Computable one-way functions on the reals

Xiaoyan Zhang

State Key Lab of Computer Science
Institute of Software, Chinese Academy of Sciences

NUS Logic summer school
July 11, 2024

Joint work with George Barmpalias

Notations. Let $x, y, \cdots$ denote infinite binary sequences (reals). Let 2^ω be the set of all reals. Let $x(n)$ denote the n^{th} bit of x.

Notations. Let $x, y, \cdots$ denote infinite binary sequences (reals). Let 2^ω be the set of all reals. Let $x(n)$ denote the n^{th} bit of x.

We care about functions from 2^ω to 2^ω.

Notations. Let $x, y, \cdots$ denote infinite binary sequences (reals). Let 2^ω be the set of all reals. Let $x(n)$ denote the n^{th} bit of x.

We care about functions from 2^ω to 2^ω.

We regard an oracle Turing functional Φ as a partial function f from 2^ω to 2^ω, where

- $f(x)$ is defined if $\Phi^x(n)\downarrow$ for all n;
- the output of $f(x)$ is the y such that $y(n) = \Phi^x(n)$.

Notations. Let $x, y, \cdots$ denote infinite binary sequences (reals). Let 2^ω be the set of all reals. Let $x(n)$ denote the n^{th} bit of x.

We care about functions from 2^ω to 2^ω.

We regard an oracle Turing functional Φ as a partial function f from 2^ω to 2^ω, where

- $f(x)$ is defined if $\Phi^x(n)\downarrow$ for all n;
- the output of $f(x)$ is the y such that $y(n) = \Phi^x(n)$.

These are called **computable** functions.

One-way functions are the functions that are **easy to compute** but **hard to invert**.

Definition

Given partial f, g, and $y \in f(2^\omega)$, we say that g inverts f on y if

$$f(g(y)) = y.$$

We say that g is an inversion of f if g inverts y on all $y \in f(2^\omega)$.

Theorem (Folklore)

If f is total computable, then there is a partial computable g which inverts f on all y such that $|f^{-1}(y)| = 1$.
In particular, all total computable injections have partial computable inverse.

Let $\exists s, E(s)$ be a Σ^0_1 formula where $E(s)$ is computable and there is at most one s such that $E(s)$ holds. Then any g that inverts

$$f(x) = \begin{cases} 0^s x(0) 0^\omega & \text{if } E(s) \\ 0^\omega & \text{otherwise} \end{cases}$$

at $y = 0^\omega$ encodes the answer to $\exists s, E(s)$.

Theorem (Barmpalias, Z., 2024)

There exists a total computable f such that any inversion of f computes $\emptyset'$.
In particular, f does not have any computable inversion.

Theorem (Barmpalias, Z., 2024)

There exists a total computable f such that any inversion of f computes $\emptyset'$.
In particular, f does not have any computable inversion.

Construction.

Let $\langle \cdot, \cdot \rangle$ be a computable bijection such that $\langle n, s \rangle \geq s$. Let

$$f(x)(\langle n, s \rangle) = \begin{cases} x(n) & \text{if } n \in \emptyset'_{s+1} - \emptyset'_s \\ 0 & \text{otherwise.} \end{cases}$$

□

We give access to g a random oracle r to help inverting f. Then **the probability that** g **inverts** f **at** y is $\mu(\{r : f(g(y, r)) = y\})$.

We give access to g a random oracle r to help inverting f. Then **the probability that** g **inverts** f **at** y is $\mu(\{r : f(g(y, r)) = y\})$. More generally, we consider

$$L_{f,g} = \{(y, r) : f(g(y, r)) = y\}$$

and say that **the probability that** g **inverts** f is $\mu(L_{f,g})$.

We give access to g a random oracle r to help inverting f. Then **the probability that** g **inverts** f **at** y is $\mu(\{r : f(g(y, r)) = y\})$. More generally, we consider

$$L_{f,g} = \{(y, r) : f(g(y, r)) = y\}$$

and say that **the probability that** g **inverts** f is $\mu(L_{f,g})$.

Definition (Levin, 2023)

A partial computable f is **one-way** if $\mu(f(2^\omega)) > 0$ and for any partial computable g, $\mu(L_{f,g}) = 0$.

We give access to g a random oracle r to help inverting f. Then **the probability that** g **inverts** f **at** y is $\mu(\{r : f(g(y, r)) = y\})$. More generally, we consider

$$L_{f,g} = \{(y, r) : f(g(y, r)) = y\}$$

and say that **the probability that** g **inverts** f is $\mu(L_{f,g})$.

Definition (Levin, 2023)

A partial computable f is **one-way** if $\mu(f(2^\omega)) > 0$ and for any partial computable g, $\mu(L_{f,g}) = 0$.

Question (Levin, 2023)

Is there a random-preserving one-way function?

Theorem (Barmpalias, Z., 2024)

There is a total computable surjective random-preserving one-way function.

Theorem (Barmpalias, Z., 2024)

There is a total computable surjective random-preserving one-way function.

Construction.

Let $f(x)(\langle n,s\rangle) = \begin{cases} x(2n) & \text{if } n \in \emptyset'_{s+1} - \emptyset'_s \\ x(2\langle n,s\rangle + 1) & \text{otherwise.} \end{cases}$

Verify that

- f is total computable,
- f is surjective,
- f is random-preserving (f^{-1} is measure-preserving),
- f is one-way (Lebesgue's density theorem).

□

Remark

Any g such that $\mu(L_{f,g}) > 0$ computes $\emptyset'$.

Remark

Any g such that $\mu(L_{f,g}) > 0$ *computes* $\emptyset'$.

Remark

If either

- y *is weakly* 2*-random and* g *is partial computable*
- y *is weakly* 1*-random and* g *is total computable*

then the probability that g *inverts* f *on* y *is* 0.

