Probabilistic Computation

Probabilistic TM

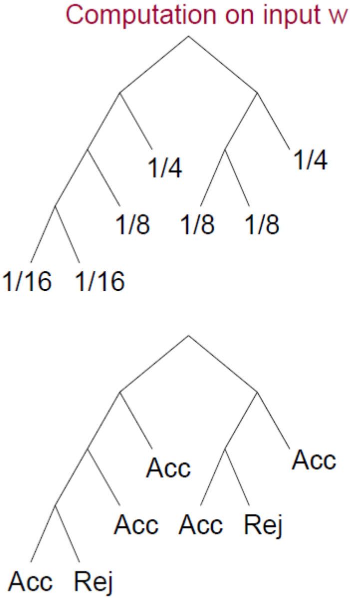
Definition 1

A probabilistic TM (PTM) is a 7-tuple $M = (Q, \Sigma, \Gamma, q_0, F, \delta_1, \delta_2)$, where $Q, \Sigma, \Gamma, q_0, F$ are the same as those in classical Turing machines and δ_1, δ_2 are two deterministic transition functions, such that at each step, the TM applies either the transition function δ_1 with prob. $\frac{1}{2}$ or the transition function δ_2 with prob. $\frac{1}{2}$, resembling a coin flip.

- We may think of transitions as being selected randomly, with equal probability of 0.5, i.e., the PTM flips a fair coin in each step.
- PTMs therefore are very similar to NTMs with (at most) two options per step.

Probabilistic TM (Cont'd)

- Probability of acceptance = $\sum_{accepting\ path\ \sigma} Prob(\sigma)$
- Probability of rejection = $\sum_{rejecting\ path\ \sigma} Prob(\sigma)$
- Example:
  - Prob. Acceptance = $\frac{1}{16} + \frac{1}{8} + \frac{1}{4} + \frac{1}{8} + \frac{1}{4} = \frac{13}{16}$
  - Prob. Rejection = $\frac{1}{16} + \frac{1}{8} = \frac{3}{16}$
- We consider TMs that halt (either accept or reject) on every branch - deciders.
- So the two probabilities total 1.

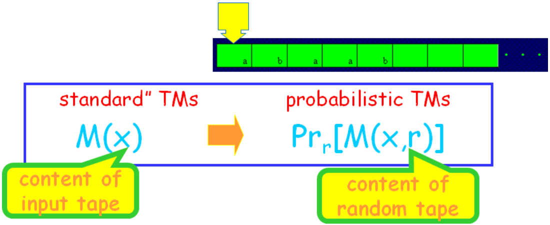


Alternative Definition of PTMs

Definition 2

A PTM is a deterministic TM that receives two inputs x and r, where $x \in \Sigma^*$ is an input word, and $r \in \{0,1\}^*$ is a sequence of random numbers placed on a *read-only random tape*. If (x, r) is accepted, we may call r a witness for x.

Note the similarity to the notion of polynomial verifiers used for NP.



A string $r \in \{0,1\}^n$ is associated with probability $\frac{1}{2^n}$.

How to Define Acceptance for PTMs?

Definition 3

$$Prob[M \text{ accepts } w] = \sum_{b \text{ is an accepting path}} prob(b)$$

$$Prob[M \text{ rejects } w] = 1 - Prob[M \text{ accepts } w]$$

A natural way to define acceptance is based on the notion of "**majority**".

Definition 4

Given a PTM M and an input w, $w \in L(M)$ iff Prob[M accepts w] $> \frac{1}{2}$.

Definition 5

For $0 \leq \epsilon < 1/2$, M accepts w with *error probability* ϵ (written as $e_M(w) = \epsilon$) if

- $w \in L(M)$ implies $Prob[M \text{ accepts } w] \geq (1 - \epsilon)$

In the above definition, ϵ depends on w. E.g., $e_M(w) = \frac{1}{2}(1 - \frac{1}{|w|})$.

Bounded Error Probability

- The problem with the previous definition of acceptance is that $Prob[M \text{ accepts } w]$ $(> \frac{1}{2})$ could be arbitrarily close to $\frac{1}{2}$, making the difference between $Prob[M \text{ accepts } w]$ and $Prob[M \text{ rejects } w]$ arbitrarily small.
- In some applications, we prefer a "gap" between the probabilities of acceptance and rejection, which leads to the notion of "bounded error probability".

