Model Definition

Verifier (function $V$): three inputs
1. Input string $w$: decide $w \in A$ or not.
2. Random input: like probabilistic choice (bits from coin flips)
3. Message history: new choice based on past dialog
   $m_1#m_2#\ldots#m_i$

Output: next message $m_{i+1}$ (to prover), or accept, or reject
$V \overset{m_{i+1}}{\leftarrow P}$ (odd messages)

Prover (function $P$): takes
1. input $w$
2. message history $m_1#m_2#\ldots#m_i$

Output next message $m_{i+1}$
$V \overset{p_{i+1}}{\rightarrow P}$ (even messages)

Denote this interaction by $V \leftrightarrow P$.

Defining Outcome

Def. A language $A$ is in $\text{IP}$ if there exists a verifier (polynomially computable function) $V$ such that for every string $w$
1. for some function $P$, $w \in A \implies \Pr[V \leftrightarrow P \text{ accepts } w] \geq \frac{2}{3}$
2. for any function $P$, $w \notin A \implies \Pr[V \leftrightarrow P \text{ accepts } w] \leq \frac{1}{3}$
   - some (honest) prover $P$ can produce likely correct accept
   - no (dishonest?) prover $P$ can produce likely incorrect accept

Can use amplification to make error probability arbitrarily small.

$\text{BPP} \subseteq \text{IP}$ $\text{BPP} \subseteq \text{IP}$ (need no $P$/ignore)

$\text{NP} \subseteq \text{IP}$ $\text{NP} \subseteq \text{IP}$ (never wrongly accepts, try often enough)

We’ll prove $\text{IP} = \text{PSPACE}$ (Shamir’s Theorem)

Motivation

- $\text{NP}$ is based on polynomial-time verifiers (short witness)
  limited capacity of verifier
- Think of two entities:
  - $\text{Prover}$ produces witness (unlimited power; may be hard to find)
  - $\text{Verifier}$ checks witness (must be efficient)
- Assymmetry of YES vs. NO
  $\text{NP}$: easy check for YES, often no easy check for NO
  $\text{coNP}$ probably (?) different from $\text{NP}$
- Interactive Proofs: give more power to verifier
  - two-way dialog
  - allow probabilistic conclusion
  - but: prover may be dishonest, verifier must cope

Graph (Non)Isomorphism

Natural problem, complexity unknown: in $\text{P}$ (?), $\text{NP}$-complete (??)
$\text{ISO} = \{(G_1, G_2) \mid G_1$ and $G_2$ are isomorphic graphs\}$
$\text{NONISO} = \{(G_1, G_2) \mid G_1$ and $G_2$ are not isomorphic graphs\}$

Verifier chooses one of $G_1$, $G_2$, reorders nodes into $H$.
Sends to Prover, asks to tell if $G_1$ or $G_2$.

IP $\subseteq$ PSPACE

Simulate an interactive proof in polynomial space

Assume: $p = p(n)$ messages of length $\leq p(n)$ exchanged
Choose prover maximizing accept probability for input $w$
$\Pr[V \text{ accepts } w] = \max_P \Pr[V \leftrightarrow P \text{ accepts } w]$

At least $\frac{2}{3}$ for $w \in A$, at most $\frac{1}{3}$ for $w \notin A$

Parameterize interaction with initial message sequence $M_j = m_1#m_2#\ldots#m_j$
Consider probability $\Pr[V \leftrightarrow P \text{ accepts } w \text{ starting at } M_j]$ over all random strings $r$ consistent with $M_j$.
Define:
$\Pr[V \text{ acc. } w \text{ start at } M_j] = \max_P \Pr[V \leftrightarrow P \text{ acc. } w \text{ start at } M_j]$
Computing Accept Probability, Bottom-Up

Why choose max prover for both accept and reject?
- best case for accept (want some proof of acceptance)
- worst case for reject (max. chance to deceive)

Compute values starting with complete histories $M_p$ of $p$ messages.

$N_{M_p} = 1$ if $m_p = \text{accept}$ and $M_p$ consistent with some random $r$
$N_{M_p} = 0$ otherwise

$N_{M_j}$ = \[ \begin{cases} \max_{m_{j+1}} N_{M_{j+1}} & j \text{ odd (prover’s turn)} \\ \text{wt-avg} m_{j+1} N_{M_{j+1}} & j \text{ even (verifier’s turn)} \end{cases} \]

weighted average of $N_{M_{j+1}}$ by probability of verifier sending $m_{j+1}$ (eliminate random values $r$ causing output inconsistent with $M_j$)

Claim: $N_{M_0} = Pr[V \text{ accepts } w]$