Where Can We Draw The Line?

On the Hardness of Satisfiability Problems

Introduction

- Objectives:
  - To show variants of SAT and check if they are NP-hard
- Overview:
  - Known results
  - 2SAT
  - Max2SAT

What Do We Know?

- Checking if a propositional calculus formula is satisfiable (SAT) is NP-hard.

Example: propositional calculus formula

$$\neg(x \wedge \neg z \wedge (\neg w \vee x)) \vee (x \wedge \neg y) \rightarrow \neg y$$

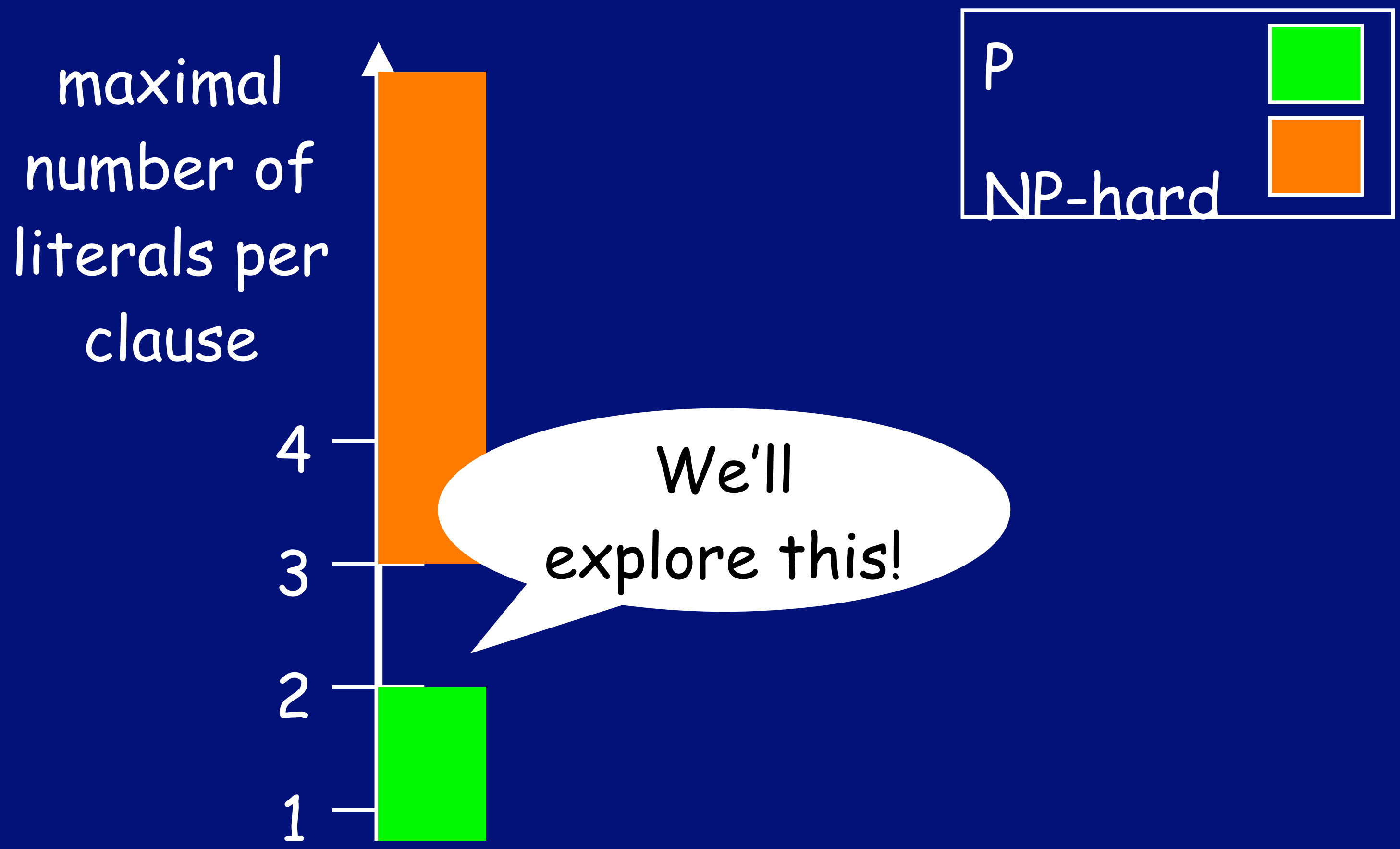
What Do We Know?

