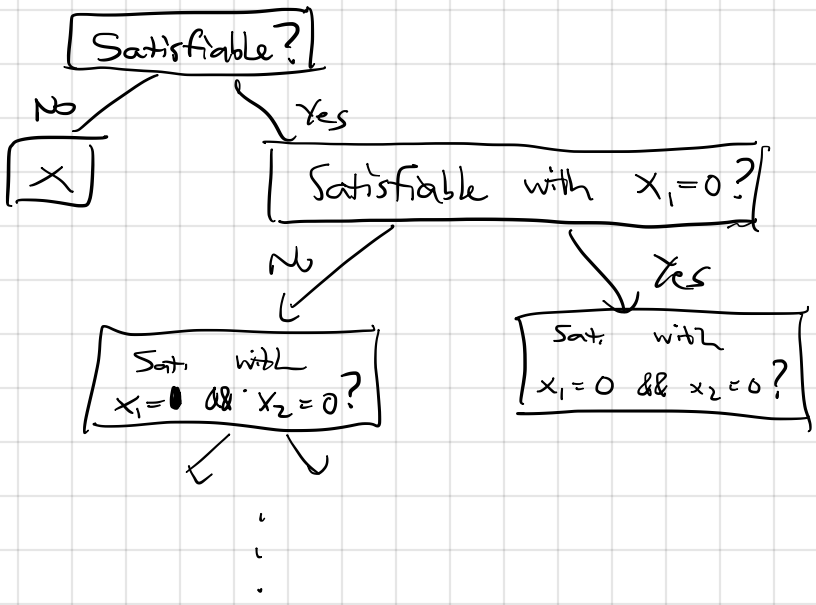
21 March 2018

End of last lecture described a procedure for deciding if an n-bit number is composite, not for factoring it.

Search problems (finding a solution) can in general be tougher than decision problems (does a solution exist?).

CIRCUIT SAT (in its decision version) can be used to solve search problems because CIRCUIT SAT is "self-reducible": deciding whether there exists a satisfying with certain variables having "hard-wired" truth values reduces to another CIRCUIT SAT problem with a smaller # of variables.



By following a single root-to-leaf path, we pose a sequence of n CIRCUIT-SAT questions and use their answers to find a satisfying assignment.

Focus on decision problems from now on.
Decision problem Π is a mapping from input strings x to {'yes', 'no'}.
The following notations are equivalent:

$$\Pi(x) = \text{yes} \equiv x \in \Pi \equiv x \text{ is a 'yes' instance of } \Pi$$

An algorithm for Π is a procedure A such that

$$\forall x \quad A(x) = \Pi(x)$$

A <u>verifier</u> for Π is a procedure V that takes two inputs: x, y.

$$\forall\, x \in \Pi \quad \exists y \quad V(x,y) = \text{yes}$$
$$\forall\, x \notin \Pi \quad \nexists y \quad V(x,y) = \text{yes}$$

Think of y as "evidence" that x is a 'yes' instance of Π. Verifier checks the evidence, and can't be fooled by a wrong piece of evidence.

<u>E.g.</u> In a verifier for CNF-SAT, y is a truth assignment of the variables. V is an algorithm that plugs the truth assignment into every clause and outputs 'yes' if all clauses satisfied.

<u>Notation.</u> $|x|$ means # of bits in the string x.

<u>Definition.</u> A decision problem Π belongs to P if there exists an algorithm A for Π that has worst-case running time $\text{poly}(|x|)$.

Π belongs to NP if there exists a verifier V for Π such that $|y| \leq \text{poly}(|x|)$ and running time of V is also $\text{poly}(|x|)$.

Clearly $P \subseteq NP$. Is $P \neq NP$? Not known.

<u>Definition.</u> A poly-time Karp reduction from Π_0 to Π_1 is a poly-time alg R transforming an input of Π_0, x, into an input of Π_1, $R(x)$, such that

$$\forall x \quad \Pi_0(x) = \Pi_1(R(x)).$$

R is a way of solving Π_0 by rewriting the problem as an instance of Π_1 and then solving Π_1.

If a poly-time reduction from Π_0 to Π_1 exists, we write

$$\Pi_0 \leq_P \Pi_1$$

Def. A problem Π is $\mathcal{NP}$-complete if $\Pi \in \mathcal{NP}$ and $\forall \Pi' \in \mathcal{NP}$ $\Pi' \leq_p \Pi$.

Cook-Levin Theorem. 3SAT is NP-complete.
(Proof deferred until next month.)

Showing other problems are NP-complete.
To show that Π is NP-complete, do two things.

1. provide a verifier: $\Pi \in NP$
2. provide a reduction from 3SAT to Π: $3SAT \leq_p \Pi$

Then, by Cook-Levin, $\forall \Pi' \in NP$, $\Pi' \leq_p 3SAT \leq_p \Pi$

$\leq_p$ is transitive because reductions can be applied in sequence.

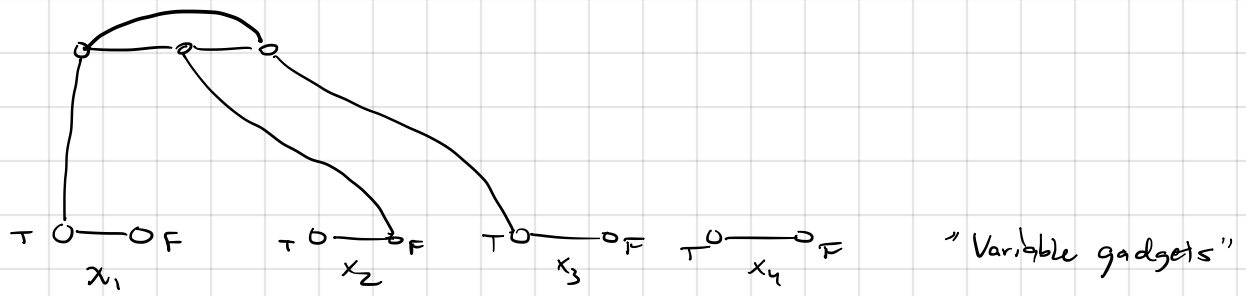
Example. INDEPENDENT SET is the problem where we are given (G, k)
G is an undirected graph, $k \in \mathbb{N}$,
does G contain k distinct vertices with no edge between any two of them?

Belongs to NP because $V(G, k, S)$ checks that $|S| \leq |V(G)|$ && $|S| = k$ and that there's no edge between any 2 elements of S. (S = set of vertices)
This takes $O(\underbrace{m+n}_{\text{read } G \text{ into memory}} + |S|^2)$ ← test for edges btw. S.

Reducing 3SAT to IND SET.

$$(\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge (x_1 \vee x_4)$$

"Gadgets": translate each piece of the problem (3SAT in this case) into pieces of a graph.



"Variable gadgets"