Agenda

- Definitions
- Chapter 2 of Lane and Ogi’s The Complexity Theory Companion
  - Theorem 2.5, has 2 claims
- Quiz in last 15 min of the class.

*Hint Hint: Try to understand the definitions*
One Way Function (OWF)?

- Function that is easy to compute but **HARD TO INVERT**
- No known One Way Function, (OWF), yet to be found!!
- But there are candidates!!

Requirements to be a One Way Function, \( f \):

- \( f \) can be computed in Polynomial Time (PTIME)
- \( f \) can’t be inverted in PTIME
- \( f \) is honest
What is Honesty?💡

Definition 2.1: Honesty

A function \( f \) is honest if the following holds:

\[
( \exists \text{ polynomial } q) \\
( \forall y \in \text{range}(f) ) \\
( \exists x) \\
[ |x| \leq q(|y|) \land f(x) = y ]
\]

\( f \) can shrink its input by no more than polynomial.
Why Honesty

Why Honesty?

● **f being honest means:** For each range element \( y = f(x) \), there is an \( x \) that is at most polynomially longer than \( y \) (i.e., for which \( f(x) \) is not more than “polynomially” shorter than \( x \))

● Intuitively: function cannot drastically “shrink” its input

● Better reflects intuitive notion of non invertibility - no “length tricks”

\[
f(x) = 1[\log \log \log (\max\{|x|,4\})]
\]

* simple example on board: \( f(x) = \log |x| \)
Polynomial Invertibility

**Definition 2.2:**

A function (possibly non total) \( f \) is PTIME invertible if there is a possible (possibly non total) PTIME computable function \( g \) such that:

\[
( \forall y \in \text{range } (f)) \quad [ y \in \text{domain}(g) \land g(y) \in \text{domain}(f) \land f(g(y)) = y ]
\]

simply means \( f \) can be reversed engineered in somewhat similar amount of time.
Definition 2.4: One to One

A function \( f \) is one to one if:

\[
( \forall y \in \Sigma^* ) \left[ \| \{ x \mid f(x) = y \} \| \leq 1 \right]
\]

Simple High School algebra:

\[
( \forall x_1, x_2 \in \Sigma^* ) \left[ f(x_1) = f(x_2) \Rightarrow (x_1 = x_2) \right]
\]
Now what?

Now we know what One Way Function (OWF) is!!!

A function $f$ is one way if:

- $f$ is polynomial-time computable,
- $f$ is not polynomial time invertible, and
- $f$ is honest
Theorem 2.5

1. One-way functions exists if and only if $P \neq NP$
2. One-to-one one way functions exist if and only if $P \neq UP$

We shall prove this theorem for the rest of the class session.

Note: Proof of 2 is simple modification of part 1. **SO PAY CLOSE ATTENTION** while we prove 1
Things to prove:

1. One-way functions exists if and only if $P \neq NP$
   a. If: $P \neq NP \implies$ One Way Function Exists
   b. Only if: One Way Function Exists $\implies P \neq NP$

2. One-to-one one way functions exist if and only if $P \neq UP$
   a. If: $P \neq UP \implies$ one-to-one one way functions exist
   b. Only if: One-to-one one way functions exist $\implies P \neq UP$
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists

- Assume $P \neq NP$
- Let $A \in NP - P$
- $\exists$ a NPTM, $N$ such that $L(N) = A$

Goal: Find a one way function $f$
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists ... 

**Quest for the magical $f$:**

- Assume some standard nice pairing function $<\cdot, \cdot>$:
  - PTIME computable and invertible
  - a bijection between $\Sigma^* \times \Sigma^*$ and $\Sigma^*$
- Consider a function $f$:

$$f(<x, w>) = \begin{cases} 0x, & \text{if } w \text{ is an accepting path for } N(x) \\ 1x, & \text{otherwise} \end{cases}$$

Remember:

$L(N) = A$ and $A \in NP - P$
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists ...

$$f(<x,w>) = \begin{cases} 
0x, & \text{if } w \text{ is an accepting path for } N(x) \\
1x, & \text{otherwise}
\end{cases}$$

**BIG PICTURE!**
- Will show $f$ is polynomial time computable, honest, hard to invert
- For the sake of contradiction assume $f$ is easily invertible in PTIME
- Establish $A \in P$
- Lead to a contradiction $A \in P \& A \in NP - P$
- $P = NP$

Remember: $L(N) = A$ and $A \in NP - P$
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists ...

**Quest for the magical $f$:**

- $f$ that takes paired values $<x, w>$ as input
- **Our Claim:**
  - $f$ is polynomial time computable
  - $f$ is honest

Remember:

$L(N) = A$ and

$A \in NP - P$
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists …

- $f$ is computable in PTIME?
  
  - checking whether $w$ is an accepting path by running $N$ on input $x$ is clearly polynomial

- $f$ is honest?
  