Definition 6

A PTM M is with bounded error prob. if $\exists \epsilon < \frac{1}{2}$, for all w.

- $w \in L(M)$ implies $Prob[M \text{ accepts } w] \geq (1 - \epsilon)$
- $w \notin L(M)$ implies $Prob[M \text{ rejects } w] \geq (1 - \epsilon)$

Note that in the above, ϵ does not depend on w.

Polynomial-Time PTMs

- We are mainly interested in PTMs that run in polynomial time. There is still an issue: polynomial in what sense (Worst-case vs. Average-case)?
- Recall that for an NTM M,
  1. $w \in L(M)$: $\exists$ a computation leading to acceptance, while the rest of the computations may lead to rejection.
  2. $w \notin L(M)$: all computations lead to rejection.
- Acceptance for classical NTMs allows *one-sided error*. See (1) above. It does not make much sense for NTMs to have two-sided error. Why?
- For PTMs, we consider both *one-sided* and *two-sided errors*.

Probabilistic TMs Accepting r.e. Sets

Theorem 7

Every r.e. set is accepted (under Def. 4) by some PTM with finite average running time.

Proof.

Let W be an r.e. set and let M be a DTM accepting W. Construct the following PTM M'

1. repeat
2. simulate one step of $M(x)$
3. if $M(x)$ accepted at last step then accept
4. until cointoss()="heads"
5. if cointoss()="heads" the accept else reject

Clearly if $x \notin W$, M' terminates only at line 5. In this case, the prob= $\frac{1}{2}$, so $x \notin L(M')$. If $x \in W$, prob = Prob(exits line3) + $\frac{1}{2} > \frac{1}{2}$.
What is the average running time? (Hint: Consider $\sum_{n=1}^{\infty}(n \times 2^{-n})$) $\square$

One-Sided Error: The classes RP and coRP

We write $M(x) = 1$ (resp., =0) for M accepts (resp., rejects) x.

Definition 8

A language $L \in$ RP (Randomized Polynomial Time), iff a probabilistic Polynomial-time TM M exists, such that

- $x \in L \Rightarrow \text{Prob}(M(x) = 1) \geq \frac{1}{2}$
- $x \notin L \Rightarrow \text{Prob}(M(x) = 0) = 1$ (or equivalently $Prob(M(x) = 1) = 0$)

Definition 9

A language $L \in$ co-RP, iff a probabilistic Polynomial-time TM M exists, such that

- $x \in L \Rightarrow \text{Prob}(M(x) = 1) = 1$
- $x \notin L \Rightarrow \text{Prob}(M(x) = 0) \geq \frac{1}{2}$

These two classes complement each other, i.e., coRP = $\{\bar{L} \mid L \in \ RP\}$.

Comparing RP with NP

- Let R_L be the relation defining the witness/guess for L for a certain TM.
- NP:
  - $x \in L \Rightarrow \exists y, (x, y) \in R_L$
  - $x \notin L \Rightarrow \forall y, (x, y) \notin R_L$
- RP:
  - $x \in L \Rightarrow Prob((x, r) \in R_L) \geq \frac{1}{2}$
  - $x \notin L \Rightarrow \forall r, (x, r) \notin R_L$
- *RP* corresponds to the so-called "Monte-Carlo Algorithm"

Theorem 10

$P \subseteq RP \subseteq NP$ and $P \subseteq co\text{-}RP \subseteq co\text{-}NP$

A Primality Testing Algorithm in co-RP

- Recall *Fermat's Little Theorem*: For prime p, $\forall a$

$$a^{p-1} \equiv 1 \; mod \; p.$$

  Hence, if $\exists \, 2 \leq a \leq p-1$ such that $a^{p-1} \not\equiv 1 \; mod \; p$, p is definitely **composite**.
- However, there exists composite integer n such that $b^{n-1} \equiv 1 \; mod \; n$ for all b with $gcd(n, b) = 1$. Such numbers are called *Carmichael Numbers*.
- Hence, if Fermat test returns "composite", the number is composite; it could return "prime" (i.e., passing the test) even if the number if composite.
- Fermat test is a co-RP algorithm for primality testing.
- A more sophisticated primality testing (co-RP) algorithm is the Miller-Rabin primality test.

