

Computable oneway real functions

Hardness of inverting functions on the reals

George Barmpalias

Chinese Academy of Sciences

Joint work with M. Wang and X. Zhang

Logic Colloquium, Vienna 2025

Effective functions on the reals

This is a standard notion in **computable analysis** due to Turing (1936).

A Turing machine computing $x \mapsto f(x)$

input	x	0	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1	...
output	$f(x)$	0	1	0	1	1	0											...

Computability of real functions is **effective continuity**:

- computable real functions are continuous
- every continuous real function is computable in some oracle

Time-complexity was defined on this basis (Ko and Friedman 1980s).

Overview

1. Inversions of real functions
2. Oneway real functions
3. Collisions and hashing
4. Weakly oneway
5. Nearly injective
6. Unresolved problems

Partial computable real functions

- Continuous with Π_2^0 domain and Σ_1^1 range
- if total then **uniformly** continuous with Π_1^0 range

Properties of interest include many-to-one versus one-to-one and

- **Positive:** they map a positive set to a positive set of reals
- **Random-preserving:** they map randoms to randoms

Proposition. A partial computable function is positive iff it extends a random-preserving partial computable function.

Positive and random-preserving maps can be **treated as one**.

Complexity of inversion

There is no degree bound, even for total functions.

Theorem

There is a total poly-time computable random-preserving surjection f which has no continuous inversion.

Total computable **injections** have computable inverses.

Proposition. If a total computable f is injective on R then $f : R \rightarrow 2^\omega$ is effectively invertible.

Inversion-hardness from **non-injectivity** and **partiality** (domain complexity).

Using domain complexity

The true sentences of arithmetic form a Π_2^0 singleton: $\emptyset^\omega \in \Pi_2^0$.

Proposition. There are partial computable $f, h : \subseteq 2^\omega \rightarrow 2^\omega$ such that

- f is injective and no arithmetical g can invert f on any real
- $\text{dom}(h)$ is uncountable and no $g \in \Delta_1^1$ can invert h on any real.

Many other examples using facts from computability:

- positive domain and/or range
- random outputs or even random-preserving

for example $x \oplus z \mapsto x$ restricted to randoms.

Randomized computations

Probabilistic Turing machines

computing $x \mapsto f(x)$ where $f(x)$ is a random variable

input	x	0	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1	...
random	r	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	0	...
output	$f(x)$	0	1	0	1	1	0											...

computing $(x, r) \mapsto f(x, r)$

Randomized computations can occasionally be replaced by deterministic ones.

Probabilistic inversion

Given $f, g : \subseteq 2^\omega \rightarrow 2^\omega$ we say that g is a

- **positive** inversion of f if $\mu(\{y : f(g(y)) = y\}) > 0$
- **probabilistic** inversion of f if $\mu(\{y \oplus r : f(g(y \oplus r)) = y\}) > 0$

Example. There is an effective positive f which is probabilistically invertible but has no positive inversion $\leq_T 0'$.

Restrict $x \oplus z \mapsto z$ to a $\Pi_1^0(0')$ class of 2-randoms

Proposition. The following are equivalent:

- for positive many r there is a positive inversion $g \leq_T r$ of f
- there is a probabilistic inversion $g \leq_T \emptyset$ of f .

Examples

Proposition. There is a partial computable $f : \subseteq 2^\omega \rightarrow 2^\omega$ which is

- random-preserving and nowhere effectively invertible
- a.e. probabilistically invertible.

Theorem

If f is a positive partial computable function then it has a positive inversion $g \leq_T 0''$. If f is total then $g \leq_T 0'$.

Question. What is the complexity of probabilistic inversions of positive partial computable functions?

