Basic Properties of IP

• Definition (IP): For an integer $k \geq 1$ (may depend on the input length), say a language $L$ is in IP$[k]$ if $\exists$ a probabilistic poly-time TM $V$ that can have a $k$-round interaction with any function $P: \{0,1\}^* \rightarrow \{0,1\}^*$ s.t.

- $x \in L \Rightarrow \exists P, \Pr[\text{Out}_V < V, P>(x)=1] \geq 2/3$ (completeness)
- $x \not\in L \Rightarrow \forall P, \Pr[\text{Out}_V < V, P>(x)=1] \leq 1/3$ (soundness)

• Define $\text{IP} = \bigcup_{c \geq 0} \text{IP}(n^c)$
Basic Properties of IP

- Definition (IP): say a language L is in IP[k] if \(\exists\) a probabilistic poly-time TM V that can have a k-round interaction with any function \(P: \{0,1\}^* \rightarrow \{0,1\}^*\) s.t.

  \[x \in L \Rightarrow \exists P, \Pr[Out_V <V, P>(x)=1] \geq 2/3\] (completeness)

  \[x \notin L \Rightarrow \forall P, \Pr[Out_V <V, P>(x)=1] \leq 1/3\] (soundness)

- The class IP does not change if we replace 2/3 by 1/2+1/n^c and replace 1/3 by 1/2-1/n^c.

- The class IP does not change if we replace 2/3 by 1-2^{-n^c} and replace 1/3 by 2^{-n^c}.

- Idea: the verifier repeats the protocol for some \(m=poly(n)\) times using independent random bits, then accept \(x\) iff more than 1/2 runs result in an accept.
Basic Properties of IP

• The class IP does not change if we replace $2/3$ by $1-2^{-n^c}$ and replace $1/3$ by $2^{-n^c}$.

• Idea: the verifier repeats the protocol for some $m=poly(n)$ times using independent random bits, then accept $x$ iff more than $1/2$ runs result in an accept.

• If $x \in L$, then can use the same $P$. Since each time $Pr[\text{accept}] \geq 2/3$ the expected number of acceptance is at least $2m/3$.

• By a Chernoff bound $Pr[\text{accept} < m/2 \text{ runs}] \leq 2^{-\Omega(m)}$. 
Basic Properties of IP

• The class IP does not change if we replace $2/3$ by $1-2^{-n^c}$ and replace $1/3$ by $2^{-n^c}$.

• Idea: the verifier repeats the protocol for some $m=\text{poly}(n)$ times using independent random bits, then accept $x$ iff more than $1/2$ runs result in an accept.

• If $x \in L$, then can use the same $P$. Since each time $\Pr[\text{accept}] \geq 2/3$ the expected number of acceptance is at least $2m/3$.

• By a Chernoff bound $\Pr[\text{accept} < m/2 \text{ runs}] \leq 2^{-\Omega(m)}$. 
Basic Properties of IP

The Normal Distribution

\[ f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]
Basic Properties of IP

• Allowing prover to be probabilistic does not change IP.

• Replacing 2/3 by 1 does not change IP. This is non-trivial.

• Replacing 1/3 by 0 is equivalent to having a deterministic verifier, thus IP becomes NP.

• Can reduce the error probability using parallel repetitions (instead of sequential repetitions).
Basic Properties of IP

• The prover can be all powerful, but is it really necessary?

• Turns out it is enough to use a prover in PSPACE.

• Since a PSPACE prover can compute the optimal answers.

• In other words, we show $\text{IP} \subseteq \text{PSPACE}$. 
\[ IP \subseteq \text{PSPACE} \]

- Show \( \forall L \in IP, L \in \text{PSPACE} \).

- \( \forall x, \text{ we give a PSPACE algorithm to decide if } x \in L \).

- Recall the definition of IP.
\( \textit{IP} \subseteq \text{PSPACE} \)

- A language \( L \) is in \( \text{IP} \) if \( \exists \) a probabilistic poly-time TM \( V \) that can have a poly\((n)\)-round interaction with any function \( P: \{0,1\}^* \to \{0,1\}^* \) s.t.
  
  \[ x \in L \Rightarrow \exists P, \Pr [\text{Out}_{V < V, P}(x)=1] \geq 2/3 \text{ (completeness)} \]
  
  \[ x \not\in L \Rightarrow \forall P, \Pr [\text{Out}_{V < V, P}(x)=1] \leq 1/3 \text{ (soundness)} \]

- The \( \text{PSPACE} \) algorithm will compute the optimal prover \( P \) and the maximum probability \( \Pr [\text{Out}_{V < V, P}(x)=1] = P_{\text{max}}, \) given \( V \)'s poly-time algorithm.

- If \( P_{\text{max}} \geq 2/3 \) then accept, otherwise reject.
$\textbf{IP} \subseteq \textbf{PSPACE}$

- Recursive algorithm similar to TQBF, starting with $i=0$.

- At step $2i+1$, enumerate all possible $a_{2i+1}$ and record $\Pr[a_1, \ldots, a_{2i+1}]$, using the same space.

- For each $a_{2i+1}$, enumerate all possible $a_{2i+2}$, and recursively compute the max probability of acceptance (using the same space).

- Choose the max one as the correct $a_{2i+2}$. 
**IP ⊆ PSPACE**

- Choose the max one as the correct $a_{2i+2}$, and this gives $P_{\max}[a_1, \ldots, a_{2i+1}]$.

- Now compute the sum $\Sigma P_{\max}[a_1, \ldots, a_{2i+1}]$ over all possible $a_{2i+1}$. Return to previous level.

- Base case: at the $k$-th message, count the corresponding probability iff $V$ accepts.

- Each level of recursion uses an additional poly(n) space.