  - If $w$ is an accepting path, no path in $N$ can be longer than some polynomial $p(|x|)$. If $w$ is not an accepting path, $|w|$ might not be polynomial. But honesty only requires that some short preimage for $1x$ exist (easy to come up with one).

  Also, PTIME computability and invertibility of pairing function “prevents $f$ from destroying the honesty condition”
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists ...

- We are almost there!
- Showed that our $f$ is
  - computable in polynomial time ✓
  - honest ✓
  - hard to invert
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists ... 

**BIG PICTURE!**
- Showed $f$ is polynomial time computable, honest
- **Need to show hard to invert**
- For the sake of contradiction assume $f$ is “easily” invertible in PTIME
- Gives us $A \in P$
- Will construct a DPTM $M$, s.t $L(M) = A$
- Lead to a contradiction $A \in P$ & $A \in NP-P$
- $P = NP$

“easily” means in **polynomial time**
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists ...

- Goal: Show $f$ is hard to invert
- Assume $f$ is invertible via a polynomially computable function $g$
- $g$ allows us to accept $A$ in PTIME
- We get $A \in P$
- We will show that $A \in P$ by constructing a DPTM $M$ such that $L(M) = A$

Remember:
$L(N) = A$ and $A \in NP - P$
1(a) if: \( P \neq NP \) ⇒ One Way Function Exists …

- Construction \( M: \)

\[
f(<x, w>) = \begin{cases} 
0x, & \text{if } w \text{ is an accepting path for } N(x) \\
1x, & \text{otherwise}
\end{cases}
\]

Goal: Show \( f \) is hard to invert
1(a) if: \( P \neq NP \Rightarrow \) One Way Function Exists ...

- Construction \( M \):
- On input \( x \in \Sigma^* \), check if \( 0x \in \text{domain of } g \)
- if not, \text{REJECT!}

\[
f(<x,w>) = \begin{cases} 
0x, \text{if } w \text{ is an accepting path for } N(x) \\
1x, \text{otherwise}
\end{cases}
\]

Goal: Show \( f \) is hard to invert.
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists ...

- Construction $M$:
  - On input $x \in \Sigma^*$, check if $0x \in \text{domain of } g$
  - if not, REJECT!
  - if yes:
    - compute $g(0x)$, which returns $<x,w>$
    - test if $w$ is an accepting path in $N(x)$
      - if yes, ACCEPT!
      - Otherwise, REJECT!

\[
f(<x,w>) = \begin{cases} 
0x, \text{if } w \text{ is an accepting path for } N(x) \\
1x, \text{otherwise}
\end{cases}
\]

Goal: Show $f$ is hard to invert

Mr. RABBIT will use this construction in 2(a)
1(a) if: $P \neq NP \Rightarrow$ One Way Function Exists ...

- $M$ accepts $A$ in deterministic polynomial time
- Under our assumption $P \neq NP$
- Just showed a DPTM for $M$ and $L(M) = A$
- Showed $A \in P$
- Assumed $A \in NP - P$
- Contradiction!
What actually happened?

- We constructed DPTM $M$ assuming a inverse of $f$, $g$ existed, which is polynomial time computable.
- But $g$ does not exist i.e no polynomial time computable inverse of $f$ exists.
- $f^{-1}$ must not be polynomially computable, $f^{-1}$ is HARD to invert.
- $f$ must not be polynomially invertible.
- Vola!!! $f$ is now a One Way Function ✓
Where are we so far?

1. One-way functions exists if and only if $P \neq NP$
   
   a. if: $P \neq NP \Rightarrow \text{One Way Function Exists}$ ✓
   
   b. only if: One Way Function Exists $\Rightarrow P \neq NP$

2. One-to-one one way functions exist if and only if $P \neq UP$
   
   a. if : $P \neq UP \Rightarrow \text{one-to-one one way functions exist}$
   
   b. only if : One-to-one one way functions exist $\Rightarrow P \neq UP$
1(b) only if: One Way Function Exists $\Rightarrow P \neq NP$

- Reverse our previous strategy: given a one-way function $f$, assume $P = NP$, lead to contradiction.

- Let $p$ be $f$'s honesty polynomial

- Think of this language:
  \[
  L = \{ <z, \text{pre} > | (\exists y)[|y| + |\text{pre}| \leq p(|z|) \land f(\text{pre}.y) = z] \}
  \]

- What does this language “mean”?
  - Prefixes of the inverse of $z$, i.e. $f^{-1}(z)$, that are sufficiently short (for honesty)
1(b) only if: One Way Function Exists $\Rightarrow P \neq NP$

$$L = \{ < z, \text{pre} > \mid (\exists y)[|y| + |\text{pre}| \leq p(|z|) \land f(\text{pre}.y) = z] \}$$

- Clearly $L$ is NP: guess a string $y$, then check if $f(\text{pre}.y) = z$.
- Since we assumed $P = NP$, $L$ is also $P$!
- We’ll use this fact to invert $f$ “easily” (in P-time) - contradicting that $f$ is a one-way function.
1(b) only if: One Way Function Exists $\Rightarrow P \neq NP$

- Goal: given $z$, find its inverse with respect to $f$ in polynomial time
  - (find $x$ such that $f(x) = z$)
- Since (we assumed) $L \in P$, there is a DPTM accepting $L$.
- If $<z, \text{pre}>$ is in $L$, $\text{pre}$ is a prefix of $z$’s inverse
- We can check “easily” (P-time) whether something is in $L$!

