Advanced Proof Systems - Course Material

Yosef Israel Goren

November 16, 2022

Lecture 1

Recap

P - Polynomial (Class)

$L \in \{0,1\}^*$ is also in P if there exists an efficient algorithm which decides it.

NP - Nondeterministic Polynomial

$L \in \{0,1\}^*$ is also in NP if there exists an efficient algorithm V and a polynomial p which follow:

1. Completeness: $\forall x \in L, \exists y : V(x,y) = 1 \wedge |y| < p(|x|)$
2. Soundness: $\forall x \notin L \forall y : V(x,y) \neq 1 \vee |y| \geq p(|x|)$

PPT - Probobalistic Polynomial Time

This is a class of algorithms which must run in time polynomial to the size of their input, but also - must be capable of randomization, or 'flipping coins'.

IP - Interactive Proof

A key difference between an Interactive Proof and a proof for an NP proof is that the latter necessarily requires the prover to provide the verifier with something he can use to prove the truth of the calim to others.

We denote $(P,V)(x)$ to be the output of V (verifier) after the interaction between P and V on the input x. Both P and V can be thought of as PPT algorithms or programs which are capable of communicating with one another.

These interactions are often described with an interaction diagram:

- P sends to V something

- V sends to P something else
- ...
- V accepts iff ...

Formal Definition: We say that $L \in IP$ if there exists a polynomial algorithm V, an unbounded algorithm P and some constant $c \in (0.5, 1]$ s.t.

1. Completeness: $\forall x \in L, Pr[(P, V)(x) = 1] > c$
2. Soundness: if $x \notin L, \forall P^* \in \mathbb{M}, Pr[(P^*, V)(x) = 1] < 1 - c$

Note: $\mathbb{M}$ denotes the set of turing machines.

Equivalence of IP separation constants

Iterative Runs

Given an IP protocol (P, V), let (P^k, V^k) be the protocol obtained by running (P, V) k times sequentially. V^k accepts iff in all iterations V accepted.

Lemma

if (P, V) is IP with perfect completeness then for every polynomial k, (P^k, V^k) is IP with perfect completeness and soundness error $2^{-\Omega(k)}$

Proof:

1. V^k is efficient (composition of polynomials).
2. Perfect Completeness - due to Perfect Completeness of the original protocol - each iteration is guaranteed to succeed thus the protocol always does.
3. Soundness: Let $x \in L, P^*$. We will show that $Pr[(P^*, V^k)(x) = 1] \leq 2^{-k}$. Denote by E_i the event that V^k accepts in the i'th iteration. Thus:

$$Pr[E_1 \wedge E_2 \wedge \dots E_k] = \prod_{i=1}^{k} Pr[E_i | E_1 \wedge E_2 \wedge \dots E_k]$$

Claim: $Pr[E_i | E_1 \wedge E_2 \wedge \dots E_k] \leq 0.5$.
Proof: Assume toward a contradiction that $Pr[E_i | E_1 \wedge E_2 \wedge \dots E_k] > 0.5$
We design a prover P^{**} that convinces V with up to > 0.5:
P^{**} emulates (P^*, V) for iterations $1...i-1$ until the event $E_1 \wedge E_2 \wedge \dots E_k$ happens and then runs (P^*, V) as the i'th iteration. Since the run of (P^*, V) for the i'th iteration only happens under the condition $Pr[E_i | E_1 \wedge E_2 \wedge \dots E_k]$ - the probability for (P^*, V) to happen on the i'th iteration is exactly $Pr[E_i | E_1 \wedge E_2 \wedge \dots E_k] > 0.5$ but this is also the

probability for $Pr[(P^{**}, V)(x) = 1]$. Contradiction.

Thus:

$$Pr[E_1 \wedge E_2 \wedge \ldots E_k] = \prod_{i=1}^{k} Pr[E_i | E_1 \wedge E_2 \wedge \ldots E_k] \leq \prod_{i=1}^{k} 0.5 = 2^{-k}$$

Graph Isomorphism & IP Example

Graph Isomorphism - Definition

The graphs $G_1 = (V, E_1), G_2 = (V, E_2)$ are isomorphic or $(G_1, G_2) \in GI$ if $\exists \pi : V \longrightarrow V$ s.t. $(u, v) \in E_1 \iff (\pi(u), \pi(v)) \in E_2$.
More simply - two graphs are isomorphic if they are identical up to a renaming of their vertices.

