NP Completeness (Intro)

Announcement: today (Mar 18) is the last date to drop classes.

Some problem
you want to
solve (A)

Some problem
you already
can solve (B)

Input to A
Encode

Input to B
Solve

"Karp reduction"

Output of A
Decode

Output of B

When I a Karp reduction from A to B running in polynomial time, we denote that relation as

\[ A \leq_p B \]

and interpret this as "if we can solve B then we can solve A," or "A is at least as easy to solve as B."

Equivalently: If we cannot solve A (efficiently) then we also cannot solve B (efficiently).
The "root of hardness" (one problem assumed by most people to be computationally hard) is SAT. ("Boolean satisfiability")

Given:
- variables \( x_1, \ldots, x_n \) taking \{T,F\} values.
- clauses \( C_1, \ldots, C_m \) each a disjunction of 2 or more literals. ("literal" = "variable or its negation")

\[ C_1 = x_2 \lor \overline{x}_4 \lor x_5 \lor \overline{x}_8 \lor x_9 \]

Question: Does there exist a truth assignment for \( x_1, \ldots, x_n \) that satisfies every clause?

For \( x_1, \ldots, x_n \): does there exist a truth assignment that satisfies every clause?

Example:
\[
\begin{align*}
C_1 &= (x_1 \lor \overline{x}_2) \lor (x_2 \lor x_3) \lor (\overline{x}_3 \lor \overline{x}_1) \\
C_2 &= (x_1 \lor x_2) \lor (x_1 \lor \overline{x}_2) \\
C_3 &= (x_2 \lor x_3) \lor (\overline{x}_2 \lor \overline{x}_3) \\
C_4 &= (x_3 \lor x_1) \lor (\overline{x}_3 \lor \overline{x}_1)
\end{align*}
\]

\( x_1 = T \quad x_2 = T \quad x_3 = F \)

Not satisfiable!

Status of SAT: We don't know any algorithm easier than \( O(2^n - \epsilon n) \).
Even solving in \( O(1.99^n) \) time would be a major breakthrough.

3-SAT: The special case of SAT where each clause has 3 literals.

Known: SAT \( \leq_p \) 3-SAT

Believed: 3-SAT requires \( \geq c^n \) running time for some \( c > 1 \).
**INDEPENDENT SET.**

Given: Graph $G$ (undirected) $\forall k \in \mathbb{N}$

Question: Does $G$ have a set of $k$ vertices with no edge joining any two of them? (Such a vertex set is called an "independent set").

Ex. $G = \begin{tikzpicture}
  \draw[red] (0,0) -- (1,0);
  \draw[red] (1,0) -- (2,0);
  \draw[red] (2,0) -- (3,0);
  \draw[red] (3,0) -- (4,0);
  \draw[red] (4,0) -- (5,0);
\end{tikzpicture}$ $k = 3$ Yes

$G = \begin{tikzpicture}
  \draw[red] (0,0) -- (1,0);
  \draw[red] (1,0) -- (2,0);
  \draw[red] (2,0) -- (3,0);
\end{tikzpicture}$ $k = 3$ No

Claim: INDEPENDENT SET is at least as hard as 3-SAT.

3-SAT $\leq_p$ IND SET

"How can we transform a 3-SAT problem to an independent set problem, so that solving the IND SET instance gives you the answer to 3-SAT?"