--

Oneway real functions

Oneway functions

In computational complexity they are finite maps between **strings** that are

easy to compute but hard to invert, even probabilistically

Levin (2023) extended them to **real** functions:

partial computable, positive, with no effective probabilistic inversion.

and **asked if they exist**. Last year (2024)

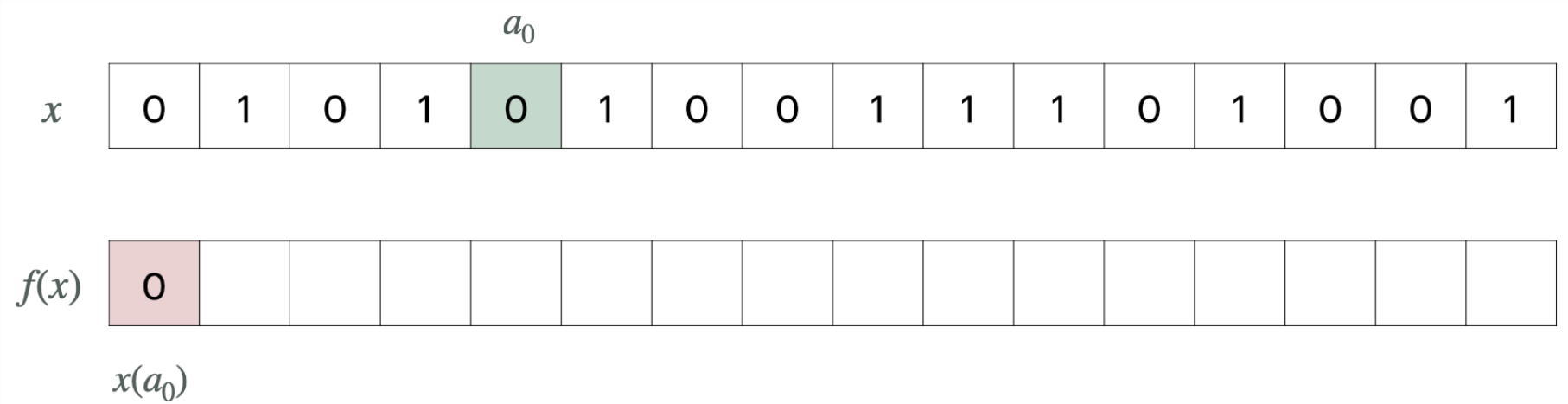
- Gacs constructed one, except that it was not positive
- We came up with a total random-preserving oneway surjection.

Properties of interest: total, surjective, injective, collision-resistant.

Shuffle maps on the reals

A shuffle $f : 2^\omega \rightarrow 2^\omega$ is given by a computable injection $(a_i) \in \omega^\omega$ and

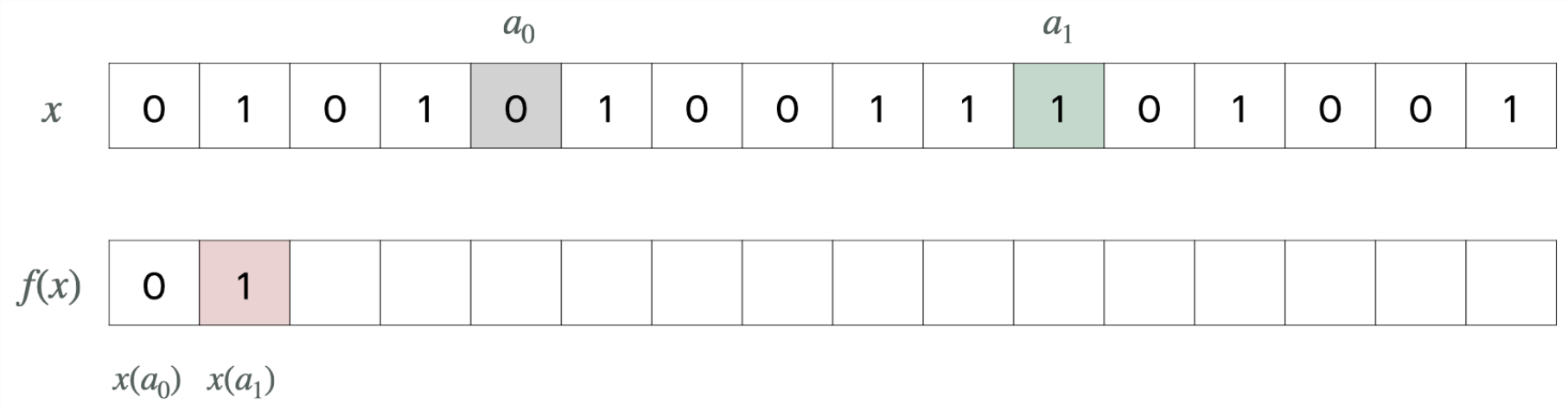
$$f(x)(i) := x(a_i).$$



Shuffle maps on the reals

A **shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by a computable injection $(a_i) \in \omega^\omega$ and