Theorem 11 (Agrawal-Kayal-Saxena, 2002)

Primality testing is in P.

Polynomial Identity Testing

Definition 12 (PIT)

Determine if two multi-variable polynomial functions f and g are equal, i.e., have the same results on all inputs

- Challenge: The polynomials are not given in their normal form (as a sum of monomials ($2x^2y^3z$). E.g., $(x_1 + y_1)(x_2 + y_2)\ldots(x_n + y_n)$ has 2^n monomials.
- PIT is equivalent to testing *"Zero Polynomial"* (ZEROP) (i.e., =0 on all inputs) by considering $f - g$.

Polynomial Identity Testing

Lemma 13

(Schwartz-Zippel Lemma): Consider a non-zero multivariate polynomial $p(x_1, \ldots, x_m)$ of total degree $\leq d$, and a finite set S of integers. If $a_1, \ldots, a_m$ are chosen randomly (with replacement) from S, then

$$Prob\left[p(a_1, \ldots, a_m) = 0\right] \leq \frac{d}{|S|}.$$

Consider the following algorithm: For polynomial $P(x_1, \ldots, x_m)$,

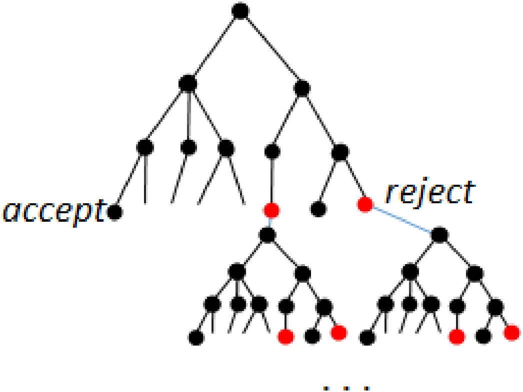
1. Randomly select $a_1, \ldots, a_m \in \{1, \ldots, 3 \times 2^n\}$. Note: $1 - \frac{2^n}{3 \times 2^n} = \frac{2}{3}$.
2. Evaluate the polynomial to compute $p(a_1, \ldots, a_m)$
3. Accept if $p(a_1, \ldots, a_m) = 0$ and reject otherwise.

- If $p \in \text{ZERO}P$, the algorithm will always accept. Otherwise, if $p \notin \text{ZERO}P$, it will reject with probability $\geq \frac{2}{3}$.
- (Problem?) if the degree of the polynomial is as high as 2^n, then the output can be as high as $(3 \times 2^n)^{2^n}$, requiring $O(2^n)$ bits to store!
- (Fix) Use modulo arithmetic.

Amplification

- The constant $\frac{1}{2}$ in the definition of RP is arbitrary.
- If we have a probabilistic TM M that accepts $x \in L$ with probability $p < \frac{1}{2}$, we can run this TM several times to "amplify" the probability.
- 1. Run M on x
  2. if a run leads to acceptance (with prob. p), accept.
  3. if a run leads to rejection (with prob. $1 - p$), Repeat (1).
  4. Exit if (1) is repeated n times.
- If $x \notin L$, all runs will return 0.
- If $x \in L$, and we run it n times than the probability of acceptance is Prob$(M_n(x) = 1)$ = 1-Prob$(M_n(x) \neq 1)$ = 1-Prob$(M(x) \neq 1)^n$ = 1-(1-Prob$(M(x) = 1))^n$ = $1 - (1 - p)^n$

Amplification



Robustness of RP

Definition 14

$L \in RP_1$ iff $\exists$ probabilistic Poly-time TM M and a polynomial $p(.)$, s.t.

- $x \in L \Rightarrow \text{Prob}(M(x) = 1) \geq \frac{1}{p(|x|)}$
- $x \notin L \Rightarrow \text{Prob}(M(x) = 1) = 0$

Definition 15

$L \in RP_2$ iff $\exists$ probabilistic Poly-time TM M and a polynomial $p(.)$, s.t.

- $x \in L \Rightarrow \text{Prob}(M(x) = 1) \geq 1 - 2^{-p(|x|)}$
- $x \notin L \Rightarrow \text{Prob}(M(x) = 1) = 0$

Def. 14 has a high error prob. (i.e., $1 - \frac{1}{p(|x|)}$), while the error prob. under Def. 15 is small (i.e., $2^{-p(|x|)}$).