- We concentrated on a special case: CNF formulas.

structure of CNF formulas

$(..\vee..\vee.....\vee..)\wedge...\wedge(..\vee..\vee.....\vee..)$

What Do We Know?



2SAT

- Instance: A 2-CNF formula φ
- Problem: To decide if φ is satisfiable

Example: a 2CNF formula

$$(\neg x \vee y) \wedge (\neg y \vee z) \wedge (x \vee \neg z) \wedge (z \vee y)$$

PAP 184-185

2SAT is in P

Theorem: 2SAT is polynomial-time decidable.

Proof: We'll show how to solve this problem efficiently using path searches in graphs…

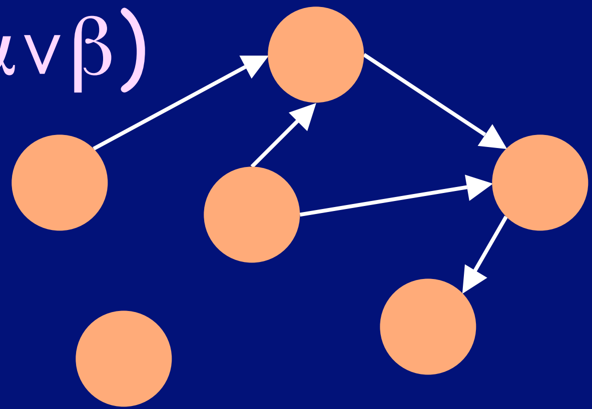
Searching in Graphs

<u>Theorem:</u> Given a graph $G=(V,E)$ and two vertices $s,t\in V$, finding if there is a path from s to t in G is polynomial-time decidable.

<u>Proof:</u> Use some search algorithm (DFS/BFS). ■

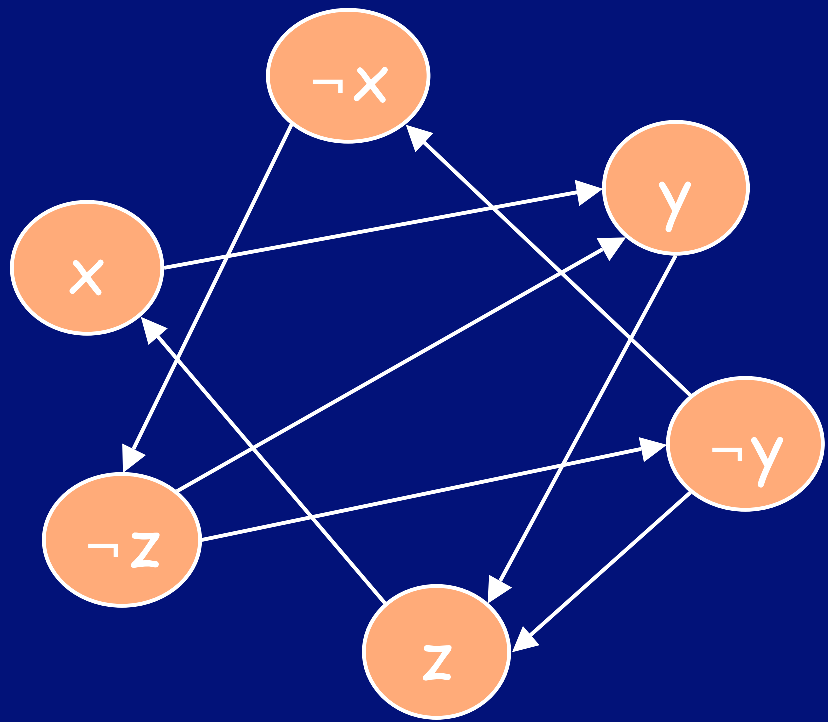
Graph Construction

- Vertex for each variable and a negation of a variable
- Edge (α,β) iff there exists a clause equivalent to $(\neg\alpha \vee \beta)$



Graph Construction: Example

$(\neg x \vee y) \wedge (\neg y \vee z) \wedge (x \vee \neg z) \wedge (z \vee y)$



Observation

Claim: If the graph contains a path from α to β, it also contains a path from $\neg\beta$ to $\neg\alpha$.