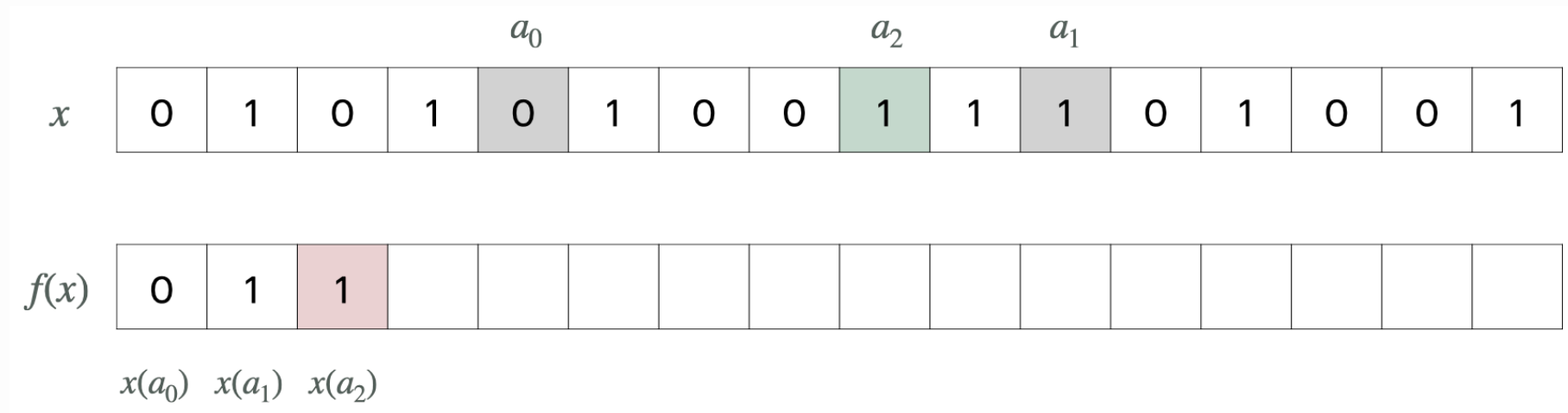
$$f(x)(i) := x(a_i).$$



Shuffle maps on the reals

A **shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by a computable injection $(a_i) \in \omega^\omega$ and