Robustness of RP

Theorem 16

$RP = RP_1 = RP_2$ and co-RP = co-RP_1 = co-RP_2

Proof.

$RP_2 \subseteq RP \subseteq RP_1$ follows from the definitions.
To show $RP_1 \subseteq RP_2$, given an x repeat M (for RP_1) $p(|x|)^2$ times and accept if at least one of the runs accepts. For $x \in L(M)$,
$Prob(M(x) = 0) \leq (1 - \frac{1}{p(|x|)})^{p(|x|)^2} = ((1 - \frac{1}{p(|x|)})^{p(|x|)})^{p(|x|)} \leq \frac{1}{e^{p(|x|)}}$
$\leq \frac{1}{2^{p(|x|)}}$. Hence, $Prob(M(x) = 1) \geq 1 - 2^{-p(|x|)}$. $\square$

Note: $(1 - \frac{1}{t})^t \leq \frac{1}{e}$.

Zero-Sided Error: The class ZPP

Let $\chi_L(x)$=1 if $x \in L$; = 0 if $x \notin L$.

Definition 17

$L \in ZPP$ (Zero-Error Polynomial Probabilistic Time) iff there exists a polynomial-time probabilistic TM M, such that $\forall x \in L$:
$M(x) = \{0, 1, \perp\}$,

- $\text{Prob}(M(x) = \perp) < \frac{1}{2}$, and
- $\text{Prob}(M(x) = \chi_L(x) \vee M(x) = \perp) = 1$

- $\text{Prob}(M(x) = \chi_L(x)) > \frac{1}{2}$
- The symbol $\perp$ is "I don't know".
- The value $\frac{1}{2}$ is arbitrary and can be replaced by $2^{-p(|x|)}$ or $1 - \frac{1}{p(|x|)}$.
- Also known as "Las-Vegas algorithm"

Alternative Definition of ZPP

Definition 18

ZPP is the class of languages accepted by a PTM with *polynomial expected running time* such that $\forall x \in \Sigma^*$,

- $x \in L \Rightarrow Prob[M(x) = 1] = 1$
- $x \notin L \Rightarrow Prob[M(x) = 0] = 1$

- Note that it is possible for the running time to be unbounded, we do not analyze the worst-case running time, but instead the average running time.
- If we "trim" the height of a computation when exceeding a certain polynomial, and mark those trimmed configurations as $\perp$, we get Def. 17.

ZPP = RP ∩ coRP

Theorem 19

$ZPP \subseteq RP \cap coRP$

Proof.

- Let $L \in ZPP$, M be the PTM that recognizes L.
- Define $M'(x) =$
  - let $b = M(x)$
  - $b = \perp$ then return 0, else return b
- If $x \notin L$, $M'(x)$ will never return 1.
- If $x \in L$, $\text{Prob}(M'(x) = 1) > \frac{1}{2}$, as required.
- ZPP $\subseteq$ RP
- The same way, ZPP $\subseteq$ coRP.

□

ZPP = RP ∩ coRP

Theorem 20

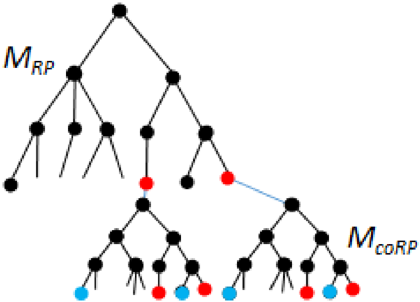
$RP \cap coRP \subseteq ZPP$

Proof.

- Let $L \in RP \cap coRP$, M_{RP} and M_{coRP} be the PTMs that recognize L according to RP and $coRP$.
- Define: $M'(x) =$
  - if $M_{RP} = YES$, return 1
  - if $M_{coRP} = NO$, then return 0, else return $\perp$
- $M_{RP}(x)$ never returns YES if $x \notin L$, and $M_{coRP}(x)$ never returns NO if $x \in L$. Therefore, $M'(x)$ never returns the opposite of $\chi_L(x)$.
- The probability that M_{RP} and M_{coRP} are both wrong $< \frac{1}{2} \Rightarrow$ $\text{Prob}(M'(x) = \perp) < \frac{1}{2}$.
- RP ∩ coRP ⊆ ZPP

□

ZPP = RP ∩ coRP



- In the above, **black**: accept; red: reject; blue: $\perp$.
- if $x \in RP$, $M_{RP}(x)$ has both black and red.
- if $x \in coRP$, $M_{coRP}(x)$ has all black. Black turns into blue.
- if $x \notin RP$, $M_{RP}(x)$ has all red.
- if $x \notin coRP$, $M_{coRP}(x)$ has both black and red. Black turns into blue.

