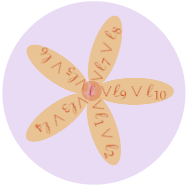
MAJORITY-3SAT (and Related Problems) in Polynomial Time



$\cdots$ $1/4$ $1/2$ 1

Ryan Williams

Shyan Akmal

(appeared in FOCS’21)

CNF-SAT

Given a CNF formula ϕ over variables $x_1, \ldots, x_n$

Question: Is ϕ satisfiable?

Equivalent Question: Is $\Pr_{a \in \{0,1\}^n}[\phi(a) = 1] > 0$?

Is the fraction of satisfying assignments positive?

Complexity: NP-complete, even 3SAT

#CNF-SAT

Given a CNF formula ϕ over variables $x_1, \ldots, x_n$

Question: Compute number of SAT assigns to ϕ

Equivalent Question: Compute $\Pr_{a \in \{0,1\}^n}[\phi(a) = 1]$

Determine the exact fraction of satisfying assignments

#P-complete, even #2SAT

MAJ-SAT

Given a CNF formula ϕ over variables $x_1, \ldots, x_n$

Question: Is $\Pr_{a \in \{0,1\}^n}[\phi(a) = 1] \geq 1/2$?

Equivalently: Compute the most significant bit of the number of SAT assignments

PP-complete

[Simon'75,Gill'77]

$P^{PP} = P^{\#P}$

MAJ-kSAT

Given a k-CNF formula ϕ over variables $x_1, \ldots, x_n$

Question: Compute MAJ-SAT for ϕ

Complexity was still open!

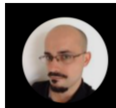
In some papers, the hardness of MAJ-3SAT and extensions was being *assumed* in order to show hardness for other problems…

[BDK'07] Computing if #SAT$(\phi) \geq 2^{n/2}$ is PP-complete for 3-CNF ϕ

It seems most people working in the area believed that MAJ-3SAT was PP-complete, and that we were just lacking a good reduction.

[PLMZ10]

Unfortunately, the resulting decision problem MAJORITY-3SAT is not known to be **PP**-complete. In particular, the standard reduction from SAT to 3SAT [5] is not applicable here, as it requires the addition of "dummy" variables, which increase the number of possible assignments without necessarily increasing the number of satisfying ones: this can decrease the ratio of satisfying assignments over total assignments from above 1/2 to a value less than or equal to this threshold.



Antonio E. Porreca
@aeporreca

Sigh, Majority-3SAT is not known to be PP-complete!

[CM18]

restriction on k, we just write CNF). MAJSAT is PP-complete with respect to many-one reductions even if the input is restricted to be in CNF; however, it is not known whether MAJSAT is still PP-complete with respect to many-one reductions if the sentence ϕ is in 3CNF. Hence we will resort in proofs to a slightly different decision problem, following results by Bailey et al. [7]. The problem

Status of PP-completeness of MAJ3SAT

Asked 5 years, 8 months ago   Modified 11 months ago   Viewed 1k times

SHORT QUESTION: Is MAJ-3CNF a PP-complete problem under many-one reductions?

It turns out MAJ-kSAT is actually easy…

Theorem 1 (MAJ-3SAT is easy)

There is an algorithm which given a 3-CNF φ decides if $\Pr[\varphi] \geq 1/2$ in linear time.

Theorem 2 (THR$_\rho$-kSAT is easy)

Fix any positive integer k and rational $\rho \in (0, 1)$ with constant denominator. There is an algorithm which given a k-CNF φ decides if $\Pr[\varphi] \geq \rho$ in linear time.

A Variant: Greater-Than-MAJ-SAT

MAJ-SAT

Given a CNF formula ϕ, is $\Pr[\phi] \geq 1/2$?

PP-complete

GtMAJ-SAT

Given a CNF formula ϕ, is $\Pr[\phi] > 1/2$?

PP-complete

GtMAJ-3SAT

Given a 3-CNF ϕ, is $\Pr[\phi] > 1/2$?

We prove:

P

GtMAJ-4SAT

Given a 4-CNF ϕ, is $\Pr[\phi] > 1/2$?

NP-complete!