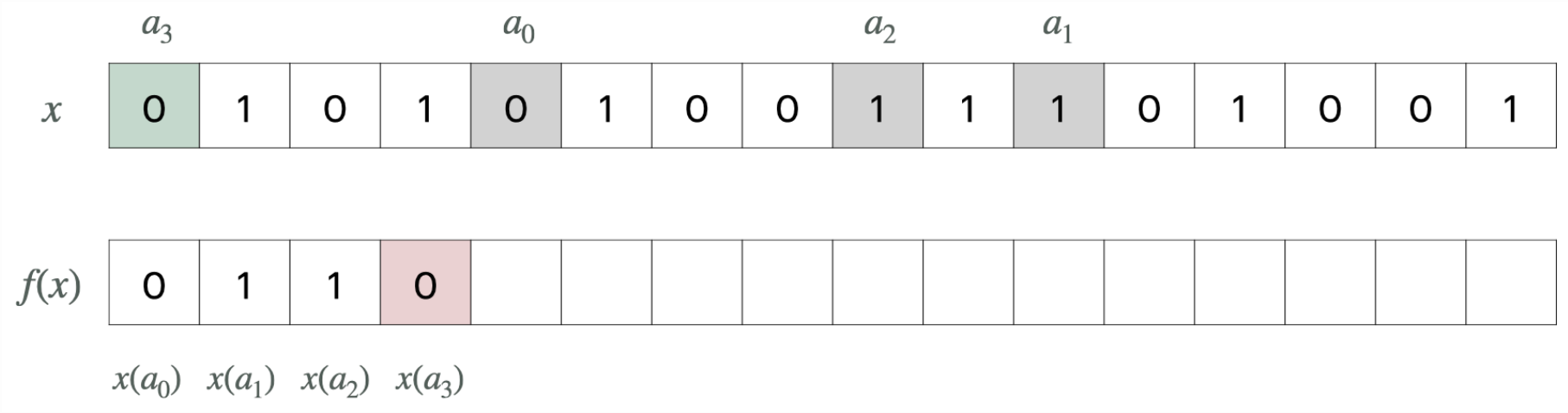
$$f(x)(i) := x(a_i).$$



Shuffle maps on the reals

A shuffle $f : 2^\omega \rightarrow 2^\omega$ is given by a computable injection $(a_i) \in \omega^\omega$ and

$$f(x)(i) := x(a_i).$$



Shuffle maps on the reals

A shuffle $f : 2^\omega \rightarrow 2^\omega$ is given by a computable injection $(a_i) \in \omega^\omega$ and

$$f(x)(i) := x(a_i).$$

	a_3				a_0				a_2		a_1			a_4		
x	0	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1
$f(x)$	0	1	1	0	0											
	$x(a_0)$	$x(a_1)$	$x(a_2)$	$x(a_3)$	$x(a_4)$											

Shuffle maps on the reals

A **shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by a computable injection $(a_i) \in \omega^\omega$ and

$$f(x)(i) := x(a_i).$$

Fact. The (a_i) -shuffle is total computable (as fast as $i \mapsto a_i$) and

- is random-preserving and surjective
- is strongly nowhere injective: $f^{-1}(y)$ is uncountable for each y
- every probabilistic inversion of it computes $\{a_i : i \in \omega\}$.

Letting (a_i) be a $O(i^2)$ -computable enumeration of $0'$ we get:

Theorem. There is a total poly-time computable random-preserving oneway surjection f such that any probabilistic inversion of f computes $0'$.

Why shuffles are oneway

Shuffles f are **nowhere injective** so an inversion g of f

- **makes choices** on the unused bits
- the set of unused bits is **undecidable**.

	a_3			a_0			a_5	a_2		a_1			a_4			
x	0	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1
$f(x)$	0	1	1	0	0	0										
	$x(a_0)$	$x(a_1)$	$x(a_2)$	$x(a_3)$	$x(a_4)$	$x(a_5)$										

Why shuffles are oneway

Therefore if g attempts to effectively invert f :

- input-bits determined by g are **later appended** in the output
- this makes the output **predictable** by g
- almost all outputs are unpredictable (random)

So g fails to invert f almost everywhere.

	a_3			a_0			a_5	a_2		a_1			a_4			
x	0	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1
$f(x)$	0	1	1	0	0	0										
	$x(a_0)$	$x(a_1)$	$x(a_2)$	$x(a_3)$	$x(a_4)$	$x(a_5)$										

Variations on shuffles

If (a_i^z) is the z -computable enumeration of z' and $h^z(x)$ is given by

$$h^z(x)(i) := x(a_i^z)$$

the **relativized shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by

$$f(x \oplus z) := h^z(x) \oplus z.$$

Fact. The relativized shuffle is computable as fast as (a_i) is and

- is random-preserving and surjective
- is oneway and has **no continuous inversion**
- has a positive inversion $g \leq_T 0'$

and is strongly nowhere injective: $f^{-1}(y)$ is uncountable for each y .

Variations on shuffles

Theorem

For each c.e. A there is a total computable function which

- is random-preserving and surjective
- has an A -computable inversion
- is **not** probabilistically invertible on any random $y \not\leq_T A$
- is oneway relative to **any** $w \not\leq_T A$.

Injective oneway functions require partiality:

Proposition. Every total computable random-preserving oneway function is almost nowhere injective: $f^{-1}(f(x))$ is perfect for almost all x .

--

Collisions and hashing

Collision resistance

$\mathcal{C} \subseteq 2^\omega$ is **negligible** if the set of oracles that compute a member of \mathcal{C} is null.

Definition

We say that $f : \subseteq 2^\omega \rightarrow 2^\omega$ is **collision-resistant** if

$$S_f := \{(x, z) : x \neq z \wedge f(x) = f(z)\}$$

is negligible. The members of S_f are called **f -siblings**.

