polynomial time : $O(n^c)$
for some constant c

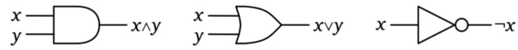
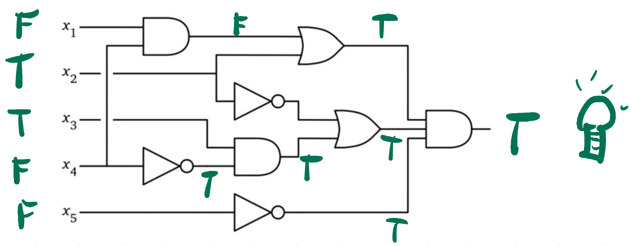


Figure 15.1. An AND gate, an OR gate, and a NOT gate.



Given a Boolean circuit with n input gates $x_1, \ldots, x_n$

Connected as a DAG with AND, OR, + NOT gates.

Circuit has 1 output wire.

Can we set inputs to True/False so output is True?

Circuit satisfiability

Circuit SAT.

Algorithm: For each setting of input, compute output

$O(2^n n)$ time

Nobody knows how to do better!

But we can check a single input suggestion in $O(n)$ time.

Decision Problems:

Output is True or False.
(Yes or No)

(main)
Three <u>classes</u> of decision problems.

P: Can solve in polynomial time.

Ex: Decision version of min spanning tree (given G & a number k, does MST of G cost at most k?)

NP: Decision problem where, if answer is True, there exists a proof you can verify or dismiss in polynomial time.

Ex: CircuitSAT

co-NP: If answer is False, there is a proof you can check in poly time.

Ex: Prime: Given a n-bit integer w, is w prime?

NP : Non-deterministic polynomial (time)

(not the same as quantum)

Facts: $P \subseteq NP$ (use empty proof. "verify" by solving from scratch)

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

Big Question: $P \overset{?}{\neq} NP$

Most think $P \neq NP$.

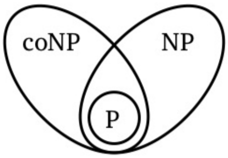
Clay Mathematics Institute

7 Millennium Prize Problems

\$1,000,000 to prove or disprove $P = NP$.

Another problem: $NP \stackrel{?}{=} \text{co-}NP$

The World?



Probem B (decision or not) is <u>NP-hard</u> if for every problem $A \in NP$, we can reduce A <u>to</u> B in polynomial (IAD) time.

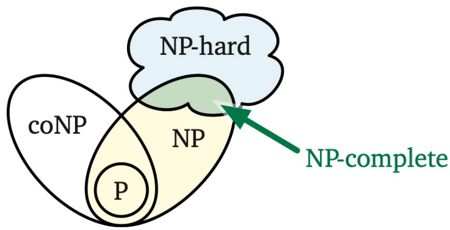
=> A poly time alg for B would imply a poly time alg for all $A \in NP$.

=> a poly time alg for B implies $P = NP$.

~> B probably has no poly time alg.

If $B \in NP$ & B is NP-hard, we say B is (is in) NP-complete.

The World?



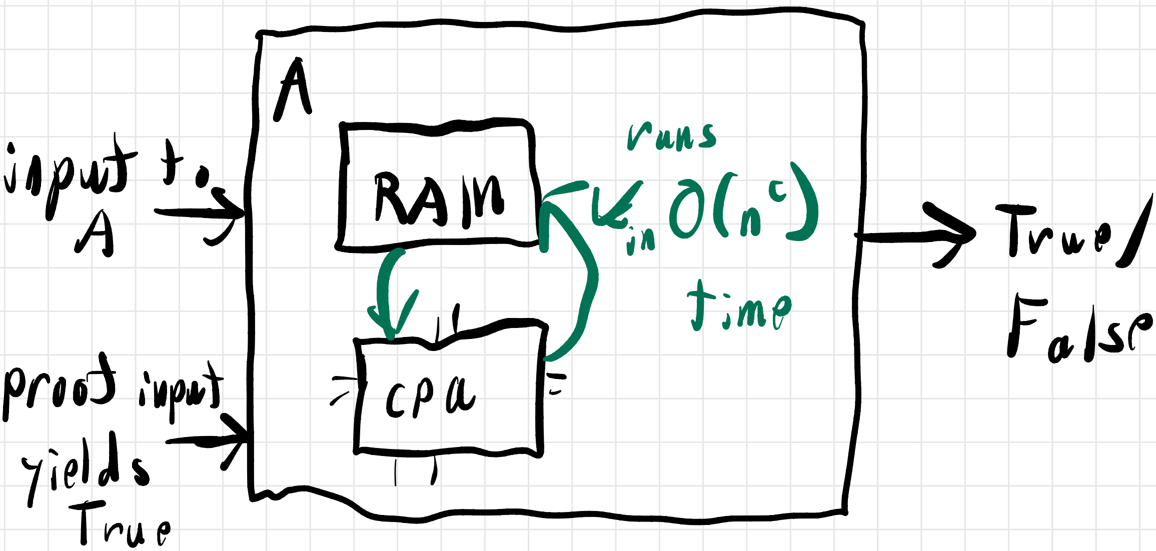
Thm [Cook '71, Levin '73]:

