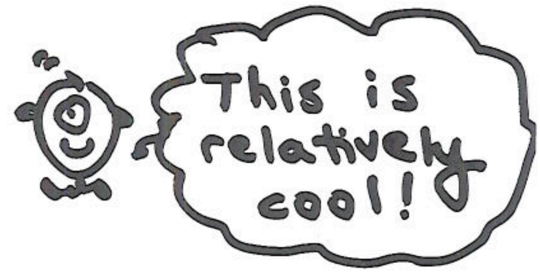
Relativization

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



oracle $= \emptyset$

Step 1 : [Foil M_1]

- Choose n_0 s.t. $c_1 n^{p_1} < 2n \ \forall n \geq n_0$
- Run M_1^{oracle} on input 0^{n_0}
- If M_1 asks oracle about a string q, oracle answers "No" ($q \notin$ oracle)
- If M_1 accepts 0^{n_0} we declare <u>all</u> strings of length n_0 NOT in oracle.
- If M_1 rejects 0^{n_0} we declare all unqueried strings of length n_0 to be in oracle.

Step i: [Foil M_i] oracle has some "yes" and some "No" strings from prev. steps

- Choose n_0 s.t. $c_i n^{p_i} < 2n \ \forall n \geq n_0$

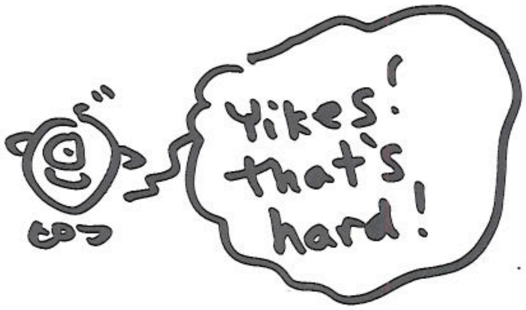
AND n_0 longer than any string previously queried!

- Run M_i^{oracle} on input 0^{n_0}
- 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 n_0, oracle answers "No
- If M_i accepts 0^{n_0}...
- If M_i reject 0^{n_0}...

so now what? 

#P

$\#3SAT = \{ \langle C \rangle, k \mid$ 3CNF instance C has exactly k satisfying assignments$\}$



#Vertex Cover

Clique

What does it mean to be #P complete?

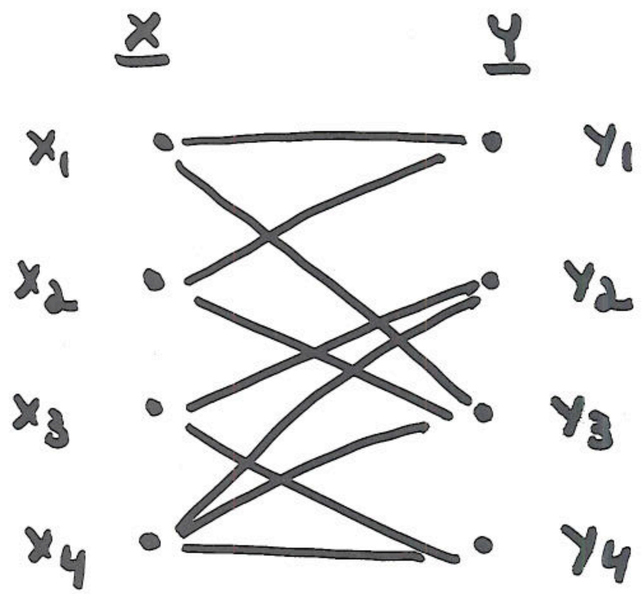
Some Possible Objections to #P

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")



$$A = \begin{array}{c|cccc} & y_1 & y_2 & y_3 & y_4 \\ \hline x_1 & & & & \\ x_2 & & & & \\ x_3 & & & & \\ x_4 & & & & \end{array}$$

$$A = \begin{array}{cc} & \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} & \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

$$\text{perm } A = \sum_{\pi} \prod_{i=1}^{n} A_{i,\pi(i)}$$

$1 \to 1$
$2 \to 3$
$3 \to 2$
$4 \to 4$

Interactive Proofs (IP)

$w \in L$

NP

Prover		Verifier
Unlimited power	poly certif $\overbrace{c}$ $\longrightarrow$	poly-time limit; Verify(w, c)
	$\longleftarrow$ accept/reject	

IP

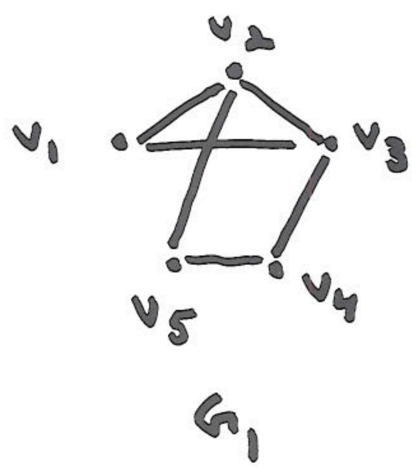
Prover		Verifier
Unlimited power	poly size communications	poly-time limit
	$\longleftarrow m_1$	
	$m_2 \longrightarrow$	
	$\vdots$	
	$\longleftarrow$ accept/reject	

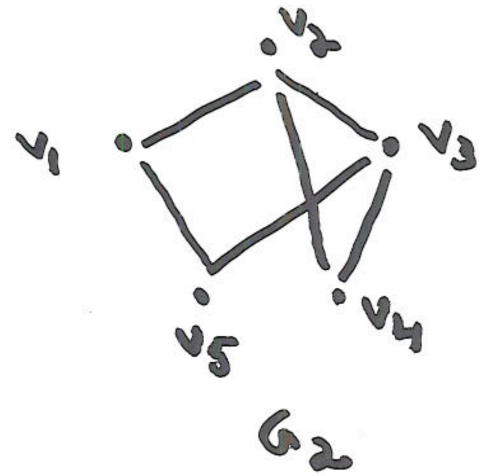
polynomial (in n) interactions

+ $\boxed{|||||}$ random bits available to verifier

1. If $w \in L$, Verifier accepts with "high probability" $(1-\epsilon)$
2. If $w \notin L$, verifier...

A "Hard" Problem in IP





(GI) Graph Isomorphism: G_1, G_2

(GNI) Graph NON Isomorphism: G_1, G_2

Claim 1: GNI $\in$ IP

Claim 2: #3SAT $\in$ IP

#SAT $\in$ IP

(Attempt #1)

- Let the variables be $x_1, \dots, x_m$
- Let $f_i(x_1, \dots, x_i)$ be a function where $f_i(b_1, \dots, b_i) \triangleq$ number of sat. valuations when $x_1 = b_1, \dots, x_i = b_i$
- Consider, for example, ...

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

$$f_1(0)$$

$$f_0()$$

- Claim: $f_i(b_1, \dots, b_i) =$

$$f_{i+1}(b_1, \dots, b_i, 0) + f_{i+1}(b_1, \dots, b_i, 1)$$

The "Protocol" (It's not quite right!)

on input $\langle I \rangle, K$

variables $x_1, \dots, x_m$

K satisfying assignments exactly

total length is n

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 $f_1(\cdot)$ 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!

⋮

Step m: Prover tries to show V that $f_m(\cdots)$ was correct...

P sends V $f_m(0,\dots,0), \dots, f_m(1,\dots,1)$

V checks that...

If not, reject!

Step m+1: Rubber hits the road!

V checks that each $f_m(\cdots)$ is correct