"Efficient" $\equiv$ polynomial time
\[ O(n^c) \] for some constant $c$

![Figure 15.1. An AND gate, an OR gate, and a NOT gate.](image)

Can you set $x_1, \ldots, x_5$ to True or False so circuit outputs True? (Yes for this example)

[Underline]circuit satisfiability (Circuit SAT)
Easy verify a "yes" answer given the correct assignments. Hard to solve from scratch: try all \( O(2^n) \) inputs?
Decision Problems: Any kind of (finite) input, output
True or False (Yes or No) (1 or 0)

Three main classes (for 6363):

\[ \text{Polynomial} \]

\[ P: \text{Decision problems with poly time algorithms.} \]
\[ \text{(Given } G, k, \text{ does } G \text{ have a spanning tree of weight } \leq k?) \]

\[ \text{Non-deterministic} \]

\[ NP: \text{Decision problems where True instances have proofs that can be verified in poly time.} \]
\[ (\text{Circuit SAT}) \]
\[ NP \supseteq P \]
co-NP: Decision problems where \textit{false} instances have a "disproof" that can be verified in poly time.

(Given $n$-bit number $X$, is $X$ prime?)

\textit{\wedge} actually in P!

$P \subseteq \text{NP}$ (if $\text{A} \in \text{P}$, always use "empty proof")

But, does $P = \text{NP}$?

\text{I think } P \neq \text{NP} (P \not\subseteq \text{NP})
Another problem: does \( NP = \text{co-NP} \)?

![Venn Diagram](image)

The world? (no proof)

Problem B (decision or not) is \( NP \)-hard if we can reduce every problem \( A \in \text{NP} \) to an algorithm for B with poly time overhead.

\[
\Rightarrow \text{Any poly time algorithm for B yields poly time alg for all } A \in \text{NP}.
\]
\( \Rightarrow (\text{Poly time alg for } B \Rightarrow P = NP) \Rightarrow \text{no poly time alg for } B \), probably.

\text{NP-complete: in } NP \text{ and NP-hard}

(\text{HALT is NP-hard but not in } NP)

Each NP-complete problem is in \( P \) iff \( P = NP \).
Cook[71], Levin [73];

Circuit SAT is NP-complete.

A ∈ NP,

so we can verify True proofs in poly time.

Verifier for A

instance of A (size n)

proof

uses poly(n) clock cycles

uses poly(n) bits of RAM

True/False
Suppose we're given instance $x$ of $A$, $n=1x$.

a giant boolean circuit
of size $\text{poly}(n)$
(can instance of CircuitSAT!)
answer for $x$ is True iff
if a proof to be verified
iff answer for CircuitSAT
instance is True
so Circuit SAT is NP-hard.
Reduction Arguments:
To prove problem $B$ is NP-hard, take a known NP-hard problem $A$ and reduce $A$ to $B$ with poly time overhead.

**Formula Satisfiability (SAT)**

Given boolean formula of any form. Can you set the variables so whole formula is true?
Reduce from CircuitSAT

Given a circuit...
- add a new variable for the output of each gate
- write a big conjunction of equations describing each gate
- add output of final gate to end of conjunction
The formula is SAT if the circuit is SAT.

**CircuitSAT**

- **Input:** Boolean circuit $K$
- **Transform:** Boolean formula $\Phi$ in $O(n)$ time
- **SAT:**
  - True: $\Phi$ is satisfiable
  - False: $\Phi$ is not satisfiable
- **Output:**
  - True: $K$ is satisfiable
  - False: $K$ is not satisfiable

Time for CircuitSAT is $O(n) +$ time for SAT

$\Rightarrow$ SAT is NP-hard
SAT \in \text{NP} \ (\text{show me how to set the variables})

so \ SAT \ \text{is NP-complete!}