CircuitSAT $\in$ NP-complete

Sketch Proof:

Let $A \in NP$ be any problem in NP.

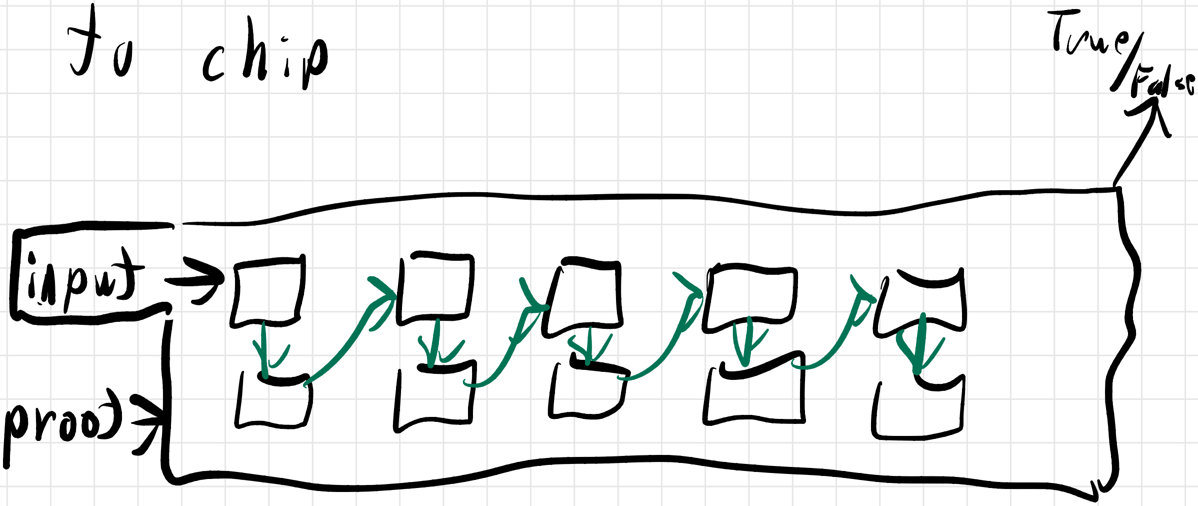
$A \in NP$ implies we can build a little proof verification machine.



Reduce A to Circuit SAT.

Given just <u>input</u>.

Buy a whole bunch of chips. Replace CPU clock with passing info from chip to chip



$O(n^c)$ chips

in a DAG

Machine is now a DAG

Is an instance of CircuitSAT.

of size $m = O(n^c)$.

Reduction took $O(n^c)$ time.

Return CircuitSAT answer in $O(m^{c'})$ time, to get an $O((n^c)^{c'})$ time alg for A. " $O(n^{cc'})$ "

So CircuitSAT is NP-hard.

Also, CircuitSAT $\in$ NP.

So CircuitSAT $\in$ NP-complete.

To prove a problem B is NP-hard, do a polynomial time reduction from some NP-hard problem A <u>to</u> B.

Formula Satisfiability (SAT):

Given a boolean <u>formula</u> like $(a \vee b \vee c \vee \bar{d}) \Leftrightarrow ((b = c) \vee$ not...$)$

Can you set the variables so the formula evaluates to Treue?

SAT $\in$ NP

Thm: SAT $\in$ NP-complete

Proof by reduction from
Circuit SAT...