Proof: If there's an edge (α,β), then there's also an edge $(\neg\beta,\neg\alpha)$.

Correctness

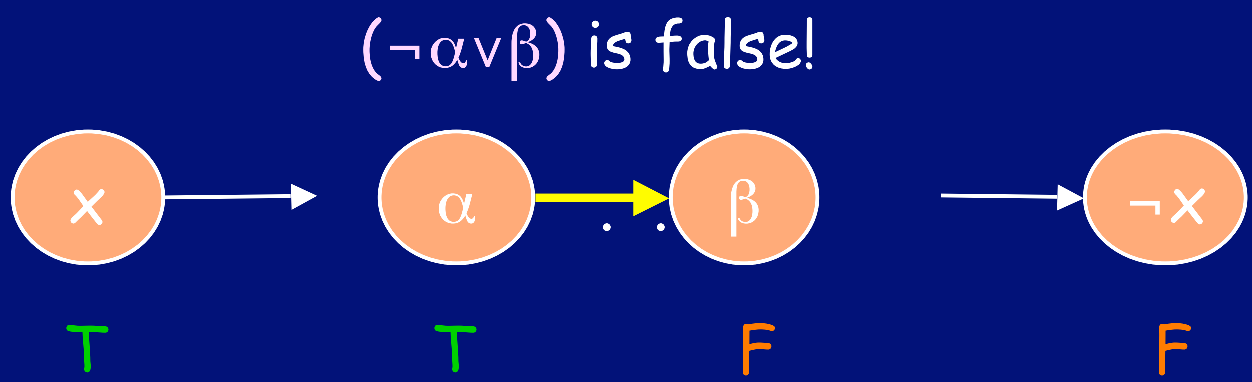
Claim:

a 2-CNF formula φ is unsatisfiable iff there exists a variable x, such that:

1. there is a path from x to $\neg x$ in the graph
2. there is a path from $\neg x$ to x in the graph

Correctness (1)

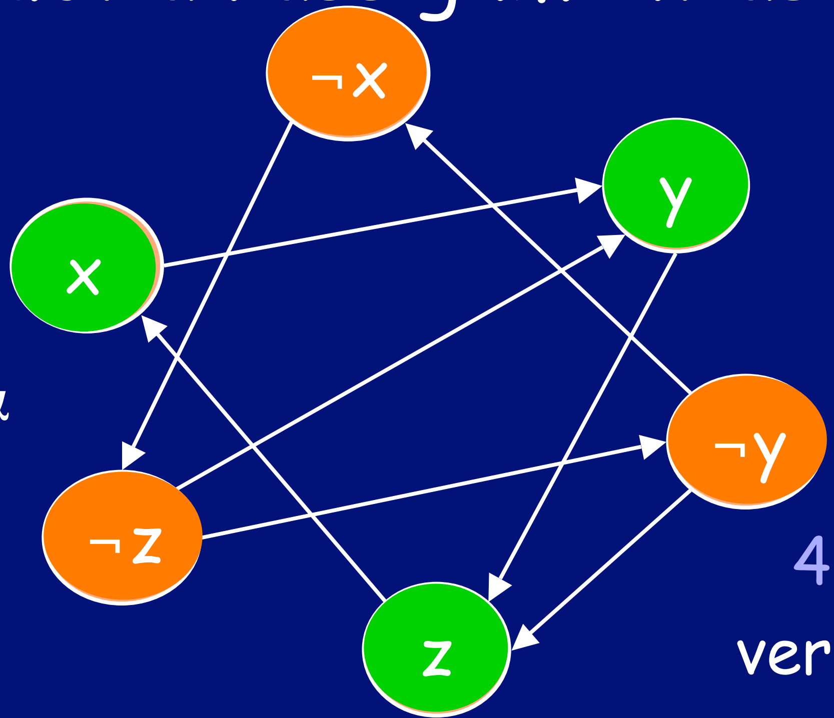
- Suppose there are paths $x..\neg x$ and $\neg x..x$ for some variable x, but there's also a satisfying assignment ρ.
- If $\rho(x)=T$ (similarly for $\rho(x)=F$):