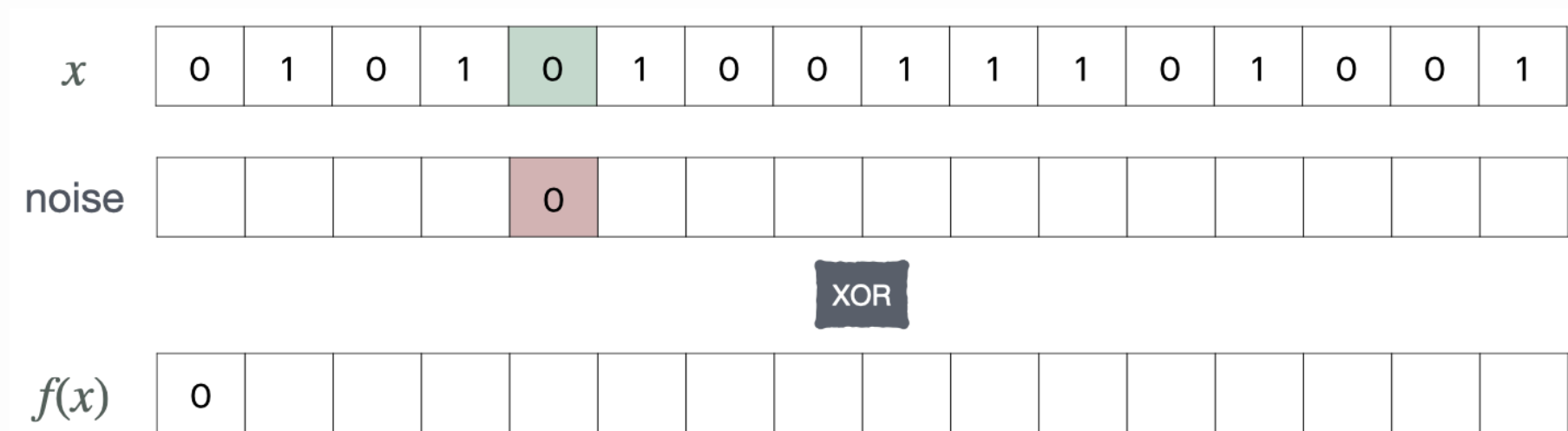
Levin (2024) noticed that shuffles

- can be made **weakly** collision-resistant
- **cannot** be made collision-resistant.

and asked for a collision-resistant computable oneway real function.

Hashing the shuffles

The idea is to XOR the shuffle output with "random" bits.



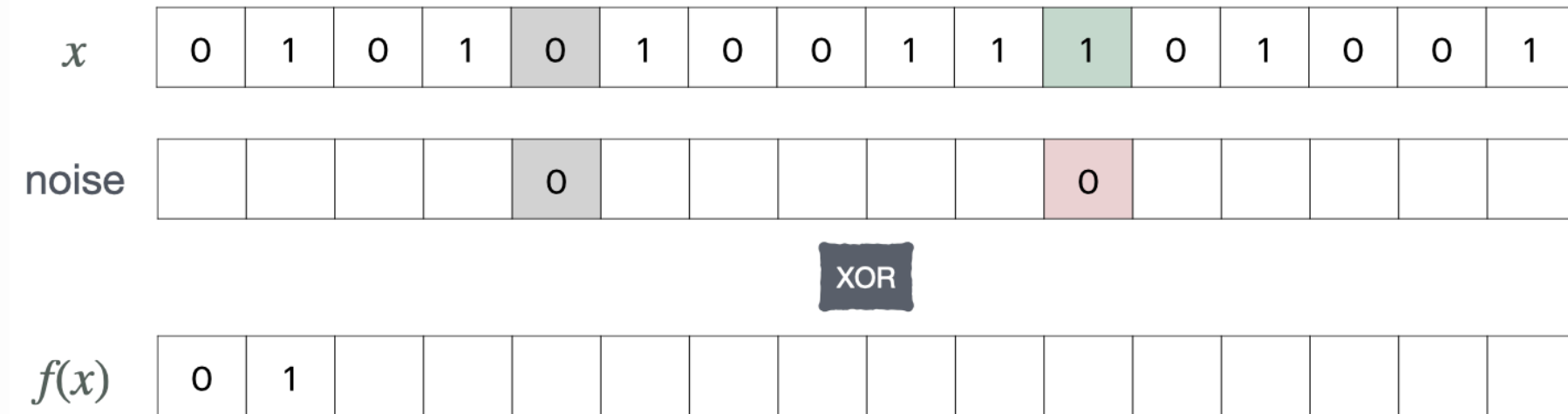
Definition. A **hash-shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by