GNI is the set of pairs of graphs which are isomorphic.

- Claim (no proof): $NP \subseteq IP$.

- Claim (proof sketch): $GI \in IP$.
  By finding the permutation π it is easy to check the GI condition over a given (G_1, G_2) thus we have an NP relation, meaning $GI \in NP$.

- Claim (no proof): $GNI \in IP$.

Tutorial 1

IP and NP

Claim: $NP \subseteq IP$.
Proof: let $L \in NP$. There exists some NP relation R for L, with an efficient algorithm M_R which decides it.
Now define an IP protocol:

- Both P and V get x.
- If $x \in L$ P find y s.t. $(x, y) \in R$, and send it to V. Otherwise send ϵ.
- V checks if $(x, y) \in R$ by running M_R (known to be efficient) and accepts iff $M(x, y)$ accepts.

1. Completeness: If $x \in L$, such y must exist (NP definition) thus P will find it, and V will have $(x, y) \in R$ so M_R and V will accept.
2. Soundness: If $x \notin L$, there is no y which such that $(x, y) \in R$ so no matter what any P^* sends - $M_R(x, y)$ rejects and so V rejects too.

Similar Proof Systems

Arthur-Merlin - AM

- Both parties get some input x.
- Arthur sends Merlin some randomized α.
- Merlin sends back some β.
- Arthur accepets according to some PPT algorithm which is a function of x, α and β (usually denoted $A(x, \alpha, \beta)$).

Merlin-Arthur - MA

- Both parties get input x.
- Merlin sends β to Arthur.
- Arthur generates some random value α.
- Arthur accepts according to some PPT algorithm which is a function of x, α, β.

Theorem: $MA \subseteq AM$.
Proof: Let $L \in MA$. WLOG (without loss of generality), L has a MA protocol with perfect completeness (we will come back to this assumption later in the course).
Denote by $p(n)$ the length β ($|\beta| \leq p(n)$).

Using repetition we can get to any protocol with perfect completeness and a soundness error of $2^{-p(n)-1}$ (as seen in lecture).
Sketching the repeated protocol would look like:

- M' sends β to A'
- A' sends back a list of α values (as many as there are repetitions).
- A' decides wether to accept.

This is because there is not reason for the prover's proof (β) to change due to different sampling of α, so it is always the same and can be sent once. So the length of Merlin's message does not change in the repeating protocol.

Now consider the same M', A' protocol but where A' sends the aggregated α before M' sends β.

Claim: This new protocol is AM. Proof:

1. Completeness: $\forall x \in L, M'$ sends the same β without looking at α:

$$Pr[(M', A)(x) = 1] = Pr[A(x, \alpha, \beta) = 1] = 1$$

2. Soundness: Let $x \notin L$, fix M^*. Consider:

$$Pr[(M^*, A')(x) = 1] = Pr[\exists \beta \in \{0,1\}^p : A'(x, \alpha, \beta) = 1]$$

$$= \cup_{\beta \in \{0,1\}^p} Pr[A'(x, \alpha, \beta) = 1] \leq_{UB} 2^p 2^{-(p+1)} = \frac{1}{2}$$

Note: UB denotes Union Bound.

Lecture 2

Recap

Remarks on GI

- For any graph G, the set of all (G, G'), where G' is isomorphic to G is an equivalence relation.
- For any graph G: $(\pi(G))_{\pi \leftarrow \mathbb{U}} \approx \mathbb{I}\sim\rtimes(\mathbb{G})$ - meaning uniformly sampling a permutation (over the set of nodes) of G and applying it to G will yield a uniformly sampled graph from the same equivalence partition.

$GNI \subseteq IP$

To prove this we will present an interactive protocol that runs on input G_0, G_1:

- V samples $b \leftarrow \{0,1\}, \pi \leftarrow \{\texttt{permutations}\}$.
- V sends $G = \pi(G_b)$.
- P checks: if $G \approx G_0$, $b' = 0$ else $b' = 1$.
- P sends b' to V.
- V accepts iff $b = b'$.