Greater-Than-MAJ-4SAT is NP-hard

Given a 3CNF ϕ on variables $x_{1,} \dots, x_n$:
introduce a new variable y, and add y to every clause of ϕ

The new formula has strictly more than ½ satisfying assignments if and only if ϕ is satisfiable!

So **GtMAJ-4SAT** is NP-hard

It turns out there is also an NP verifier for this problem!

There is a huge difference between MAJ-4SAT and GtMAJ-4SAT
(assuming P $\neq$ NP)

Exists-MAJ-SAT

EMAJ-SAT

Given a CNF formula $\phi(\vec{x}, \vec{y})$ on vars $\vec{x}$ and $\vec{y}$,
is $\exists a \, [\Pr_b [\phi(a,b)] \geq 1/2]$ true?

NP^{PP}-complete

EMAJ-kSAT

Given a k-CNF formula $\phi(\vec{x}, \vec{y})$,
is $\exists a \, [\Pr_b [\phi(a,b)] \geq 1/2]$ true?

We prove:

P for k = 2
NP-complete for $k \geq 3$

Many other results!

Outline for the Rest

- **Some Intuition**
- **MAJ-2SAT is Easy**
- **MAJ-3SAT is Easy**
- **Conclusion**

Some Intuition…

General CNFs

A single clause may have a high fraction of SAT assignments

$$\phi = (x_1 \vee \cdots \vee x_n) \qquad \Pr[\phi] = 1 - \frac{1}{2^n} \approx 1$$

2-CNFs

A single clause already restricts the fraction considerably

$$\phi = (x_a \vee x_b) \wedge \cdots \qquad \Pr[\phi] \leq \frac{3}{4}$$

Two “disjoint” clauses restrict the fraction further…

$$\phi = (x_a \vee x_b) \wedge (x_c \vee x_d) \wedge \cdots \qquad \Pr[\phi] \leq \left(\frac{3}{4}\right)^2 < 0.57$$

a, b, c, d are distinct indices

Some Intuition…

2-CNFs

Three "disjoint" clauses already restrict the fraction below 1/2

$$\phi = (x_a \vee x_b) \wedge (x_c \vee x_d) \wedge (x_e \vee x_f) \wedge \cdots \qquad \Pr[\phi] \le \left(\frac{3}{4}\right)^3 < 0.43$$

a, b, c, d, e, f are distinct

Completely analogous reasoning holds for k-CNFs!

If ϕ contains a variable-disjoint set of t clauses of width k,

$$\Pr[\phi] \le \left(1 - \frac{1}{2^k}\right)^t \le e^{-\frac{t}{2^k}}$$

So let's look for large sets of disjoint clauses! **But if we can't find them, we need to do something else…**

MAJ-2SAT Algorithm

Idea: Search For Variable-Disjoint Clause Sets

Given a 2-CNF ϕ, is $\Pr_{a \in \{0,1\}^n}[\phi(a) = 1] \geq 1/2$?

$$\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee x_5) \wedge (x_1 \vee \neg x_6) \wedge \cdots$$

} Satisfied at most $3/4$ of the time

Implies that $\Pr[\varphi] \leq 3/4$

} An independent constraint

Implies that $\Pr[\varphi] \leq (3/4)^2$

MAJ-2SAT Algorithm

Greedy Algorithm for Disjoint Sets:

Initialize $S := \emptyset$

Pass through the clauses one at a time

If clause C is variable-disjoint from all of S, add C to S

Produces maximal disjoint set S:
for all clauses C' not in S, there is a clause C in S such that C and C' share at least one variable

$$
\begin{aligned}
\varphi = &(x_1 \vee \neg x_2) \wedge \\
&(x_2 \vee \neg x_3) \wedge \\
&(x_4 \vee x_5) \wedge \\
&(x_1 \vee \neg x_6) \wedge \\
&(x_4 \vee x_6) \wedge \\
&(x_7 \vee \neg x_2) \wedge \\
&(x_3 \vee x_7) \wedge \\
&(x_3 \vee x_4) \wedge \\
&\vdots
\end{aligned}
$$

MAJ-2SAT Algorithm

Given a 2-CNF φ is $\Pr[\varphi] \geq 1/2$?

