PSPACE: space used (in turing machine) \( \leq p(1^k) = \text{polynomial} \)
so \( \text{WLOG, time} \leq 2^{O(p(n))} \) \( 2^{O(n)} \)

IP: protocol for deciding whether \( x \in L \) (language) \( x \in \{0,1\}^n \)
- \( V \)-verifier - p polynomial time probabilistic algorithm, public coins
- \( P \)-prover - WLOG a function

\( V \) is a turing machine with a special communication tape.
\( V \) runs, writes \( \text{q} \) on its communication tape, \( P \) responds with \( P(b_1, \ldots, b_i, r_1, \ldots, r_i) \). At some point \( V \) must accept/reject.

If \( V,P \) is an IP for \( L \):
- if \( x \in L \), \( \Pr[V \text{ accepts}] = 1 \) (>90%).
- if \( x \notin L \), then for all \( p \) (evil provers) \( \Pr[V,q \text{ working w/ } p \text{ accepts}] \leq 1/2^n \).

(Think of ourselves as \( V \))

Show IP \( \leq \) PSPACE - exhaustively branch on random sequences

Show \( \text{IP} \leq \text{PSPACE} \)

First, a proof that doesn't work: switch divide-and-conquer

- PSPACE computation
  - initial state of a Turing machine, exponentially long
  - \( q \)-accept/reject

\( V \): ask \( P \) for \( 1/2 \) config, \( P \) gives it
\( V \): flip 0,1 at random; if \( v \approx 0 \), ask \( 1/4 \) config; if \( v \approx 0 \), ask \( 3/4 \) config

At the end, get adjacent config to check if they follow

But an evil prover can pick some random point, and
switch to a computation of something in the language
- unless \( V \) happens to land on the switch point, \( V \) will be tricked.
Correct proof: basically we need more information from the proof.

Quantified Boolean Formulas (QBF)

QBF is PSPACE-complete

Variable $x_1, x_2, \ldots, l, v, z$.

$\forall x_1 \exists x_2 \forall x_3 ((x_1 \land x_2) \land x_3)$

It will suffice to prove QBF ∈ IP

Idea: a QBF via a polynomial:

(a) WLOG, negatives are only on variables (De Morgan: $\lnot (\psi_1 \land \psi_2) = \psi_1 \lor \lnot \psi_2$)

(b) For boolean $x_i$, we have numeric (integer) $8_i \in \{0, 1\}$

$x_i \mapsto 0_i := 8_i \mod 2$ (i.e., for $8 \in \mathbb{N}$ numeric vars 1/poly $\land x_i \mapsto 0 \equiv \overline{x}_i$. 0 is false $> 0$ in true)

$b = \oplus x_i : 2^n \land \lnot \psi$.

We don't do simplification.

$8 \cdot 8 = 8 \cdot 8 = 8 \cdot 8$. 8. 8. 8.

8 - polynomial one $N$, written compactly. Suppose $\psi$ has $|\psi|$

symbols, $\deg (8_{\psi}) = 2^{\|m\|}$ if no $V \in N$, then $\deg (8_{\psi}) 2^{\|m\|}$.

$
\deg (8_{\psi}) = 2^{\|\psi\|}$

If $\psi$ has a free variable $x$, $\deg_x (8_{\psi}) = 2^{\|N, x\|}$ in the scope of $x$).

Goal: (Evil) $P$ wants to convince $V$ that $8_{\psi} > 0$, i.e., $\psi$ is true

(for $\psi$ a sentence - all vars bound).

Send $8_{\psi}$ to $Z_p$, $|Z_p| = 2^{\|\psi\|}$, such $Z_p = B$.

First step: $P$ will pick prime $p$, send to $V$. New goal: convince $V$ $\psi$ true.

If $8 \equiv 0 \mod p$,

WLOG $p$ can be $2^{\|\psi\|}$ by CRT and Prime $\in$ theorem.

$(P$ sends a Pratt certificate of the primality of $p$)

Suppose $\psi : \forall x \exists z \forall x \exists z \ldots$ subformula $\phi : \forall x \exists z \forall x \exists (x, x)$

$8_{\psi} = 8_{\psi} (z_1, z_2)$. $\phi$ free in $\psi$

$\deg_x (8_{\psi}) = 2^{\|m\|}$, want $\deg (8_{\psi}) = 2^{\|m\|}$

Trick: WLOG $\psi$ is originally in prenex form, but will be transformed to a nearly prenex form.

Say $\phi$ is simple if for every subformula $\psi$, $x$ is free

$\psi$, $x$ appears in the scope of at most one $x$ in $\psi$.

$\deg_x (8_{\psi}) = 2^{\|\psi\|} \leq 2^{\|m\|}$
Theorem: Consider a subformula of $\varphi$ of form

$$\forall x_2 \chi(x_1, x_3, x_4)$$

replace by $\forall x_2 \exists x_1 \exists x_1 \ldots \exists x_{k-1} \chi(x_1, x_3, x_4)$.

Perform transformations to $\varphi$ in parallel but keep names $x_2, x_3, \ldots, x_{k-1}$.

Then, if $\varphi$ is QBF, $\exists \varphi'$ formed as above s.t. $deg(\psi) \geq (\varphi)$ for

subformulas $\chi$ of $\varphi'$, $x$ free in $\psi$, $(\varphi) = \Omega(\varphi)^2$.

IP-protocol:

Step 0: If is WLOG simple and nearly prefix.

Step 1: $P$ tells $V$ a prime $p$ (with certificate), a value $b \in \mathbb{Z}_p$, $P$ claims $b^p \equiv b \mod p)$, $1 \leq b \leq p^2$. $V$ rejects if any of these don't hold.

Step 2: We have a subformula $\chi$ of $\varphi$, value $b$, $\varphi \equiv \chi(x_1, x_2)$

values $x_1, x_2 \in \mathbb{Z}_p$ and $P$ is claiming $\varphi(x_1, x_2)$.

Step 3: If $\varphi$ is quantifier-free, then $V$ evaluates $\chi$ and accepts if $\text{eval}(\chi) = b$ or not

(a) $\varphi$ is $\forall x_1 \chi$. $P$ asks $P$ for a $\log(b) + 2(\varphi)$ policy over $\mathbb{Z}_p$.

(b) $\varphi$ is $\exists x_1 \chi$ where $x_1$ is quantifier-free. $V$ checks that $\text{eval}(\chi) = b$, $P$ not $V$ rejects. Otherwise $V$ chooses $x_1 \in \mathbb{Z}_p$ at random, compute $b' = \chi(x_1)$, set $b = b'$, $\psi = \chi$, and go to next step. (I.e., $P$ was claiming

$\chi(x_1) = b \equiv \chi(x_1) \mod p$.

(c) $\exists x_1 \chi$, name, but use $\text{eval}(\chi) = b$.}