CS 314 - NP-Hardness

Note Title 3/22/2010

Announcements

- Midterms graded: Max: 59/60
  Mins in 20's
  If near 30 (50% range) or have D/F so far this semester, check in!

  (Remember, withdrawal requires petition to dean, so need to handle it today/tomorrow.)

- No HW due this week - next one will come out Wed. or Thurs.

Ch 8: NP-Hardness (also lecture notes)

So far, we've been looking at (mostly) polynomial time algorithms.

Why?

Historical Note: In 60's, CS folks decided that an algorithm running in time $O(n^c)$ was a minimal requirement.

c is constant

But what about hard problems??

Still stuck!

Many useful problems can't be proven "hard".

Usually, by "hard" we'd like to show there is no $O(n^c)$ algorithm possible for any constant c.

So what have we been doing for the last 50 years?!

P, NP, & co-NP

Consider decision problems - output is a single boolean (yes or no).

Define:
- P : the set of problems that can be solved in polynomial time

nondeterministic polynomial

- NP: the set of decision problems where if the answer is Yes, there is a proof of this that can be checked in polynomial time
- co-NP: If answer is No, that can be checked in polynomial time.

Examples:

- Is this list sorted? in P

- In a graph G, is there an independent set of size $\geq k$?
  don't know if in P or co-NP
  in NP: if I give you k vertices, you can check if they form an ind. set

- Is the number x a prime number?
  – in P — about 5 years old — in co-NP

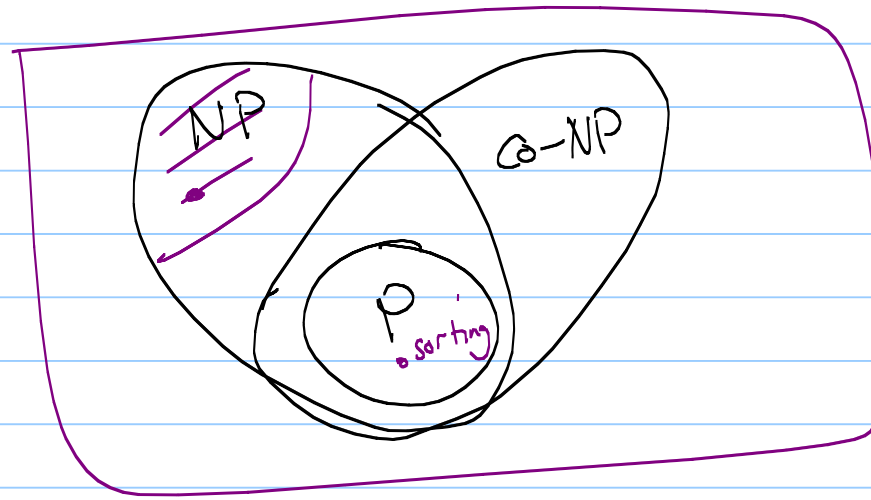
- Is the number x a composite #?
  – in P

Input: log n bits

Largest open question in CS

Is $P = NP$?

Know: $P \subseteq NP$
$P \subseteq co\text{-}NP$



(Don't even know if $NP = co\text{-}NP$.)

NP-Hard

Dfn: A problem Π is NP-Hard

$\Longleftrightarrow$ (if & only if)

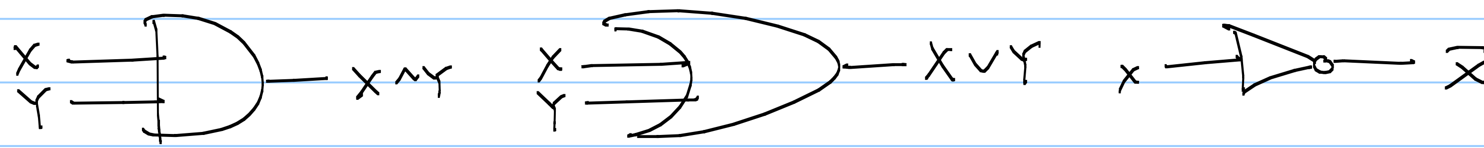
if Π can be solved in polynomial time, then P = NP.

Dfn: A problem is NP-Complete if it is both NP-Hard and in NP.

These are the "hardest" problems in NP.

Circuit-SAT

Input: boolean circuit, with T/F inputs

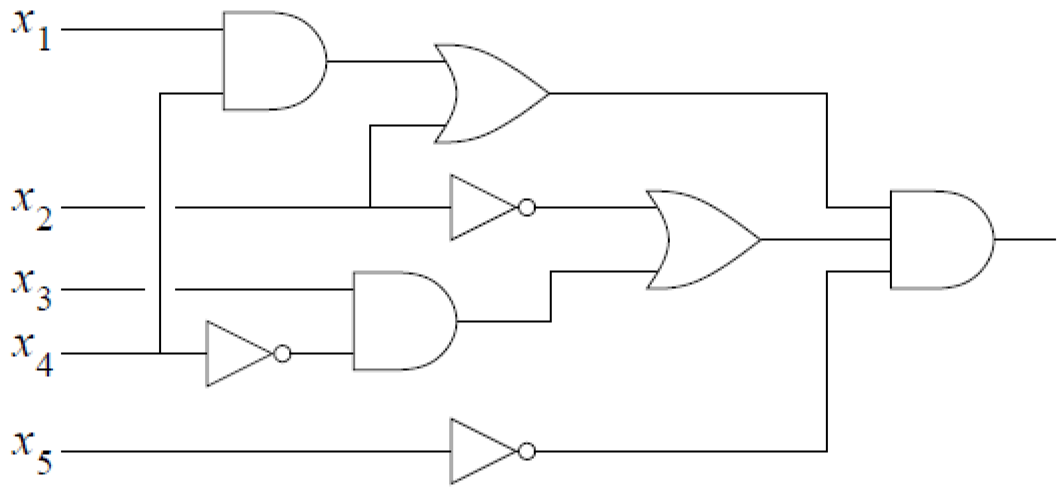


AND gate OR gate NOT gate

Output: T or F

Example:

x_1
$\vdots$
x_n



A boolean circuit. inputs enter from the left, and the output leaves to the right.

8 gates
5 inputs

2^n possible inputs
m gates

Question: Given a circuit, is there a set of inputs which makes the output be True?

Where does this fit?

in NP - check output of circuit in $O(m)$ time.

Cook-Levin Theorem:

Circuit-satisfiability is NP-Complete.