Now we prove that this interactive protocol is indeed an interactive proof for GNI:

1. Soundness: Let $G_0 \approx G_1$ and let P^* be a (possibly) cheating prover.
If $b = 0$ then G is uniformly distributed over all graphs isomorphic to G_0.
If $b = 1$ then G is uniformly distributed over all graphs isomorphic to G_1, but due to transitivity, this is the same as the prior case, meaning in any case, G is distributed uniformly over the equivalence to G_0.
Since this is the case, no information has been passed from V to P^* by sending G (P^* could have just sampled a graph from $Iso(G_0)$ itself!), P^* could not do more than guess b, and it's probability for success is no more than $\frac{1}{2}$.

2. Completeness: If the graphs are not isomorphic, $G \approx G_0$ iff $b = 0$, which will mean $b = b'$ is always the case and V always accepts.

3. Runtime: V does nothing of high complexity, P is unbounded.

Equivalence between IP and $AM[Poly]$

Lemma: Set Lower Bound Protocol

Theorem

- $IP[k]$: k-round interactive proof.
- $AM[k]$: k-round public coin interactive proof.

Claim (no proof): $AM[k] \subseteq IP[k]$.
Theorem: $IP[k] \subseteq AM[k + 2]$.
Corollary: $IP = AM[Poly]$.
Theorem Proof:
Let L be a language with an IP which satisfies the following assumptions:

1. $k = 2$.
2. Perfect Completeness.
3. There is a set of N of possible verifier messages which is constant for any specific protocol input x.
4. The message m sent by V is uniformly distributed over the N possible messages that could be sent.
5. The soundness error is $\frac{1}{100}$.

We want to show that L is in $AM[k + 2]$.
First, note how our IP can be described as follows when running on input x:

- V samples some $r \leftarrow \{0, 1\}$.
- V calculates some $m = V_1(x, r)$ for some efficient function V_1.

- V sends m to P.
- P sends back some π.
- V accepts iff $V_2(x, \pi, r) = 1$ for some efficient function V_2.

Define:

$$R_m = \{r' \mid m = V_1(x, r')\}, S_m = \{r' \in R_m \mid V_2(x, \pi, r') = 1\}$$

Now consider an alternative protocol, which is the same as the one above, only V also samples $r' \leftarrow R_m$ and accepts iff $V_2(x, \pi, r')$.

Note how $|R_m| = \frac{2^l}{N}$, and if $x \in L$ then $S_m = R_m$. Claim (missing proof): if $x \in L$ then $\mathbb{E}(|S_m|) = \frac{2^l}{N}$ and otherwise $\mathbb{E}(|S_m|) \leq \frac{1}{100}\frac{2^l}{N}$.

Pairwise Independent Hash Functions

Definition: A set of hash functions $\{h : X \leftarrow Y\}$ is pairwise independent if for all $x_1, x_2 \in X$ where $x_1 \neq x_2$ and $y_1, y_2 \in Y$, they satisfy:

$$Pr_{h \leftarrow H}[h(x_1) = y_1 \wedge h(x_2) = y_2] = \frac{1}{|Y|^2}$$

Construction:
Let $\mathbb{F}$ be a finite field. Given $a, b \in \mathbb{F} : h_{a,b}(x) = ax + b$.
Let $H = \{h_{a,b} \mid a, b \in \mathbb{F}\}$.
Construction proof:
Let x_1, x_2, y_1, y_2 as described in definition.

$$Pr_{a,b}[ax_1 + b = y_1 \wedge ax_2 + b = y_2] = Pr_{a,b}[\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}]$$

$$= Pr_{a,b}[\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}] = \frac{1}{|\mathbb{F}|^2}$$

For the last transition: Note how the right value of the equality (in the probability) is just a constant value, so we just need to randomly sample a, b to be some spesific values within a finite field.

Tutorial 2

Perfect Completeness?

We have seen in the lecture how an IP can be reduced to an AM proof, or in other words; a public coin IP. Here, we would like to show how any AM can be reduced to an AM with perfect completeness (which is also an IP with perfect

completeness).
This means that $\forall L \in IP$, it must have a public coin, perfectly complete interactive proof.
To construct the reduction, we start with the following z-round public coin protocol ($AM[z]$), which runs on input X:

- A samples $\alpha \leftarrow \{0,1\}^{rc}$, and sends it to M.
- M calculates $\beta = M(X, \alpha)$ and sends it to m.
- A accepts iff $A(X, \alpha, \beta) = 1$.