1. Run greedy algorithm for disjoint sets, get back a clause set S.
2. Suppose $|S| \geq 3$.

   Implies $\Pr[\varphi] \leq (3/4)^3 < 1/2$

   Return **NO** for MAJ-2SAT

3. Suppose $|S| < 2$… ***what to do, then?***

$$\begin{aligned}\varphi = & (x_1 \vee \neg x_2) \wedge \\ & (x_2 \vee \neg x_3) \wedge \\ & (x_4 \vee x_5) \wedge \\ & (x_1 \vee \neg x_6) \wedge \\ & (x_4 \vee x_6) \wedge \\ & (x_7 \vee \neg x_2) \wedge \\ & (x_3 \vee x_7) \wedge \\ & (x_3 \vee x_4) \wedge \\ & \vdots\end{aligned}$$

MAJ-2SAT Algorithm: case of small $|S|$

Fact: If S is a **maximal disjoint set**, then the union of all variables in all clauses of S forms a ***hitting set for all clauses in ϕ***

Hitting set $H = \{x_1, x_2, x_4, x_5\}$

Consider any assignment $\alpha : H \rightarrow \{0,1\}$

This sets at least one variable in every clause, so the formula simplifies to a **1-CNF**

Example: If $x_1, x_2, x_4 \mapsto 0$ and $x_5 \mapsto 1$ …

$$\varphi = (x_1 \vee \neg x_2) \wedge$$

$$\varphi_\alpha = \neg x_3 \wedge \neg x_6 \wedge x_6 \wedge x_7 \wedge x_3 \wedge \vdots$$

MAJ-2SAT Algorithm: case of small $|S|$

It is easy to solve #SAT on 1-CNF!

If constraints are inconsistent: $\Pr[\varphi_\alpha] = 0$

If constraints are consistent, and v distinct variables appear: $\Pr[\varphi_\alpha] = 1/2^v$

$$\Pr[\varphi] = \sum_{\alpha:H\to\{0,1\}} \Pr\left[\varphi|_\alpha\right]$$

<u>Idea:</u> Enumerate all assignments to H that satisfy the clauses they appear in, solve #1SAT on each subformula obtained.
We'll compute #SAT exactly in this case!

$$\varphi_\alpha = \neg x_3 \wedge \neg x_6 \wedge x_8 \wedge x_7 \wedge x_9 \wedge \vdots$$

MAJ-2SAT Algorithm

Given a 2-CNF φ is $\Pr[\varphi] \geq 1/2$?

1. Run greedy algorithm for disjoint sets, get back a clause set S.
2. Suppose $|S| \geq 3$.

   Implies $\Pr[\varphi] \leq (3/4)^3 < 1/2$

   Return **NO** for MAJ-2SAT
3. Suppose $|S| \leq 2$.
   Try all SAT assignments to S, obtaining 1-CNFs. Solve #SAT on each of them to **determine #SAT** for the entire formula.

$$\begin{aligned}\varphi = &(x_1 \vee \neg x_2) \wedge \\ &(x_2 \vee \neg x_3) \wedge \\ &(x_4 \vee x_5) \wedge \\ &(x_1 \vee \neg x_6) \wedge \\ &(x_4 \vee x_6) \wedge \\ &(x_7 \vee \neg x_2) \wedge \\ &(x_3 \vee x_7) \wedge\end{aligned}$$

The same strategy works for *all* thresholds, not just ½

Alternative Perspective: MAJ-2SAT

Every 2-CNF has one of two possible forms:

Random-Like

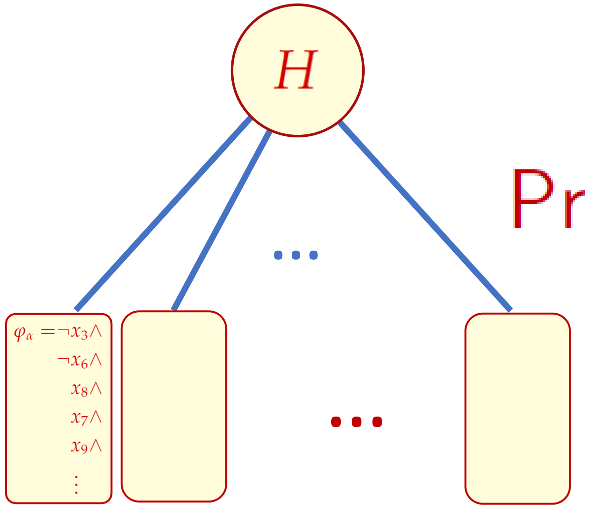
Has "Bad" Subformula

$$\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee x_5) \wedge (x_1 \vee \neg x_6) \wedge (x_4 \vee x_6) \wedge (x_7 \vee \neg x_2) \wedge (x_3 \vee x_7) \wedge (x_3 \vee x_4) \wedge \vdots$$

