $F \circ XOR$ has size $\exp(\Omega(d))$.

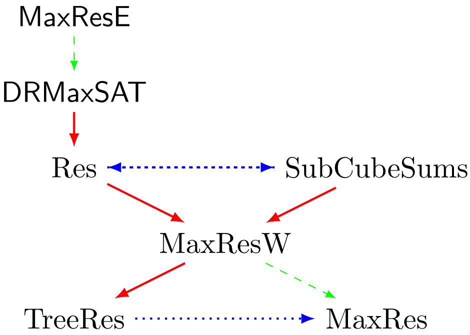
Outline:

- $\mathrm{viol}_F(\alpha_1 \oplus \alpha_2) = \mathrm{viol}_{F \circ \oplus}(\alpha_1, \alpha_2)$. Hence
- $\mathrm{viol}_{F \circ \oplus} - 1 = ((\mathrm{viol}_F) \circ \oplus) - 1 = (\mathrm{viol}_F - 1) \circ \oplus$.
- A "size-width" relation holds for SubCubeSums:
  [SubCubeSums width for F] is
  $O(\log[\text{SubCubeSums size for } F \circ XOR])$.

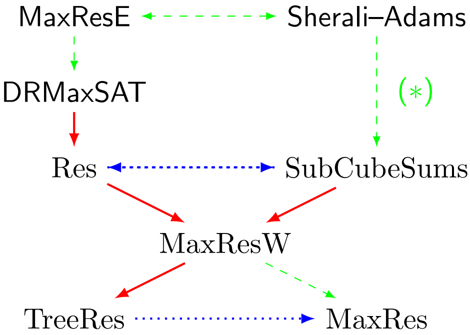
Recent Developments

- A width lower bound for SubCubeSums; [FlemingGöösGrosserRobere 2022].
- The hard formula F is a specific kind of pebbling contradiction. $F \circ XOR$ is easy to refute in Resolution.
- Thus, Resolution is strictly stronger than MaxResW, and incomparable with SubCubeSums.
- Another close variant of MaxRes defined and separated from Resolution; [GöösHollenderJainMaystrePiresRobereTao 2022].

Relations between proof systems



Relations between proof systems



SubCubeSums: An Algebraic View

- $\text{viol}_F(\alpha) = \text{viol}_G(\alpha) + 1.$

SubCubeSums: An Algebraic View

- $\mathrm{viol}_F(\alpha) = \mathrm{viol}_G(\alpha) + 1$.
- Encode clauses as polynomials. $(x \vee \neg y \vee z \to x(1-y)z)$
  $-p_F(\alpha) + p_G(\alpha) + 1 = 0$ for all $\alpha \in \{0,1\}^n$.

SubCubeSums: An Algebraic View

- $\mathrm{viol}_F(\alpha) = \mathrm{viol}_G(\alpha) + 1$.
- Encode clauses as polynomials. $(x \vee \neg y \vee z \rightarrow x(1-y)z)$
  $-p_F(\alpha) + p_G(\alpha) + 1 = 0$ for all $\alpha \in \{0,1\}^n$.
- Polynomials multilinear, hence this is an identity: $-p_F + p_G + 1 \equiv 0$.
  A restricted Sherali–Adams proof!
- Many negated literals in a clause $\implies$ too many monomials.
  Standard approach: use twin variables $x, \bar{x}$ and axioms $x + \bar{x} = 1$.
- $-p_F + p_G + 1 + (\text{a polynomial combination of Boolean axioms}) \equiv 0$.

Algebraic Measures for SubCubeSums

$$-P_F + P_G + 1 + \left(\begin{array}{l}\text{polynomial} \\ \text{combination} \\ \text{of Boolean} \\ \text{twin axioms}\end{array}\right) \equiv 0$$

$$-P_F + P_G + 1 + \sum_j q_j(1 - x_j - \bar{x}_j) \equiv 0$$

Degree

$$D = \max\{D_1, D_2, 0, D_3\}$$

$D_3 \le \max\{D_1, D_2\}$

$$= \max\{D_1, D_2\}$$

D_1: width(F); D_2: width(G)

$\to$ degree of Conical Junta

$\to$ size

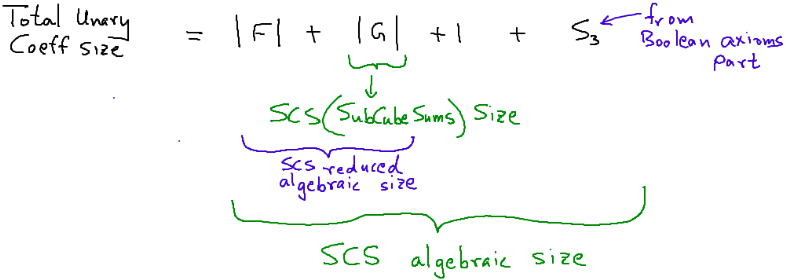
\# Monomials

$$M = M_1 + M_2 + 1 + M_3$$

$$= |F| + |G| + 1 + M_3$$

Total Coeff Size (Unary) $= |F| + |G| + 1 + S_3$

Algebraic Measures for SubCubeSums (cont'd)



- For an SCS proof,
  SCS size $\leq$ SCS reduced algebraic size $\leq$ SCS algebraic size.
- For any unsat formula,
  - Sherali–Adams size $\leq$ unary Sherali–Adams size $\leq$ SCS algebraic size.
  - Sherali-Adams degree $\leq$ SCS degree $= \max\{\mathrm{width}(F), \mathrm{width}(G)\}$.

Take-away

- Using the MaxSAT Resolution and MaxSAT weakening rules to certify unsatisfiability:
  - no worse than TreeRes.
  - on some formulas, exponentially better.
- Key to understanding MaxSAT: rearrangements of Boolean subcubes.
- SubCubeSums proof system:
  - simulates and strictly better than MaxSAT resolution,
  - incomparable with Resolution.
  - can be viewed as a restriction of Sherali–Adams.

Take-away

- Using the MaxSAT Resolution and MaxSAT weakening rules to certify unsatisfiability:
  - no worse than TreeRes.
  - on some formulas, exponentially better.
- Key to understanding MaxSAT: rearrangements of Boolean subcubes.
- SubCubeSums proof system:
  - simulates and strictly better than MaxSAT resolution,
  - incomparable with Resolution.
  - can be viewed as a restriction of Sherali–Adams.

Thank you for listening!