Where we assume completeness error $\epsilon > 0$:

$$\forall x \in L : Pr[(M, A)(X) = 1] \geq 1 - \epsilon$$

$$\Rightarrow \forall x \in L : Pr[\exists \beta : A(X, \alpha, \beta) = 1] \geq 1 - \epsilon$$

Now, we want to use it to construct an equivalent with perfect completeness; for that end, consider the alternative protocol:

- M' samples $s_1, s_2, ..., s_k \leftarrow \{0,1\}^{rc}$. s.t. (*)

  $$\forall \alpha \{0,1\}^{rc}, \exists i \in [k] : s_i \oplus \alpha \notin REJ$$

  .
- M' sends $s_1, s_2, ..., s_k$ to A'.
- A' samples $\alpha \leftarrow \{0,1\}^{rc}$ and sends it to M'.
- M' calculates $\forall i : \beta_i = M(X, s_i \oplus \alpha)$ and sends it.
- A' accepts iff $(\exists i : A(X, s_i \oplus \alpha, \beta_i)) = 1$

Lemma 1: if $x \in L$ then pre-processing succeeds.
We denote $\bar{s} = (s_1, s_2, ..., s_k)$. We say that $\bar{s}$ is 'good' if it satisfies (*).
To prove the lemma, we can show that:

$$\exists \bar{s} : \bar{s} \texttt{ is good}$$

We will show this by first showing:

$$Pr[\bar{s} \texttt{ is good }] > 0$$

To start off:

$$Pr_{\bar{s}}[\bar{s} \texttt{ is not good }] = Pr_{\bar{s}}[\exists \alpha, \forall i : s_i \oplus \alpha \in REJ]$$

$$= Pr_{\bar{s}}[\bigcup_{\alpha} (\forall i : s_i \oplus \alpha \in REJ)] \leq 2^{rc} \cdot \epsilon^k$$

From here we can see that for k of at-least rc, the probability is less then 1, meaning that the complementory probability is more than 0, meaning :

$$\exists \bar{s} : \bar{s} \texttt{ is good}$$

Completeness is trivial given Lemma 1.
Soundness of the protocol can be found in notes of Tutorial 2 in the website.

Lecture 3

Public Coin Protocol For Set Size

Lemma (Set LB)

Exists a protocol (P, V) such that: Given membership access to $s \subseteq U, t \subseteq \mathbb{N}$:

1. Completeness:
$$|s| \geq t \Rightarrow Pr[V \texttt{ accepts}] \geq \frac{2}{3}$$
2. Soundness:
$$\forall P^*, |s| \leq \frac{t}{100} \Rightarrow Pr[V \texttt{ accepts}] \leq \frac{1}{3}$$

The Protocol

1. V samples $h \leftarrow H$ where $H = \{h : U \rightarrow [t]\}$
2. P finds $x \in S$ s.t. $h(x) = 17$ and sends it to V (if none exist, P fails).
3. accepts iff $x \in S$ and $h(x) = 17$

Proof

For $x \in U, h \in H$, denote E_x if $h(x) = 17$.
Soundness: $|S| \leq \frac{t}{100}$.

$$Pr_{h\leftarrow H}[\exists x \in S, h(x) = 17] = Pr_{h\leftarrow H}[\bigcup_{x\in S} E_x] \leq \sum_{x\in S} Pr[E_x] = \frac{|S}{t} \leq \frac{1}{100}$$

Completeness: $|S| = t$.

$$Pr_{h\leftarrow H}[\exists x \in S, h(x) = 17] = Pr_{h\leftarrow H}[\bigcup_{x\in S} E_x] \geq \sum_{x\in S} Pr[E_x] - Pr_{x,x'\in S, x<x'}[E_x \cap E_{x'}]$$

$$= \frac{|S|}{t} - \binom{|S|}{2}\frac{1}{t^2} \geq \frac{|s|}{t} - \frac{|s|^2}{2t^2} = \frac{1}{2}$$

Zero Knowlage Proofs (ZKP)

ZKP protocol for GI

Given input G_0, G_1:

- P finds a permutation $\psi \in S_n$ s.t. $G_0 = \psi(G_1)$.
- P uniformly samples a random permutation $\pi \leftarrow S_n$.
- P sends $G = \pi(G_0)$ to V.
- V samples $b \leftarrow \{0, 1\}$ and sends it to P.
- P defines $\sigma = \pi$ if $b = 0$ else $\sigma = \pi \circ \psi$ and sends it to V.
- V accepts iff $\sigma(G_b) = G$.