$$f(x)(i) := x(a_i) \otimes \mathbf{noise}(i)$$

where (a_i) is a computable enumeration of A without repetitions.

Hashing the shuffles

The idea is to XOR the shuffle output with "random" bits.



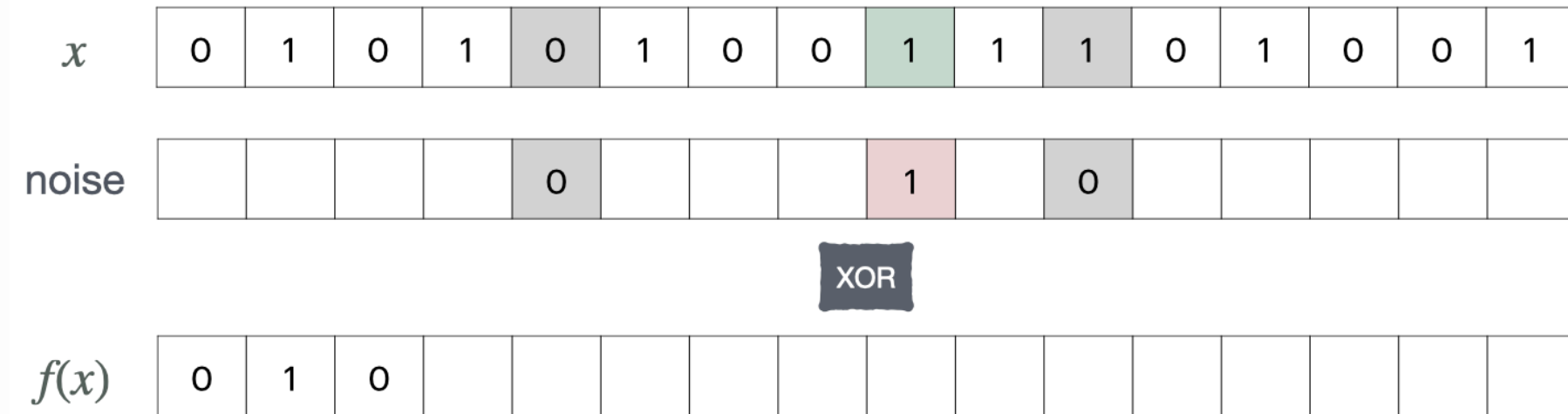
Definition. A **hash-shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by

$$f(x)(i) := x(a_i) \otimes \mathbf{noise}(i)$$

where (a_i) is a computable enumeration of A without repetitions.

Hashing the shuffles

The idea is to XOR the shuffle output with "random" bits.



