MaxSAT Resolution and Subcube Sums

Yuval Filmus Meena Mahajan *Gaurav Sood* Marc Vinyals

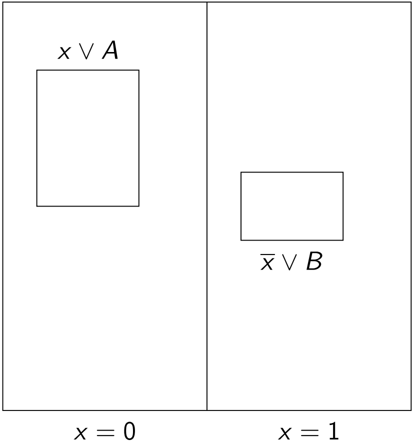
SAT 2020

The MaxSAT problem

- Input: $\mathcal{F}$, a CNF formula (n variables, m clauses).
- Output:

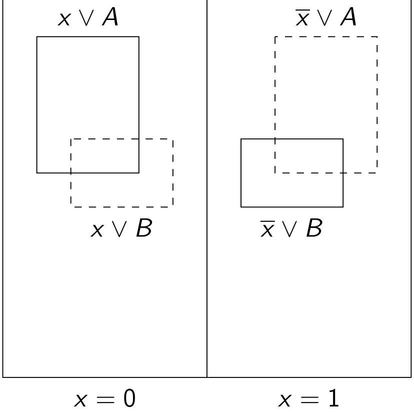
$$\text{MaxSAT}(\mathcal{F}) = \min_{x \in \{0,1\}^n} \{\text{number of clauses falsified by } x\}$$

MaxSAT Resolution (subcube view)



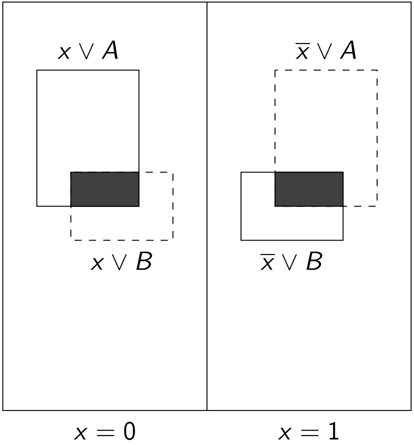
Boolean hypercube $\{0, 1\}^n$

MaxSAT Resolution (subcube view)



Boolean hypercube $\{0, 1\}^n$

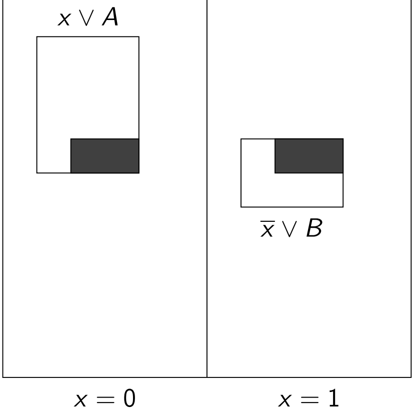
MaxSAT Resolution (subcube view)



Boolean hypercube $\{0, 1\}^n$

- ■ is $A \vee B$

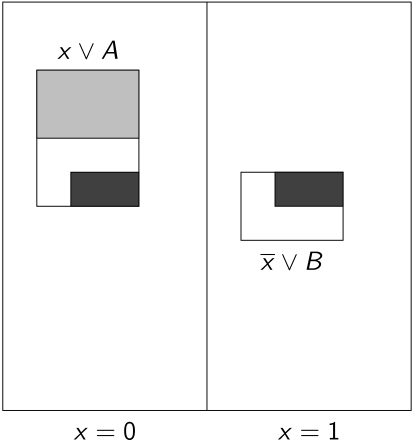
MaxSAT Resolution (subcube view)



Boolean hypercube $\{0, 1\}^n$

- ■ is $A \vee B$

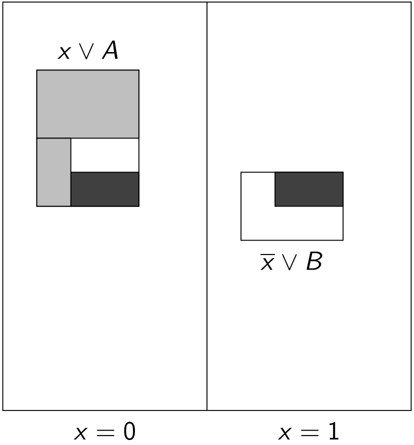
MaxSAT Resolution (subcube view)



Boolean hypercube $\{0, 1\}^n$

- ■ is $A \vee B$
- ■ are fragments of $x \vee A$ and $\overline{x} \vee B$.

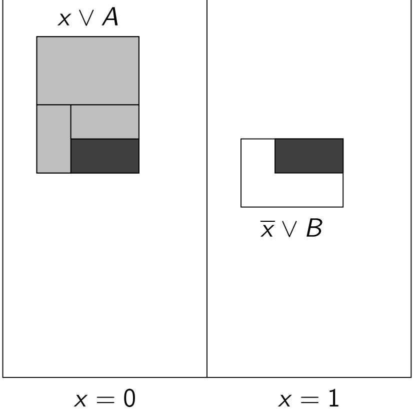
MaxSAT Resolution (subcube view)



Boolean hypercube $\{0, 1\}^n$

- ■ is $A \vee B$
- ■ are fragments of $x \vee A$ and $\overline{x} \vee B$.