Recall that

Theorem (Folklore)

All total computable injections have partial computable inverse.

Recall that

Theorem (Folklore)

All total computable injections have partial computable inverse.

Observation

*Among computable functions, being **total**, being **injective** and being **one-way** are incompatible.*

Recall that

Theorem (Folklore)

All total computable injections have partial computable inverse.

Observation

Among computable functions, being ***total****, being* ***injective*** *and being* ***one-way*** *are incompatible.*

What if we weaken or remove one of these conditions?

Recall that

Theorem (Folklore)

All total computable injections have partial computable inverse.

Observation

*Among computable functions, being **total**, being **injective** and being **one-way** are incompatible.*

What if we weaken or remove one of these conditions?

Observation

The random oracle r does not help invert functions.

From now on we drop the random oracle r.

Theorem (Barmpalias, Z., 2024)

If f is total computable and one-way, then $f^{-1}(y)$ has no isolated path (in particular, $|f^{-1}(y)| = 2^{\aleph_0}$) for almost all $y \in f(2^\omega)$.

Theorem (Barmpalias, Z., 2024)

If f is total computable and one-way, then $f^{-1}(y)$ has no isolated path (in particular, $|f^{-1}(y)| = 2^{\aleph_0}$) for almost all $y \in f(2^\omega)$.

Proof.

If $f^{-1}(y)$ has an isolated path x, then there is $\sigma \prec x$ that seperate x from all other paths.

Use σ to build a partial computable g that inverts f on y. Now

$$\begin{aligned} &\{y : f^{-1}(y) \text{ has an isolated path}\} \\ \subseteq &\{y : \text{some partial computable } g \text{ inverts } f \text{ at } y\} \end{aligned}$$

Finally note that there are only countably many partial computable functions, and each of them inverts f with 0 probability. □

f is two-to-one if $|f^{-1}(y)| \leq 2$ for all y.

Theorem (Barmpalias, Z., 2024)

There is a total computable two-to-one surjection, such that if there is g and σ such that g inverts f on all $y \succ \sigma$, then g computes $\emptyset'$.

f is two-to-one if $|f^{-1}(y)| \leq 2$ for all y.

Theorem (Barmpalias, Z., 2024)

There is a total computable two-to-one surjection, such that if there is g and σ such that g inverts f on all $y \succ \sigma$, then g computes $\emptyset'$.

Let $x \oplus y$ be the z such that $z(2n) = x(n)$ and $z(2n+1)$ is $y(n)$.

Proof Sketch.

Define $f(x \oplus z) = h^z(x) \oplus z$.
The function h picks, for each n, a particular index p_n^z and copy $x(p_n^z)$ to $h^z(x)(n)$. The oracle z can (partially) control this process.
We make sure at most one bit is not copied into $h^z(x)$.
We use z to control which question this bit is allocated to: let the event $z(\langle n, s\rangle) = 1$ indicate h to skip the question "$n \in \emptyset'$?". □

Theorem (Barmpalias, Z., 2024)

There is a total computable two-to-one surjection, such that if there is g such that g inverts f on almost all y, then g computes $\emptyset'$. In particular, any partial computable g cannot invert f with probability 1.

Theorem (Barmpalias, Z., 2024)

There is a total computable two-to-one surjection, such that if there is g such that g inverts f on almost all y, then g computes $\emptyset'$. In particular, any partial computable g cannot invert f with probability 1*.*

Proof Sketch (for the "in particular" case).

Similarly, except z indicates h to skip the question "$n \in \emptyset'$?" when some corresponding column of z appears to be non-random.
Fix $y \oplus w$ weakly 1-random, y random but no column of w is random, and that $y \oplus w$ is incomplete.
Build several z obtained by replacing a column of w by y.
These z are weakly 1-random, and $L_{f,g}$ is a Π^0_2 class, so if $\mu(L_{f,g}) = 1$ then all such $z \in L_{f,g}$. □

Summary for injectivity requirement

Assuming f is total computable, then

- without injectivity requirement, it could be that any partial computable g inverts f with probability 0;
- by requiring that $|f^{-1}(y)| < 2^{\aleph_0}$, there is always partial computable g that inverts f with positive probability;
- even when requiring that $|f^{-1}(y)| \leq 2$, it could be that no partial computable g inverts f with probability 1;
- by requiring that $|f^{-1}(y)| \leq 1$, a single partial computable g inverts f on all y.

Theorem (Barmpalias, Z., ongoing)

There is a partial computable random-preserving one-way injection.

Rather than considering $L_{f,g} = \{y : f(g(y)) = y\}$ and requiring that $\mu(f(2^\omega)) > 0$, we can instead consider

$$\{x : f(g(f(x))) = f(x)\}.$$

Definition (Gács, 2024)

A partial computable f is **semi-oneway** if $\mu(\{x : f(x)\downarrow\}) > 0$ and for any partial computable g, $\mu(\{x : f(g(f(x))) = f(x)\}) = 0$.

Theorem (Gács, 2024)

There is a partial computable semi-oneway function.

Theorem (Barmpalias, Z., 2024)

For total computable random-preserving f with $\mu(f(2^\omega)) > 0$*, f is one-way if and only if it is semi-oneway.*

Theorem (Barmpalias, Z., 2024)

For total computable random-preserving f with $\mu(f(2^\omega)) > 0$, f is one-way if and only if it is semi-oneway.

Proof.

For total computable f, let ν_f be the measure defined by

$$\nu_f([\![\sigma]\!]) = \mu(f^{-1}([\![\sigma]\!])).$$

- ν_f is a computable measure,
- x is random if and only if x is ν_f-random,
- ν_f and μ have the same null sets.

□

Thanks!