Relativization
(why simulation & diagonalization aren't the secrets to ALL happiness)

This is relatively cool!
Oracle = y

Step 1: [Foil M_i]

- Choose no s.t. c_i n_i < 2^n \forall n \geq 0
- Run M_i oracle on input 0^n
- If M_i asks oracle about a string q, oracle answers "No" \(q \notin \text{oracle}\)
- If M_i accepts 0^n, we declare all strings of length no NOT in oracle.
- If M_i rejects 0^n, we declare all unqueried strings of length no to be in oracle.

Step i: [Foil M_i] oracle has some "yes" and some "No"

- Choose no s.t. c_i n_i < 2^n \forall n \geq 0
- Run M_i oracle on input 0^n
- If M_i asks about a string already determined to be in or out of oracle, answer consistently
- If M_i asks about a string q of length no, oracle answers "No"
- If M_i accepts 0^n...
So now what?
# P

\[ \#3SAT = \{ <C>, k \mid 3\text{CNF instance } C \text{ has exactly } k \text{ satisfying assignments} \} \]

- # Vertex Cover
- # Clique

what does it mean to be \#P complete?
Some Possible Objections to #P

1. "Answering those kinds of questions is dumb! They are only hard because there are an exponential number of possible solutions!"

2. "#3SAT is only hard because 3SAT is hard."
   "My favorite polytime problem" wouldn't be hard.
# Matching (aka "Permanent")

\[
A = \begin{pmatrix}
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

\[
\text{perm } A = \sum_{\pi} \prod_{i=1}^{n} A_{\pi(i), i}
\]
Interactive Proofs (IP)

NP

Prover
unlimited power

poly certif

\[ C \]

Verifier
poly-time limit

\[ \text{Verify}(w, c) \]

\[ \leftarrow \text{accept/reject} \]

IP

Prover
unlimited power

poly size communications

\[ \leftarrow m_1 \]

\[ m_2 \]

\[ \vdots \]

\[ \leftarrow \text{accept/reject} \]

Verifier
poly-time limit

\{ polynomial (in n) interactions \}

+ random bits available to verifier

1. If \( w \in L \), verifier accepts with "high probability" (1-\( \delta \))
2. If \( w \not\in L \), verifier...
A "Hard" Problem in IP

Graph Isomorphism: $G_1, G_2$

Graph NON Isomorphism: $G_1, G_2$

Claim 1: GNI $\in$ IP

Claim 2: $\#3SAT \in$ IP
\#SAT ∈ IP

(Attempt #1)

- Let the variables be $x_1, \ldots, x_m$.

- Let $f_i(x_1, \ldots, x_i)$ be a function where $f_i(b_1, \ldots, b_i) =$ number of sat. valuations when $x_1 = b_1, \ldots, x_i = b_i$.

- Consider, for example, ...

  $$f_m(0, 1, 1, 0, \ldots, 1)_{m}$$

  $f_1(0)$

  $f_0()$

- Claim: $f_i(b_1, \ldots, b_i) = f_{i+1}(b_1, \ldots, b_i, 0) + f_{i+1}(b_1, \ldots, b_i, 1)$.
The "Protocol" (It's not quite right):

on input \( \langle I \rangle, k \) satisfying assignments exactly

Step 0:

Prover (P) sends \( f_0() \) to Verifier (V)

V checks that \( f_0() = k \). If not, reject.

Step 1:

Prover tries to show V that \( f_0() \) was correct.

P sends V \( f_1(0), f_1(1) \).

V checks that \( f_0() = f_1(0) + f_1(1) \).

If not, reject.

Step 2:

Prover tries to show V that \( f_1() \) was correct.

P sends V \( f_2(0,0), f_2(0,1), f_2(1,0), f_2(1,1) \).

V checks that...

If not, reject.

\vdots

Step m:

Prover tries to show V that \( f_m(...) \) was correct.

P sends V \( f_m(0,...,0), \ldots, f_m(1,...,1) \).

V checks that...

If not, reject.

Step m+1: Rubber hits the road!

V checks that each \( f_m(...) \) is correct.