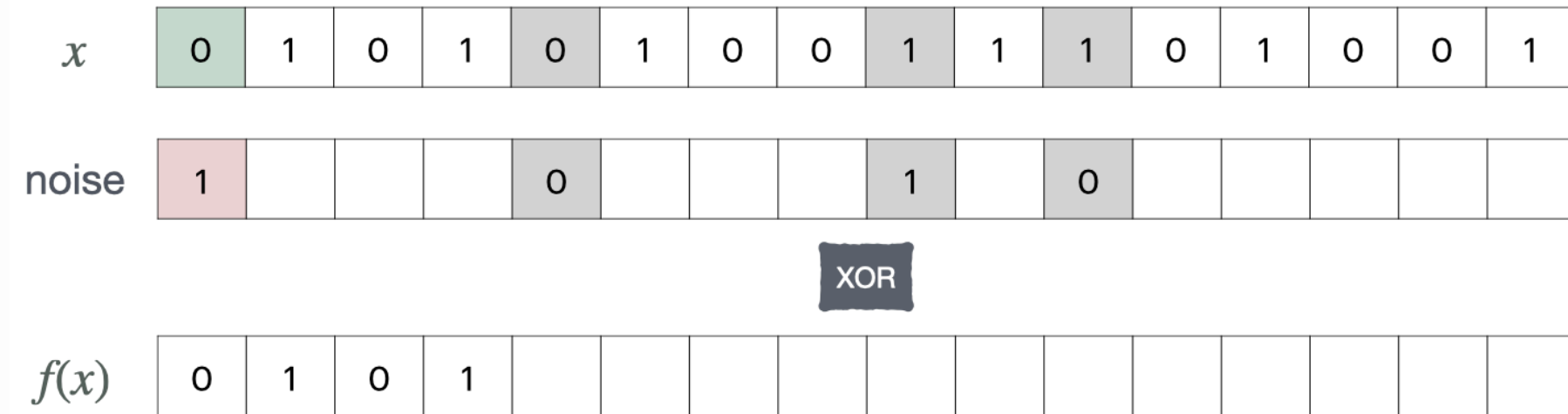
Definition. A **hash-shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by

$$f(x)(i) := x(a_i) \otimes \text{noise}(i)$$

where (a_i) is a computable enumeration of A without repetitions.

Hashing the shuffles

The idea is to XOR the shuffle output with "random" bits.



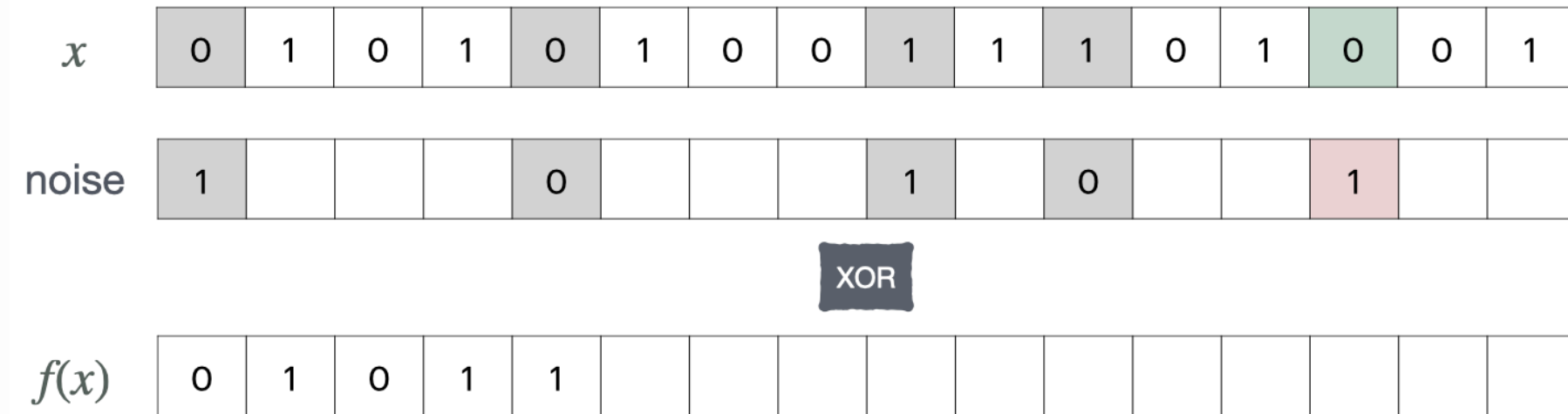
Definition. A **hash-shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by

$$f(x)(i) := x(a_i) \otimes \text{noise}(i)$$

where (a_i) is a computable enumeration of A without repetitions.

Hashing the shuffles

The idea is to XOR the shuffle output with "random" bits.



Definition. A **hash-shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by

$$f(x)(i) := x(a_i) \otimes \text{noise}(i)$$

where (a_i) is a computable enumeration of A without repetitions.

Hashing the shuffles

The idea is to XOR the shuffle output with "random" bits.

Definition. If $A \subseteq \mathbb{N}$ is an infinite c.e. set a computable

$$h : \{\sigma : |\sigma| \in A\} \rightarrow \{0, 1\}$$

is called an **A -hash** or simply a **hash**.