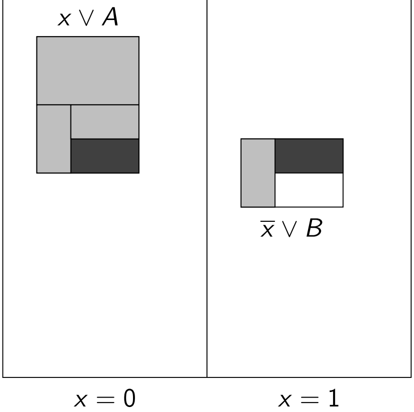
MaxSAT Resolution (subcube view)



Boolean hypercube $\{0,1\}^n$

- ■ is $A \vee B$
- ■ are fragments of $x \vee A$ and $\overline{x} \vee B$.

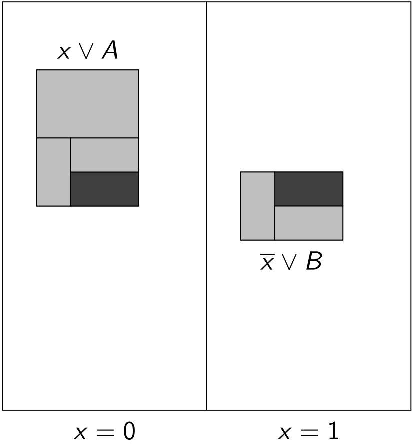
MaxSAT Resolution (subcube view)



Boolean hypercube $\{0, 1\}^n$

- ■ is $A \vee B$
- ■ are fragments of $x \vee A$ and $\overline{x} \vee B$.

MaxSAT Resolution (subcube view)



Boolean hypercube $\{0,1\}^n$

- ■ is $A \vee B$
- ■ are fragments of $x \vee A$ and $\overline{x} \vee B$.

The MaxSAT resolution rule [BonetLevyManya]

$x \vee a_1 \vee \ldots \vee a_s$	$(x \vee A)$
$\overline{x} \vee b_1 \vee \ldots \vee b_t$	$(\overline{x} \vee B)$
$A \vee B$	(the "standard resolvent")
(weakenings of $x \vee A$)	(weakenings of $\overline{x} \vee B$)
$x \vee A \vee \overline{b}_1$	$\overline{x} \vee B \vee \overline{a}_1$
$x \vee A \vee b_1 \vee \overline{b}_2$	$\overline{x} \vee B \vee a_1 \vee \overline{a}_2$
$\vdots$	$\vdots$
$x \vee A \vee b_1 \vee \ldots \vee b_{t-1} \vee \overline{b}_t$	$\overline{x} \vee B \vee a_1 \vee \ldots \vee a_{s-1} \vee \overline{a}_s$

Using the MaxSAT resolution rule for MaxSAT

Goal: Given a CNF formula $\mathcal{F}$, certify that MaxSAT($\mathcal{F}$) $\geqslant k$.
A MaxSAT Resolution proof:

- Maintain a multiset of clauses $\mathcal{C}$.
- Initially, $\mathcal{C} = \mathcal{F}$.
- At each step, pick two clauses $C_1, C_2 \in \mathcal{C}$, apply the MaxSAT resolution rule to them, and *r*eplace them in $\mathcal{C}$ by the consequents.
- If $\mathcal{C}$ contains 'k' copies of the empty clause $\Box$, then MaxSAT($\mathcal{F}$) $\geqslant k$.
- Size of the refutation: number of steps.

Using the MaxSAT resolution rule for MaxSAT (cont'd)

MaxRes is a sound and complete proof system for MaxSAT.

- Soundness:
  If $\mathcal{F} \vdash_{\mathrm{MaxRes}} \underbrace{\Box, \ldots, \Box}_{k \text{ times}}, \mathcal{G}$, then $\mathsf{MaxSAT}(\mathcal{F}) \geqslant k$.
- Completeness:
  If $\mathsf{MaxSAT}(\mathcal{F}) = k$ then $\mathcal{F} \vdash_{\mathrm{MaxRes}} \underbrace{\Box, \ldots, \Box}_{k \text{ times}}, \mathcal{G}$.

Proof system for SAT: MaxRes vs Resolution

- Using MaxRes just to certify unsatisfiability?
  Stop the derivation as soon as a single $\square$ is derived.
- Resolution simulates MaxRes.
  (by definition of MaxSAT resolution)
- In MaxRes, clauses are deleted after use (though some weakenings are added back).
  So makes sense to compare MaxRes with fragments of Resolution which restrict reuse.
- MaxRes simulates read-once resolution.
  Easy to see. But not very interesting.
- Does MaxRes simulate tree-like resolution? We don't know yet.
  Adding a MaxSAT-appropriate weakening rule suffices.