$\Pr[\varphi]$ is small

Structured

Small Sum of 1-CNFs



$\Pr[\varphi]$ is easy to compute

MAJ-3SAT Algorithm

Given a 3-CNF φ is $\Pr[\varphi] \geq 1/2$?

$$\varphi = (x_1 \vee \neg x_2 \vee x_7) \wedge$$
$$(x_2 \vee \neg x_3 \vee x_6) \wedge$$
$$(x_4 \vee x_5 \vee \neg x_8) \wedge$$
$$(x_1 \vee \neg x_6 \vee x_5) \wedge$$
$$\vdots$$

} Satisfied at most $7/8$ of the time

Implies that $\Pr[\varphi] \leq 7/8$

If we find at least d disjoint clauses...

Implies that $\Pr[\varphi] \leq (7/8)^d$

For $d \geq 6$ we have $\Pr[\varphi] \leq (7/8)^6 < 1/2$
and we can report **NO**

What can we do when $d < 6$?

As before, we get a "small" hitting set

Hitting set

$H = \{x_1, x_2, x_3, x_4, x_6, x_7\}$

Any assignment $\alpha : H \to \{0,1\}$ to φ induces a **2-CNF** φ_α

$$\Pr[\varphi] = \sum_{\alpha : H \to \{0,1\}} \Pr[\varphi_\alpha]$$

But now $\Pr[\varphi_\alpha]$ is #P-hard to compute…

$$\begin{aligned}
\varphi = &(x_1 \vee \neg x_2 \vee x_7) \wedge \\
&(x_3 \vee \neg x_4 \vee x_6) \wedge \\
&(x_1 \vee x_5 \vee \neg x_8) \wedge \\
&(x_5 \vee x_6 \vee \neg x_9) \wedge \\
&(x_6 \vee x_7 \vee x_8) \wedge \\
&\vdots
\end{aligned}$$

Search for Disjoint Sets… Again!