$HV - ZKP$ definition

An interactive proof (P, V) is a honest-verifier zero knolage proof of $L \subseteq \{0, 1\}^*$ if there exists a PPT algorithm S called 'the simulator' for which: $\forall x \in L$: The following distributions are 'similar':

- $View^{x,p}(x) := (\texttt{the input } x, \texttt{ randomized coins } r, \texttt{ communications transcript})$
- $S(x)$ `(the output of the simulator)`

Here 'similar' can mean two different things:

- If it means identiacal, meaning these two distributions are the same distribution, we call the protocol a **perfect** ZKP.
- If it means that the distributions are computationally indistinguishable, we call the protocol a **statistical** ZKP, the definition for 'computationally indistinguishable' is in the next tutorial.

The idea here is that since the verifier V could have seen everything which is seen during the interactive protocol (P, V) by running the simulator S, the verifier did not actually learn anything from it.

The reason this is 'honset-verifier' is because this definition does not catch the case where the verifier 'cheats' and does not run the protocol V, and might be able to extract information P by doing it.

ZKP definition

An interactive proof (P, V) is a zero knowlage proof for L if:

$$\forall x, V^*, \exists S : View^{P,V^*}(x) \approx S(x)$$

Like to 'similar' from the honest verifier definition, '$\approx$' can mean either 'identical' or 'computational indistinguishable'; with the former being a perfect ZKP

and the latter being a statistical ZKP. We sometimes denote computational indistinguishability by $X \approx_C Y$ and perfect indistinguishability with $X \approx_P Y$.

In words, the difference from the honest-verifier zero ZKP is that now the simulator should be able to simulate anything that ANY verifier protocol manages to extract, meaning that regardless of the verifier protocol - no information is extracted.

$HZ - ZKP$ proof for GI protocol above

$$View := (G_0, G_1, G, b, \sigma), G = \sigma(G_b)$$

Define the following protocol (simulator); on input G_0, G_1 do:

- Sample $\sigma \leftarrow S_n$
- Sample $b \leftarrow \{0, 1\}$
- Define $G = \sigma(G_b)$
- return (G_0, G_1, G, b, σ)

Now consider each cell in the output tuple of the two distributions ($View$ and $S(G_0, G_1)$), G_0, G_1 are constant and always the same. $b\sigma$ are sampled uniformly from their ranges in both cases. So far there is no corralation between the cells. G is defined exactly given the rest of the variables meaning that since the rest of the variables are the same for both distributions G is also the same. Furthermore, since the correlations were identical so far, the new correlations are too.

These proofs often (such as in this case) follow the pattern of first showing that given a subset of the cells of the distribution tuples are the same due to being uniformly sampled respectively ((G_0, G_1, b, σ) in this case), then showing how the rest of the cells (G in this case) are a deterministic function of the other cells, and so the addition of these new cells does not impare with the distributions being the same.
Formally, if we have distributions D_1, D_2 and a (deterministic) function f:

$$D_1 \approx D_2 \Rightarrow (D_1, f_{d \leftarrow D_1}(d)) \approx (D_2, f_{d \leftarrow D_2}(d))$$

$NP \subseteq ZKP$ (proof concept)

In the lecture, proof by picture is shown for $3COL \in NPC$ has a zero knowlage proof. The main course of the proof consists of the prover commiting to a spesific permutation of the colors on a graph (without showing the verifier what these are), and allowing the verifier to choose a sepsific edge and see that indeed the coloring on both sides of that edge differ. For each iteration a cheating prover has some chance of failing due to the verifier haveing a chance to guess an edge where the two sides have the same color. Furthermore, the protocol is zero knowlage since the actual colors seen in each iteration are just two random colors.