The MaxSAT weakening rule

$$\frac{C}{C \vee x \text{ and } C \vee \overline{x}} \qquad \text{where } x \text{ is a variable not in } C$$

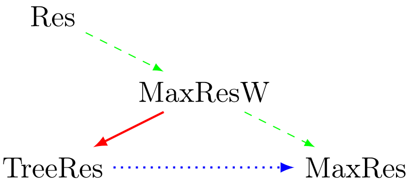
While applying this rule also, we delete the antecedent and add the consequents to the multiset.

Our Results - I

- MaxResW (MaxRes with the weakening rule) simulates TreeRes.
- There is a family of unsatisfiable formulas that is
  - easy in MaxResW,
  - easy even in MaxRes,
  - hard for TreeRes.

  (Pebbling contradictions on Pyramid graphs, composed with OR_2, are hard for TreeRes. [BenSassonWigderson]
  We add some hint clauses to make it easy to refute in MaxRes;
  we show via pebbling that despite hints it remains hard for TreeRes.)

Relations between proof systems



- $A \longrightarrow B$ denotes that A simulates B and B does not simulate A.
- $A \dashrightarrow B$ denotes that A simulates B.
- $A \cdots\!\rightarrow B$ denotes that A does not simulate B.

Lower bounds for MaxRes?

- Resolution simulates MaxResW. So, Res lower bounds translate to MaxResW.
- Question: Is MaxResW as strong as Res?
  Probably not – MaxResW maintains a stronger invariant at each step.
- To establish a separation, need lower bound techniques specific to MaxRes.
- Technique based on the stronger invariant maintained by MaxResW.

MaxResW Invariant

Let $\mathrm{viol}_{\mathcal{F}}(x)$: number of clauses of $\mathcal{F}$ falsified by assignment x.

Invariant maintained:

- MaxResW step $\mathcal{F} \vdash \mathcal{G}$.

$$\mathrm{viol}_{\mathcal{F}}(x) = \mathrm{viol}_{\mathcal{G}}(x) \qquad \text{for all } x \in \{0,1\}^n$$

- MaxResW refutation $\mathcal{F} \vdash_{MaxRes} \Box, \mathcal{G}$.

$$\mathrm{viol}_{\mathcal{G}}(x) = \mathrm{viol}_{\mathcal{F}}(x) - 1 \qquad \text{for all } x \in \{0,1\}^n$$

The SubCube Sums proof system

Abstracting the MaxResW invariant, we define SubCubeSums: a new static proof system for SAT.

- $\mathcal{F}$: an unsatisfiable CNF formula.
- A SubCubeSums refutation: a multiset of clauses $\mathcal{G}$ satisfying

$$\mathrm{viol}_{\mathcal{G}}(x) = \mathrm{viol}_{\mathcal{F}}(x) - 1 \qquad \text{for all } x \in \{0,1\}^n$$

  (This implies $\forall x, \mathrm{viol}_{\mathcal{F}}(x) \geq 1$; hence $\mathcal{F}$ unsat.)
- Size of the refutation: number of clauses in $\mathcal{G}$ (counted with multiplicity).
- Not a proof system in Cook-Reckhow sense; however, verification possible in randomized polynomial time.

Short MaxResW refutation $\Longrightarrow$ small SubCubeSums refutation.

Our Results - II

- A family of formulas requiring large size in SubCubeSums.

  (Tseitin contradictions on expander graphs.
  Lower bound based on how viol behaves, sizes of $\mathrm{viol}^{-1}(i)$.
  Intrinsically different from lower bound for Res.)

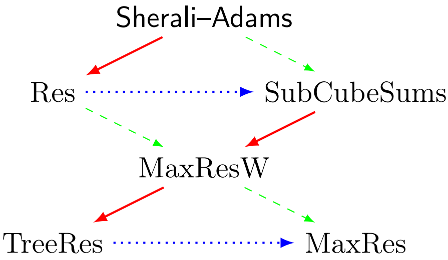
  Does not separate MaxResW from Res.

Our Results - III

- SubCubeSums can be recast as a special case of Sherali-Adams.
- A family of formulas easy in SubCubeSums but hard in Res.
  (Subset Cardinality Formulas; known to be hard for Res.)
  (PigeonHolePrinciple Formulas; known to be hard for Res.
  Easy for SubCubeSums – implicit in [LarrosaRollon-SAT20].
  We give a direct proof.)
- A Lifting Technique:

$$F \text{ requires large width in SubCubeSums}$$
$$\Downarrow$$
$$F \circ \mathrm{XOR} \text{ requires large size in SubCubeSums.}$$

Relations among various proof systems



- $A \longrightarrow B$ denotes that A simulates B and B does not simulate A.
- $A \dashrightarrow B$ denotes that A simulates B.
- $A \cdots\!\!\rightarrow B$ denotes that A does not simulate B.

Wrap-up

Contributions:

- New proof systems for SAT: MaxRes, MaxResW, SubCubeSums.
- Some simulations.
- A new lower bound technique.
- Some non-simulation results.

Some Open Questions:

- Separate MaxResW from Res.
- Understand the role of weakening for MaxSAT.
- Understand SubCubeSums better – somehow connected to integral conical juntas.