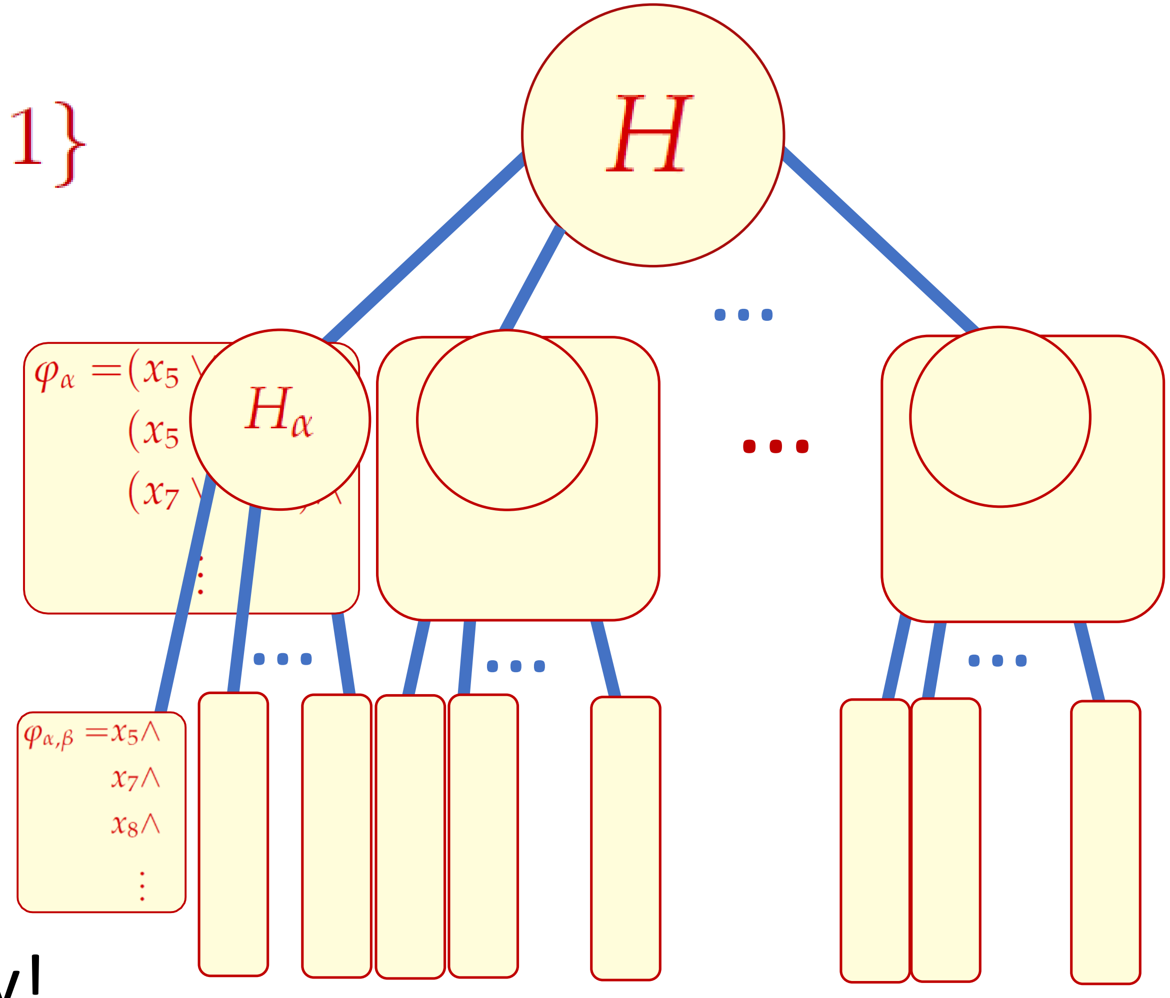
Try all assignments $\alpha : H \rightarrow \{0,1\}$

For each 2-CNF φ_α, look for **maximal disjoint set** in φ_α

Either (1) all these disjoint sets are "small",
or (2) a disjoint set is "large"

If all are small, obtain 1-CNFs

In case (1), compute $\Pr[\varphi]$ exactly!



Picking out a Sunflower

$$H = \{x_1, x_2, x_3, x_4, x_6, x_7\}$$

Suppose φ_α has a disjoint set of size at least d …

$$\begin{aligned}\varphi_\alpha = &(x_5 \vee \neg x_8)\wedge\\ &(x_9 \vee \neg x_{10})\wedge\\ &(x_{11} \vee x_{12})\wedge\\ &(x_{13} \vee \neg x_{14})\wedge\\ &\vdots\end{aligned}$$

$$\begin{aligned}\varphi = &(\ell_1 \vee x_5 \vee \neg x_8)\wedge\\ &(\ell_2 \vee x_9 \vee \neg x_{10})\wedge\\ &(\ell_3 \vee x_{11} \vee x_{12})\wedge\\ &(\ell_4 \vee x_{13} \vee \neg x_{14})\wedge\\ &\vdots\end{aligned}$$