Two-Sided Error: The class PP

Definition 21

$L \in PP$ (Polynomial Probabilistic Time) iff there exists a polynomial-time probabilistic TM M, such that $\forall x \in L$:

- if $x \in L$, $\text{Prob}(M(x) = 1) > \frac{1}{2}$, and
- if $x \notin L$, $\text{Prob}(M(x) = 1) \leq \frac{1}{2}$.

Two-Sided Error: The class BPP

Definition 22

$L \in BPP$ (Bounded-Error Polynomial Probabilistic Time) iff there exists a polynomial-time probabilistic TM M, such that $\forall x \in L$: $\mathrm{Prob}(M(x) = \chi_L(x)) \geq \frac{2}{3}$, where

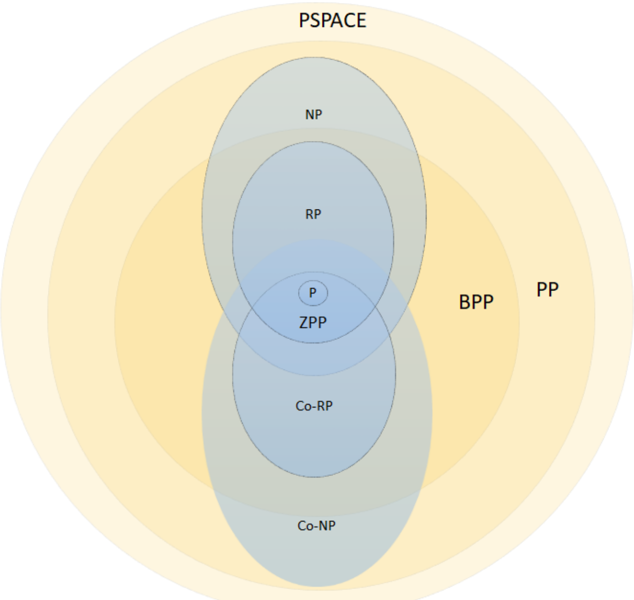
- $\chi_L(x) = 1$ if $x \in L$, and
- $\chi_L(x) = 0$ if $x \notin L$.

Theorem 23

If $L \in BPP$, then for every d, there exists a probabilistic polynomial TM M', s.t. $\forall x$, $Prob(M'(x) \neq \chi_L(x)) < 2^{-|x|^d}$

- Even a weak bound on the error is enough to obtain almost arbitrary certainty in polynomial time!
- *BPP* might be better than *P* for describing what is "tractable in practice"!

Relationship among Probabilistic Classes



Some Notes

- Probabilistic classes with one-sided error - RP and coRP - are common.
- ZPP defines random computations with zero-sided error, but probabilistic runtime.
- Many BPP algorithms have been de-randomised successfully
- Many experts believe that (Conjecture)

$$P = ZPP = RP = RP = BPP \subset PP$$

- BPP = P is equivalent to the existence of strong pseudo-random number generators, which many experts consider likely

What is a "Proof"?

- From the complexity viewpoint: meaningless unless can be efficiently verified.
- Given language L, our goal is to prove $x \in L$
- A Proof System for L is a verification algorithm V
  - (**completeness**): $x \in L \Rightarrow \exists\ \mathit{proof}$, V accepts (x, proof)

    "true assertions have proofs"
  - (**soundness**): $x \notin L \Rightarrow \forall\ \mathit{proof}^*$, V rejects (x, proof^*)

    "false assertions have no proofs"
  - (**efficiency**): $\forall x, \mathit{proof}$: $V(x, \mathit{proof})$ runs in polynomial time in $|x|$

Classical Proofs

- Recall the class *NP*:

  $L \in NP$ iff expressible as $L = \{x \mid \exists y, |y| \leq |x|^k, (x, y) \in R\}$, where k is a constant, and $R \in P$.

