One way Functions
Chapter 2.2 Hem-ogi

Group B:
Yang Feng
Anis Zaman
Ethan Johnson
Amin Mosayyebzadeh
Reminder

- one-to-one one-way functions are characterized by $P \neq UP$
- one-way functions is characterized by $P \neq NP$

- $P \neq UP \Rightarrow P \neq NP$,
  - the converse has never been established
Note

- one-to-one function $f$ is completely unambiguous in terms of inversion
  - each element of range($f$) has exactly one inverse

- "constant-to-one" functions are called bounded-ambiguity functions. They are often referred as "O(1)-to-one" functions
Definition 2.6: Bounded-Ambiguity Functions

1. For each $k \geq 1$, we say that
   a. a (possibly non-total) function $f$ is $k$-to-one if $(\forall y \in \text{range}(f))[\|\{x \mid f(x) = y\}\| \leq k]$

   compare it with one-to-one function:
   $(\forall y \in \text{range}(f))[\|\{x \mid f(x) = y\}\| \leq 1]$

2. We say that
   a. a (possibly non-total) function $f$ is of bounded-ambiguity if there is a $k \geq 1$ such that $f$ is $k$-to-one.
Big picture

• We are going to prove that
  ○ Unambiguous one-way functions exist ⇔ Bounded-ambiguity one-way functions exist
Theorem 2.7

one-to-one one-way functions exist ⇔ constant-to-one one-way functions exist
Proof

- "Only if" direction is easy:
  - All one-to-one functions are constant-to-one functions, so the "only if" direction holds
- We will show the "if" direction
Definition 2.8

A language $L$ is in $\text{UP}_{\leq k}$, $k \geq 1$, if there is an NPTM $N$ such that

1. $(\forall x \in L)[N(x) \text{ has at least one and at most } k \text{ accepting paths}]$

and

2. $(\forall x \in \overline{L})[N(x) \text{ has no accepting paths}]$
Recall

• Part 2 of Theorem 2.5
  ○ one-to-one one-way functions exist \(\Leftrightarrow P \neq UP\)
Fact 2.8

- For each $k \geq 2$, $k$-to-one one-way functions exist $\iff P \neq UP_{\leq k}$

It can be proved by exactly analogous proof of Theorem 2.5
Proving the “if” direction

- We know that:
  - $P \neq UP \iff$ one-to-one one-way functions exist
  - $P \neq UP_{\leq k} \iff$ k-to-one one-way functions exist (from previous slide)

- If we show that $P \neq UP \iff P \neq UP_{\leq k}$, then
  - one-to-one one-way functions exist $\iff$ k-to-one one-way functions exist

- We will use induction that:
  - for all $k \in \{1, 2, 3, \ldots\}$, $P = UP \Rightarrow P = UP_{\leq k}$
Proving the “if” direction

- We know that:
  - $P \neq UP \iff$ one-to-one one-way functions exist
  - $P \neq UP_{\leq k} \iff$ k-to-one one-way functions exist (from previous slide)
- If we show that $P \neq UP \iff P \neq UP_{\leq k}$, then
  - one-to-one one-way functions exist $\iff$ k-to-one one-way functions exist
- We will use induction that:
  - for all $k \in \{1, 2, 3, \ldots\}$, $P = UP \Rightarrow P = UP_{\leq k}$
Induction

- holds for $k = 1$:
  - $P = UP \Rightarrow P = UP_{\leq 1}$
- Assume:
  - $P = UP \Rightarrow P = UP_{\leq k'}$
- prove:
  - $P = UP \Rightarrow P = UP_{\leq k'+1}$

$P \neq UP \iff P \neq UP_{\leq k}$
Proving $P = UP \Rightarrow P = UP_{\leq k'+1}$
(Assuming $P = UP \Rightarrow P = UP_{\leq k'}$)

- Assume $P = UP$
- Let $L$ be an arbitrary member of $UP_{\leq k'+1}$.
- Let $N$ be an NPTM —having at most $k' + 1$ accepting paths on each input— that accepts $L$ (recall Definition 2.8)
- Consider the set
  - $B = \{ x \mid N(x) \text{ has exactly } k' + 1 \text{ accepting paths} \}$
  - Clearly, $B \in UP$, via the machine that on each input $x$ guesses each lexicographically ordered $(k'+1)$-tuple of distinct computation paths and that accepts on such a path exactly if each of the $k' + 1$ guessed paths is an accepting path on input $x$.
  - by our $P = UP$ assumption, $B \in P$
K’+1 or K’, that is the question

So, we are excluding elements with k’+1 paths (Set B)

- since B ∈ P, the set
  - $D = \{x \mid x \notin B \land x \in L(N)\}$ is in $UP_{\leq k'}$

- $x$'s paths = K’+1
- $1 \leq x$'s paths $\leq k'$
Proving \( D \in \text{UP}^{\leq k'} \)