Picking out a Sunflower

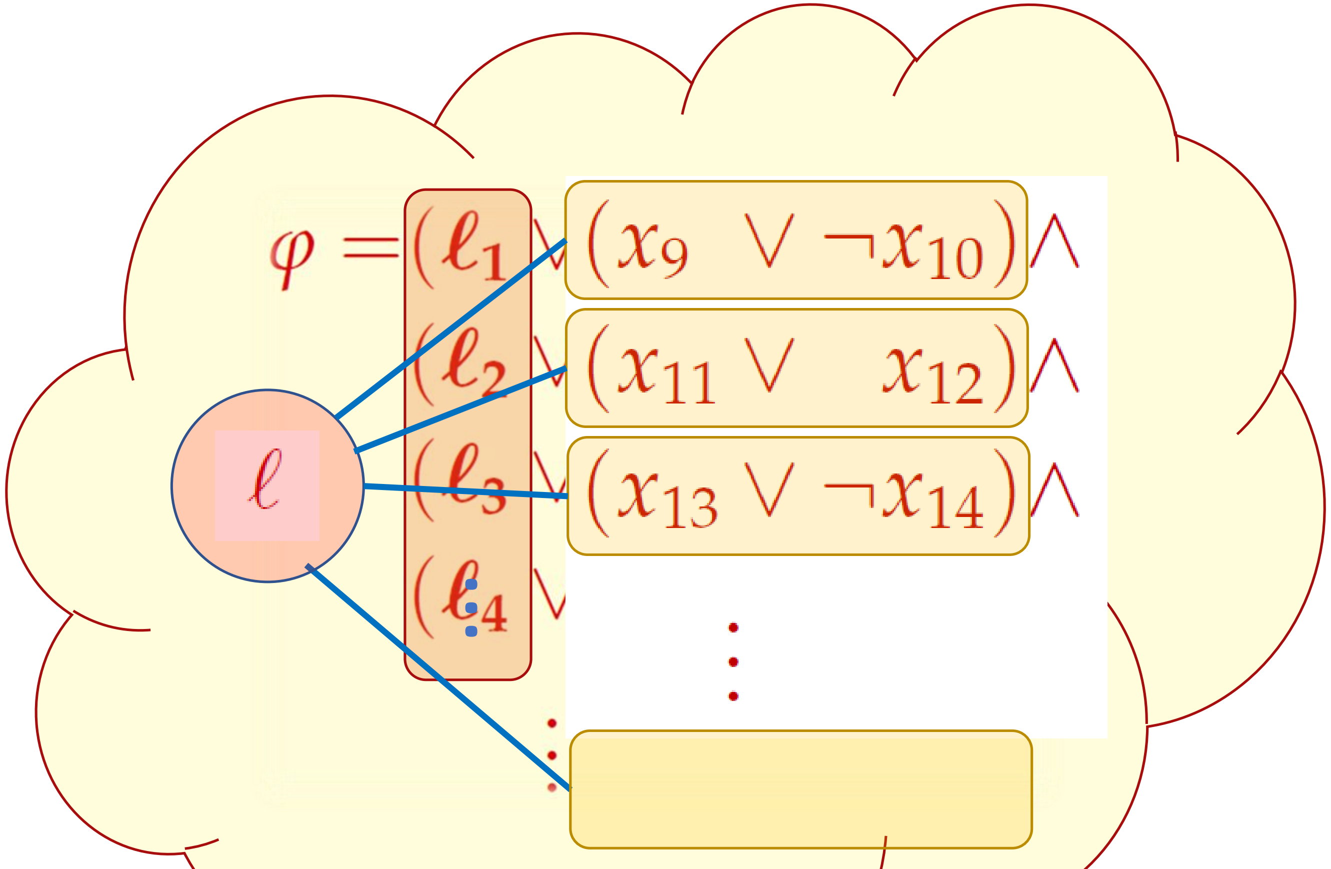
$H = \{x_1, x_2, x_3, x_4, x_6, x_7\}$

Suppose φ_α has a disjoint set of size at least d …

Some literal ℓ from H must appear many times…

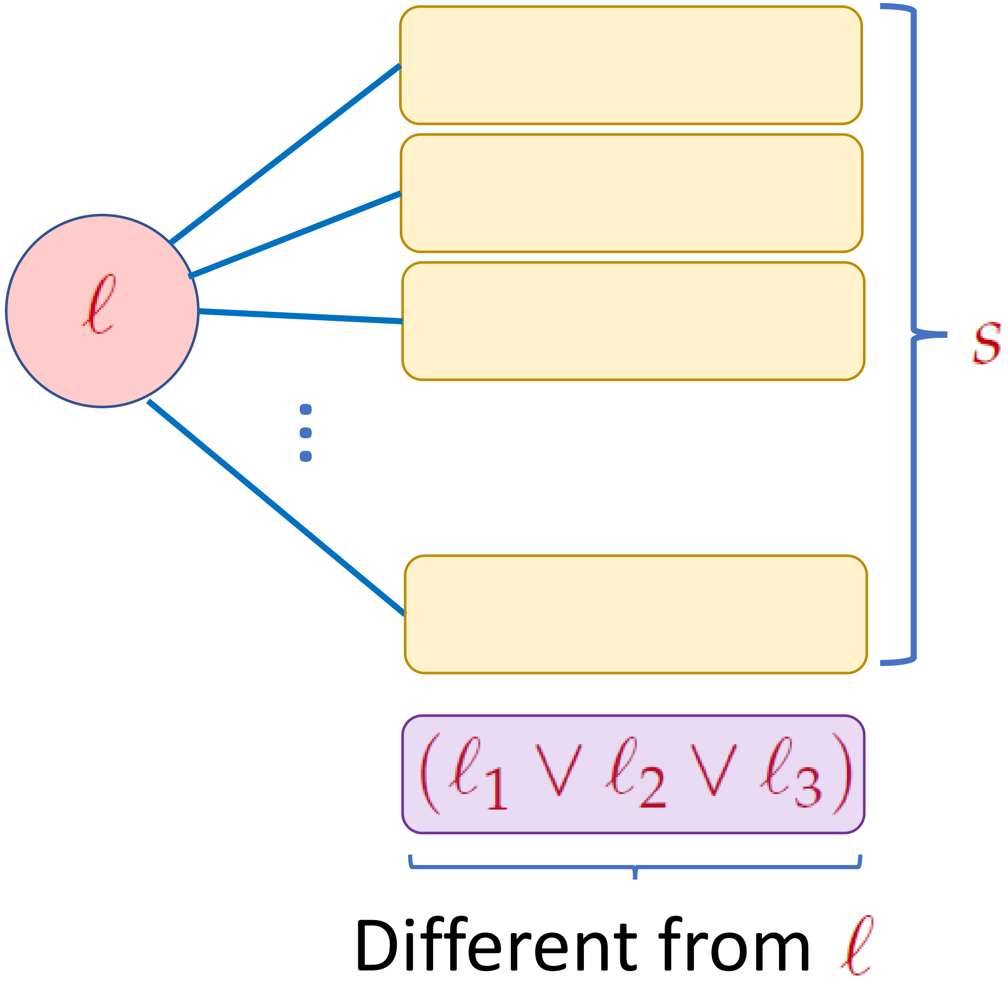
$\geq d/(2|H|)$

…by the pigeonhole principle



We obtain a sunflower with core $\{\ell\}$

How to use the sunflower



If ℓ appears in ***every*** clause,
then $\Pr[\varphi] \geq 1/2$

Otherwise, for $s \geq 8$,

$$\Pr[\varphi] \leq \underbrace{\frac{1}{2} \cdot \left(\frac{7}{8}\right)}_{\ell = 1} + \underbrace{\frac{1}{2} \cdot \left(\frac{3}{4}\right)}_{\ell = 0}^{s} < \frac{1}{2}$$