- *NP* is the set of languages with classical proof systems (R is the verifier, and y is the "proof")

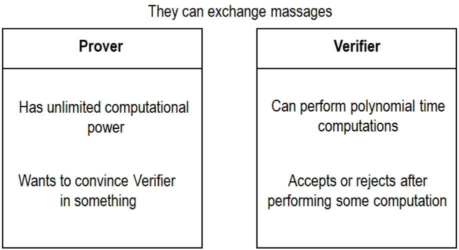
Definition 24

$L \subseteq \{0,1\}^*$ is in *NP* if $\exists$ a polynomial p and a ptime DTM M such that $\forall x \in \{0,1\}^*$

- (Completeness) $x \in L \Rightarrow \exists y \in \{0,1\}^{p(|x|)}, M(x, y) = 1$
- (Soundness) $x \notin L \Rightarrow \forall y \in \{0,1\}^{p(|x|)}, M(x, y) = 0$

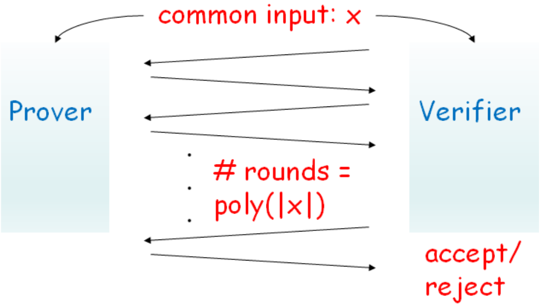
Interactive Proofs

- Two new ingredients:
  - Randomness: verifier uses randomness (e.g., tosses coins), allowing errors with some small probability
  - Interaction: rather than only "reading" a proof, verifier interacts with computationally unlimited prover
- Interaction and randomness possibly add power
  - *NP*: prover sends proof, verifier does not use randomness
  - *BPP*: randomness alone, no interaction



Interactive Proofs

- An interactive proof system for language L is an interactive protocol (P, V)
  - completeness: $x \in L \Rightarrow Pr[V \text{ accepts in } (P, V)(x)] \geq \frac{2}{3}$
  - soundness: $x \notin L \Rightarrow \forall P^*, \; Pr[V \text{ accepts in } (P^*, V)(x)] \leq \frac{1}{3}$
  - efficiency: V is p.p.t. machine

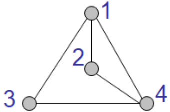


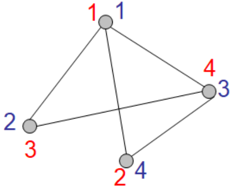
- $IP[k]$: languages that have k-round interactive proofs

Graph Isomorphism

Graphs $G_0 = (V, E_0)$ and $G_1 = (V, E_1)$ are isomorphic ($G_0 \approx G_1$) if exists a permutation $\Pi : V \to V$ for which

$$(x, y) \in E_0 \Leftrightarrow (\Pi(x), \Pi(y)) \in E_1$$





Graph (Non)isomorphism

- $GI = \{(G_0, G_1) : G0 \approx G1\}$ in NP not known to be in P, or NP-complete. Best algorithm takes $2^{O((\log n)^3)}$ time (2017).
- $GNI = \{(G_0, G_1) : G0 \not\approx G1\}$ not known to be in NP

Theorem 25

$GNI \in IP$.

indication *IP* may be more powerful than *NP*

$GNI \in IP$

input: (G_0, G_1)

Prover

Verifier

$H = \pi(G_c)$

pick $c \in \{0,1\}$ randomly and permute G_c to yield $H = \pi(G_c)$

if $H \cong G_0$ $r = 0$, else $r = 1$

r

accept iff $r = c$

$GNI \in IP$

- Completeness: If $G_0 \not\approx G_1$ then H is isomorphic to exactly one of (G_0, G_1) (honest) Prover will choose correct r. V accepts with prob=1.
- Soundness: If $G_0 \approx G_1$ then prover has no way of knowing whether H is the permutation of G_0 or G_1. Any prover P^* can "succeed" (by tricking verifier to accept wrongly) with probability at most $\frac{1}{2}$.
- Repeat the above twice can lower the error prob to $\frac{1}{4}$.