Correctness (2)

- Suppose there are no such paths.
- Construct an assignment as follows:

1. pick an unassigned literal α, with no path from α to $\neg\alpha$, and assign it T



2. assign T to all reachable vertices
3. assign F to their negations
4. Repeat until all vertices are assigned

Correctness (2)

Claim: The algorithm is well defined.

Proof: If there were a path from x to both y and ¬y,

then there would have been a path from x to ¬y and from ¬y to ¬x.

Correctness

A formula is unsatisfiable iff there are no paths of the form x..¬x and ¬x..x. ■

2SAT is in P

We get the following efficient algorithm for 2SAT:

- For each variable x find if there is a path from x to ¬x and vice-versa.
- Reject if any of these tests succeeded.
- Accept otherwise

⇒ 2SAT∈P. ■

Max2SAT

- Instance: A 2-CNF formula φ and a goal K.
- Problem: To decide if there is an assignment satisfying at least K of φ's clauses.

Example: a 2CNF formula

$(\neg x \vee y) \wedge$
$(\neg y \vee z) \wedge$
$(x \vee \neg z) \wedge$
$(z \vee y)$

PAP 186-187

Max2SAT is in NPC

<u>Theorem:</u> Max2SAT is NP-Complete.

<u>Proof:</u> Max2SAT is clearly in NP.

We'll show $3SAT \leq_p Max2SAT$.

$(..\vee..\vee..)\wedge...\wedge(..\vee..\vee..)$ $\leq_p$ $(..\vee..)\wedge...\wedge(..\vee..)$

K

Gadgets

Proof: By checking.

Claim: Let

$$\psi(x,y,z,w) = (x)\wedge(y)\wedge(z)\wedge(w)\wedge$$
$$(\neg x\vee\neg y)\wedge(\neg y\vee\neg z)\wedge(\neg z\vee\neg x)\wedge$$
$$(x\vee\neg w)\wedge(y\vee\neg w)\wedge(z\vee\neg w).$$

- Every satisfying assignment for $(x\vee y\vee z)$ can be extended into an assignment that satisfies exactly 7 of the clauses.
- Other assignments can satisfy at most 6 of the clauses.

The Construction

- For each $1 \leq i \leq m$, replace the i-th clause of the 3-CNF formula $(\alpha \vee \beta \vee \gamma)$ with a corresponding $\psi(\alpha,\beta,\gamma,w_i)$ to get a 2-CNF formula.
- Fix $K=7m$.

Make sure this construction is poly-time

Correctness

- Every satisfying assignment for the 3-CNF formula can be extended into an assignment that satisfies 7m clauses.
- If 7m clauses of the 2-CNF formula are satisfied, each ψ has 7 satisfied clauses, so the original formula is satisfied.

Corollary

$\Rightarrow$ $3SAT \leq_p Max2SAT$ and $Max2SAT \in NP$

$\Rightarrow$ Max2SAT is NP-Complete. ■

Summary

- We've seen that checking if a given CNF formula is satisfiable is:
  - Polynomial-time decidable, if every clause contains up to 2 literals.
  - NP-hard, if each clause may contain more than 2 literals.
- We've also seen Max2SAT is NP-hard.

Conclusions

- A special case of a NP-hard problem may be polynomial time decidable.
- The optimization version of a polynomial-time decidable problem may be NP-hard.