MAJ-3SAT resolved in either case!

MAJ-3SAT Algorithm

1. Find a maximal disjoint set of clauses in φ
2. If disjoint set has size ≥ 6, return NO
3. Otherwise, find a hitting set H of ≤ 18 variables for φ

(same as MAJ-2SAT)

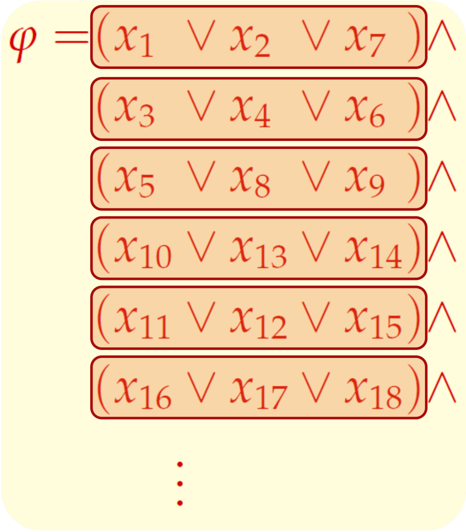
4. Try all SAT assignments to H, obtaining **2-CNFs**
5. For each **2-CNF**, find a maximal disjoint set
6. If all disjoint sets are ≤ 7, obtain **1-CNF**s and **compute** $\Pr[\varphi]$
7. Otherwise some disjoint set is ≥ 8, yielding a "large" sunflower. If the sunflower core hits all clauses return **YES**, otherwise return **NO**