Tutorial 3

Computational & Statistical Zero Knowledge

Distinguisher and Advantage - Definition

A distinguisher D is a probabilistic polynomial time algorithm; it receives an input w and tries to decide if $w \in X$ or $w \in Y$.
The advantage of D over X, Y is defined as:

$$adv_D(X,Y) := |Pr_{w \leftarrow X}[D(w) = 1] - Pr_{w \leftarrow Y}[D(w) = 1]|$$

Negligable Functions

A negligable function is a function $\epsilon : \mathbb{N} \to \mathbb{R}$ s.t.

$$\forall c \in \mathbb{R}, \epsilon(n) = o(\frac{1}{n^c})$$

This is equivalent to saying that:

$$\forall \texttt{polynomial } p(n), \epsilon(n) \leq \frac{1}{p(n)}$$

Computational Indistinguishability

Let X, Y be two ensambles of distributions, meaning that each of them consists of a seiries of distributions:

$$X = \{X_1, X_2, ...\}, Y = \{Y_1, Y_2, ...\}$$

We say that X is computationally indistinguishable from Y if for every distinguisher D there exists a negligable function ϵ such that:

$$\forall n \in \mathbb{N}, adv_D(X_n, Y_n) \leq \epsilon(n)$$

ZKP for $3COL$ (more formally)

Given an input graph $G = ([n], E)$:

- P finds a 3-coloring ϕ of G.
- P samples a permutation ξ over [3] (the colors).

- $\forall v \in [n]$, 'put $\xi(v)$ in a box' β_v and send it to V.
- V samples $(u, v) = e \leftarrow E$ and sends it to P.
- P sends the keys for β_v and β_u to V.
- V accepts iff colors 'inside' β_v, β_u are different.

To formalize the idea of these 'boxes' we define the notion of a commitment scheme:
A **commitment scheme** is a pair of PPT algorithms: *commit*, *check* with the following syntax:

- $\mathit{commit}(b; r) \to c$
- $\mathit{check}(c, b, r) \to \{0, 1\}$

And which satisfy the following conditions:

1. **Computationally Hiding**: $\mathit{commit}(0) \approx_C \mathit{commit}(1)$.
2. **Perfectly Binding**: There is no n_0, r_0, r_1 and C^* s.t.

$$\mathit{check}(C^*, 0, r_0) = \mathit{check}(C^*, 1, r_1) = 1 \texttt{ (1 means accept)}$$

Lecture 4

$coNP \subseteq IP$

Arithmetization

We interested in a reduction from a $coNP$ problem to an arithmetic problem.

<u>The Sumcheck Problem:</u>
Parameters: a finite field $\mathbb{F}$, and $n, d \in \mathbb{N}$.
Input: $Q : \mathbb{F}^n \leftarrow \mathbb{F}$, $\alpha \in \mathbb{F}$. Problem: does $\sum_{x \in \{0,1\}^n} Q(x) = \alpha$?

<u>The reduction</u> We will be reducing from $coNP$ to the sumcheck problem by reducing $3 - CNF$ to it (since $3 - CNF$ is $coNP$-complete).
Let $\phi \in 3 - CNF$ with n variables and m clauses.
We will start the construction by translating each building block of $3 - CNF$ formulas and expressing it in terms of polynomials:

1. $\phi(x_1, x_2, ..., x_n) = x_1 \longrightarrow p(x_1, x_2, ..., x_n) = x_1$
2. $x_1 \wedge x_2 \longrightarrow x_1 \cdot x_2$
3. $x_1 \wedge \neg x_3 \longrightarrow x_1 \cdot (1 - x_3)$

4. $(x_1) \vee (x_2) = \neg((\neg x_1) \wedge (\neg x_2)) \longrightarrow 1 - (1 - x_1) \cdot (1 - x_2)$

Lemma: for every $3 - CNF$ formula ϕ on m clauses and a finite field $\mathbb{F}$, there exists a ploynomial $p : \mathbb{F}^n \to \mathbb{F}$ with degree $O(m)$ s.t. $\phi = p$.
Furthermore, given ϕ and $z \in \mathbb{F}^n$, $p(z)$ can be evaluated in $poly(n, m, log(|\mathbb{F}|))$ time.

Tutorial 4