Interactive Proof for GI

- As GI is in NP, a simple IP is for P to send the isomorphism to V. The solution, however, is not zero knowledge.
- Consider the following solution. Note that if $G_0 \approx G_1$ Prover P can find two random permutations γ_0 and γ_1 such that $\gamma_0(G_0) = H = \gamma_1(G_1)$, for some H. Thus, letting $\sigma = \gamma_1^{-1}\gamma_0$, $\sigma(G_0) = G_1$. Also note that $\gamma_0\sigma^{-1}(G_1) = H$.

Repeat the following k times.

Prover P

(1) Let H be $\gamma_0(G_0)$; Send H to Verifier V.

(3)
- If $b = 0$, send $\gamma = \gamma_0$ to V;
- If $b = 1$, send $\gamma = \gamma_0\sigma^{-1}$ to V

Verifier V

(2) Choose $b \in \{0, 1\}$ randomly; Send b to Prover P

(4) Check $\gamma(G_b) = H$. If yes, accept; otherwise, reject.

- If $G_0 \approx G_1 \Rightarrow$ accept with prob. = 1
- If $G_0 \not\approx G_1 \Rightarrow$ prob. of catching a mistake = 1 - $(1/2)^k$.
- Zero knowledge

Interactive Proof for GI

- If $G_0 \approx G_1$, H can be obtained using either $\gamma_0(G_0)$ or $\gamma_1(G_1)$.
- If $G_0 \not\approx G_1$, H can only be obtained using $\gamma_0(G_0)$.
- Chosen randomly, the $b \in \{0, 1\}$ Verifier sends to Prover is to "challenge" Prover to send the correct permutation using which H can be obtained from G_b.
- If $G_0 \approx G_1$, Prover can always send the correct permutation (γ_0 if $b = 0$, or $\gamma_0\sigma^{-1}$ if $b = 1$) to Verifier.
- If $G_0 \not\approx G_1$, Prover can send the correct permutation (i.e., γ_0) only if $b = 0$, as that is what H is obtained originally. If $b = 1$, Verifier will reject as whatever γ Prover sends, Verifier will not be able to obtain H from G_1. As a result, with prob. = 1/2 Verifier will catch a mistake.
- By repeating the interaction k times, if $G_0 \not\approx G_1$ the prob. of catching a mistake will be $1 - (1/2)^k$.

The Power of *IP*

Theorem 26

IP = *PSPACE*.

Zero Knowledge Interactive Proofs

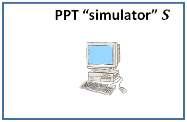
- A Zero Knowledge interactive proof system for language L is an interactive protocol (P, V)
  - Completeness: $x \in L \Rightarrow Pr[V \text{ accepts in } (P, V)(x)] \geq \frac{2}{3}$
  - Soundness: $x \notin L \Rightarrow \forall P^*, \ \ Pr[V \text{ accepts } in \ (P^*, V)(x)] \leq \frac{1}{3}$
  - Efficiency: V is p.p.t. machine
  - Zero Knowledge: no efficient V^* learns anything more than validity of $x \in L$?.

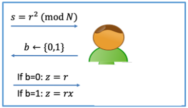
How to Define Zero Knowledge?

- After the interaction, V knows:
  - The theorem is true; and
  - A view of the interaction (= transcript + coins of V)
- P gives zero knowledge to V:
  - When the theorem is true, the view gives V nothing that he couldn't have obtained on his own without interacting with P.
- (P, V) is zero-knowledge if V can "simulate" (or "generate") his **VIEW** of the interaction <u>all by himself</u> in probabilistic ptime.

sim_S:
(s, b, z)

$view_V(P, V)$:
Transcript = (s, b, z),
Coins = b





Zero Knowledge Interactive Proof

Recall the Interactive proof for Graph Isomorphism.

- View of V= {(H, coin, random isomorphism of G_b to H }, i.e.,
  - $P \xrightarrow{H} V$
  - $V \xrightarrow{b} P$
  - $P \xrightarrow{\gamma} V$
- Simulator M: Toss coin
  - If coin=head, choose random γ_0 set $H = \gamma_0(G_0)$
  - If coin=tail, choose random γ_1 set $H = \gamma_1(G_1)$

Theorem 27

Every language in NP has a zero-knowledge interactive proof.