High-Level Intuition for MAJ-3SAT Algorithm

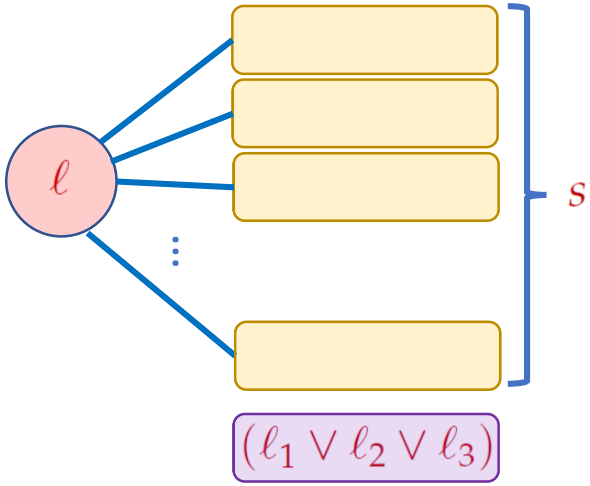
"Bad" Subformula

Structured

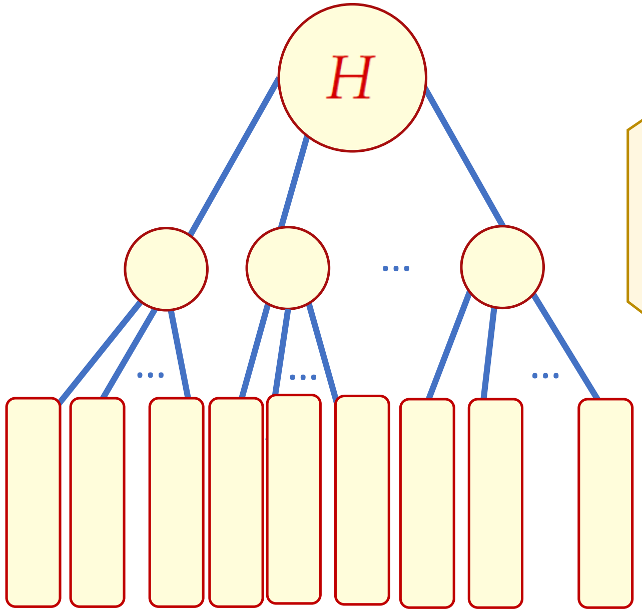
Big Disjoint Set



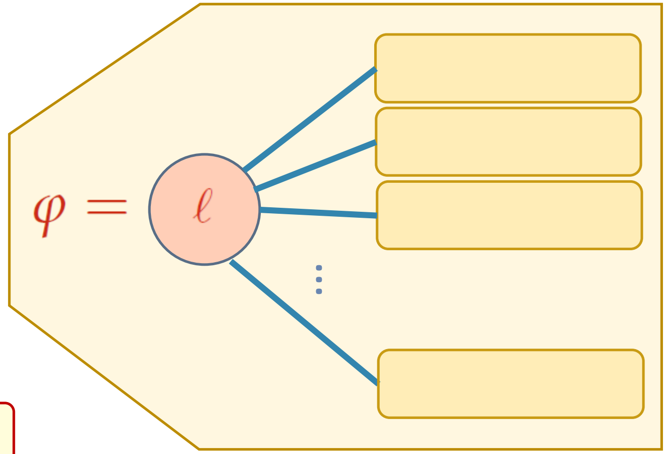
Sunflower + Extra



Sum of 1-CNFs



Just Sunflower



ℓ appears in all clauses

Going Beyond MAJ-3SAT

MAJ-3SAT

3-CNF

$\Pr[\varphi] \geq 1/2$

MAJ-kSAT

k-CNF

$\Pr[\varphi] \geq 1/2$

THR$_\rho$-kSAT

3-CNF

$\Pr[\varphi] \geq \rho$

Extract More Disjoint Sets!

Extract More Sunflowers!

Conclusion

Extract Disjoint Sets

Extract Sunflowers

For 2-CNFs, either $\Pr[\varphi]$ is either "easy" to compute, or *small*

For 3-CNFs, similar, but single literal may hit all clauses

In general: testing k-CNFs at any constant threshold is "easy"

Some Future Directions:

What other problems have easy threshold versions?
Generalization to k-CSPs of domain $d \geq 3$?

Better parameterized algorithms?
(Terrible running time dependence on k)

Thank You!