- We construct a TM \( M \) such that:
  - We first deterministically check whether \( x \) is in \( B \)
    - Using some P algorithm for \( B \).
    - Under our current assumptions, \( B \in P \). So some such algorithm exists.
  - If \( x \in B \) we reject
  - If \( x \notin B \) we directly simulate \( N(x) \).
    - This simulation will have at most \( k' \) accepting paths
      - \( x \notin B \) precludes there being exactly \( k'+1 \) paths
      - \( N \)'s choice precludes there being more than \( k' + 1 \) paths
- Since \( D \in \text{UP}^{\leq k'} \), we conclude
  - from our assumption that \( P = \text{UP} \),
  - from our inductive hypothesis (which was \( P= \text{UP} \Rightarrow P = \text{UP}^{\leq k'} \))

\( \Rightarrow D \in P. \)
Prove

- Since $P$ is closed under union, $B \cup D \in P$.
- However
  - $L = B \cup D \Rightarrow L \in P$
  - $L$ is an arbitrary member of $\text{UP} \leq k' + 1$

$\therefore P = \text{UP} \Rightarrow P = \text{UP} \leq k' + 1$
One Way Functions

Chapter 2, Lane-Ogi, Group B

Yang Feng
Anis Zaman
Ethan Johnson
Amin Mosayyebzadeh

Fall 2015, Midterm II
Two-argument (denoted 2-ary) one-way functions

\[ f(x, x') = y \]
Strong
Total
Commutative
Associative

⇔

One-Way Functions Exist

⇒⇒

One-Way Functions Exist
2-ary function honesty

Definition 2.10:

We say a 2-ary function $f: \Sigma^* \times \Sigma^* \to \Sigma^*$ is honest if

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

This definition only requires that each element of range($f$) have one appropriate pair $(x, x')$. 
2-ary function invertible

Definition 2.11:

We say a 2-ary function $f: \sum^* \times \sum^* \rightarrow \sum^*$ is polynomial-time invertible if there is a polynomial-time computable function $g$ such that, for each $y \in \text{range}(f)$,

$$y \in \text{domain}(g) \land (\text{first}(g(y)), \text{second}(g(y))) \in \text{domain}(f) \land f(\text{first}(g(y)), \text{second}(g(y))) = y,$$

where the projection functions first(z) and second(z) denote, respectively, the first and second components of the unique ordered pair of strings that when paired give z.
2-ary one-way function

We say a 2-ary function $f: \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ is one-way if

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

Are this familiar?
2-ary function s-honesty

We say a 2-ary function \( f: \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \) is s-honest if

1. \((\exists \text{ polynomial } q) \ (\forall y, a : (\exists b)[f(a, b) = y]) \)

\[(\exists b')[|b'| \leq q(|y| + |a|) \land f(a, b') = y].\]

2. \((\exists \text{ polynomial } q) \ (\forall y, b : (\exists a)[f(a, b) = y]) \)

\[(\exists a')[|a'| \leq q(|y| + |b|) \land f(a', b) = y].\]

How to understand it? See next page.
2-ary function invertible

\[ f(x_1, x_2) = y \]

\[ g(y, x_1) = \text{something similar to } x_2 \]
We say a 2-ary function \( f: \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \) is strongly noninvertible if it is s-honest and yet neither of the following conditions holds.

1. There is a (possibly nontotal) polynomial-time computable function \( g: \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \) such that \( (\forall y \in \text{range}(f))(\forall x_1, x_2 : (x_1, x_2) \in \text{domain}(f) \land f(x_1, x_2) = y)[(y, x_1) \in \text{domain}(g) \land f(x_1, g(y, x_1)) = y] \).

2. There is a (possibly nontotal) polynomial-time computable function \( g: \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \) such that \( (\forall y \in \text{range}(f))(\forall x_1, x_2 : (x_1, x_2) \in \text{domain}(f) \land f(x_1, x_2) = y)[(y, x_2) \in \text{domain}(g) \land f(g(y, x_2), x_2) = y] \).
2-ary function associative and commutative

We say a 2-ary function \( f: \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \) is associative if

\[
(\forall x, y, z)[f(f(x, y), z) = f(x, f(y, z))].
\]

We say a 2-ary function \( f: \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \) is commutative if

\[
(\forall x, y)[f(x, y) = f(y, x)].
\]

multiplication of integers? concatenation of strings?
Proposition 2.17

The following are equivalent.

1. One-way functions exist.
2. 2-ary one-way functions exist.
3. $P \neq NP$. 
\((2) \Rightarrow (1)\)

Let \(\langle \cdot, \cdot \rangle\) be a pairing function used before

Let \(f : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*\) be any 2-ary one-way function:

\[ g(z) = f(\text{first}(z), \text{second}(z)) \]

where, \(\text{first}(z)\) and \(\text{second}(z)\) denotes the first and second component of the pair mapped to \(z\).
Let $h : \Sigma^* \rightarrow \Sigma^*$ be a one-way function:

$$h'(x,y) = \langle h(x), h(y) \rangle$$