What can we do with this?
1(b) only if: One Way Function Exists ⇒ P≠NP

Complexity rabbit says…

Search **ALL the prefixes!**
1(b) only if: One Way Function Exists $\Rightarrow P \neq NP$

Searching all the prefixes:

- Check if $f(\epsilon) = z$; if so we are done ($\epsilon$ is an inverse), if not go to next step.
- Is 0 a prefix of a suitably short inverse?
  - If NO, then 1 must be a prefix!
  - Either way, we determine the first bit.
- Are we done yet? (is this prefix the whole inverse?)
  - Check if $f(pre) = z$. If yes, we’re done! Otherwise, we need to find out the next bit…
- (Let $b$ be the bit we’ve already figured out.) Is $b0$ a prefix of a suitably short inverse?
  - If no, then $b1$ must be a prefix…
  - Now we have the second bit ($c$), check if $f(bc) = z$...
1(b) only if: One Way Function Exists $\Rightarrow P \neq NP$

Can think of this as a search as a tree:

- At each step, we discover one more bit of the inverse.
- We will make progress with each step: if the next bit isn’t 0, it must be 1. No exponential expansion!
- Hence prefix search is linear in the length of the inverse!
- The honesty polynomial bounds the length of the inverse.
Recap

- We started with a one way function \( f \)
- We supposed \( P = NP \)
- We examined a language \( L \) that lets us check if a string is a prefix of \( f^{-1}(z) \)
- Since \( P = NP \), \( L \in P \)
- We used \( L \) to search for the inverse in polynomial time
- Thus \( f \) can be inverted in polynomial time. Contradiction!
- Hence our assumption was wrong: \( P \neq NP \)
Where are we so far?

1. One-way function exists if and only if $P \neq NP$
   a. $if: P \neq NP \Rightarrow \text{One Way Function Exists} \checkmark$
   b. $only if: \text{One Way Function Exists} \Rightarrow P \neq NP \checkmark$

2. One-to-one one way function exists if and only if $P \neq UP$
   a. $if : P \neq UP \Rightarrow \text{one-to-one one way functions exist}$
   b. $only if : \text{One-to-one one way functions exist} \Rightarrow P \neq UP$
What is UP?

- A complexity class like \((NP, P)\) that has unique witness
- \(L \in UP\) if:
  - NP machine \(N\) accepts \(x \in L\)
  - For all such \(x\), the computation of \(N(x)\) has at most 1 accepting path

\[
UP = \{ L \mid \exists \text{ NPTM, } N \text{ such that } L = L(N), \text{ and } \forall x \in L, N(x) \text{ has at most 1 accepting path} \}
\]
2(a) if: $P \neq UP \Rightarrow$ one-to-one one way functions exist

- Let $A \in UP - P$
- $\exists$ NPTM, $N$ such that $A = L(N)$
- Consider function $f$:

$$f(<x,w>) = \begin{cases} 0x, & \text{if } w \text{ is an accepting path for } N(x) \\ 1 < x, w >, & \text{otherwise} \end{cases}$$

Goal: want to show $f$ is 1-to-1 one way function
Where are we now?

1. One-way function exists if and only if $P \neq NP$
   a. if: $P \neq NP \Rightarrow \text{One Way Function Exists}$ ✓
   b. only if: $\text{One Way Function Exists} \Rightarrow P \neq NP$ ✓

2. One-to-one one way function exists if and only if $P \neq UP$
   a. if : $P \neq UP \Rightarrow \text{one-to-one one way functions exist}$ ✓
   b. only if : $\text{One-to-one one way functions exist} \Rightarrow P \neq UP$
2(b): only if: 1-to-1 one way functions exist \( \Rightarrow P \neq UP \)

- No changes from 1(b)
  - Replace “one way function” with “1-to-1 one way function” and “NP” with “UP” - same argument holds
- Only difference: there is only one path in the prefix search tree that will lead us to an inverse.
  - We were ignoring the extras anyway (see special note)

- 1-to-1 one way functions exist \( \Rightarrow P \neq UP \)!!!
Big Picture!!

1. Got introduced to One Way Function (1-to-1 as well)!
2. Existence of One Way Function is tied to whether $P=NP$
3. For 1-to-1 One Way Function, it is tied to a more strongly regulated version of $NP$ i.e. $UP$ (***)
4. Next class we will expand on *** to cover a constant bounded version of UP