Let \otimes denote XOR. We define **hash-shuffles**.

Definition. The **(h, A) -shuffle** $f : 2^\omega \rightarrow 2^\omega$ is given by

$$f(x)(i) := x(a_i) \otimes h(x \upharpoonright_{a_i})$$

where (a_i) is a computable enumeration of A without repetitions.

Hash-shuffles

Let $A = \{a_i : i \in \mathbb{N}\}$ be c.e. and h be an A -hash.

Proposition

The (h, A) -shuffle is total computable as fast as $h, i \mapsto a_i$ and is

- random-preserving and surjective
- nowhere injective: $f^{-1}(y)$ is uncountable for each y .
- oneway relative to all $w \not\leq_T A$.

But how should we choose h to make f **collision-resistant**?

Let h be the universal partial computable binary predicate $\varphi_i(i)$.

Universal hashing

Let $A := \{\langle \sigma_i, n_i \rangle : i \in \mathbb{N}\}$ where

- (σ_i, n_i) is an effective enumeration of $2^{<\omega} \times \emptyset'$ and
- $(\sigma, n) \mapsto \langle \sigma, n \rangle$ is an effective bijection between $2^{<\omega} \times \emptyset', \mathbb{N}$

and define the **A-hash**:

$$h(\tau) := \begin{cases} \varphi_{n_i}(n_i) & \text{if } \sigma_i \prec \tau \wedge \tau \in 2^{\langle \sigma_i, n_i \rangle} \\ 0 & \text{if } \sigma_i \not\prec \tau \wedge \tau \in 2^{\langle \sigma_i, n_i \rangle} \end{cases}$$

so the **h-shuffle** is given by $f(x)(i) =: h(x \upharpoonright_{\langle \sigma_i, n_i \rangle}) \otimes x(\langle \sigma_i, n_i \rangle)$.

Theorem. There is a total computable function which is

- random-preserving and surjective
- oneway and collision-resistant.

Variations on hash-shuffles

How can we control the strength of collision-resistance?

Avoid universality. Use c.e. **computably inseparable** sets B, C .

Every non-computable c.e. degree contains such a pair B, C .

Theorem

For each c.e. A there is a total computable function which is

- random-preserving and surjective
- oneway and collision-resistant relative to almost all oracles
- **not** oneway and **not** collision-resistant relative to A .

--

Weakly oneway functions

Left and right oneway functions (Gacs)

Definition

A partial computable f is **left-oneway** if it has **positive domain** and

$$\mu(\{x \oplus r : f(g(f(x), r)) = f(x)\}) = 0$$

for each partial computable g . It is **right-oneway** if

$$\mu(\{y \oplus r : f(g(y \oplus r)) = y\}) = 0$$

for each partial computable g and it has **positive range**.

These notions are considerably weaker than oneway functions.

Gacs (2024) showed that there is a left-oneway partial computable f .

Oneway versus left and right oneway functions

They need not need to respect measure:

Proposition. There is a

- left-oneway f which is not positive (not random-preserving)
- right-oneway f which is not positive (not random-preserving)

If they do, right-oneway is a stronger notion:

Proposition

- every random-preserving right-oneway f is left-oneway
- there is a random-preserving left-oneway f which is not right-oneway.

Oneway injections

Question. Is there an injective oneway real function?

All constructions of oneway real functions rely on non-injectivity.

We know that oneway injections are:

- partial: their domain does not contain any positive Π_1^0 class
- not oneway relative to any **almost everywhere dominating** oracle.

By a direct measure-type priority construction we show:

Theorem. There is a left-oneway partial computable injection.

--

Nearly injective

We say that $f : 2^\omega \rightarrow 2^\omega$ is **two-to-one** if $\forall y, |f^{-1}(y)| \leq 2$.

Theorem. There is a total poly-time computable $f : 2^\omega \rightarrow 2^\omega$ such that:

- f is a two-to-one random-preserving surjection
- f is almost everywhere effectively invertible
- every $g : \subseteq 2^\omega \rightarrow 2^\omega$ that inverts f computes $0'$

and the latter holds for the restriction of f in any cylinder $[\sigma]$.

The form is $f(x \oplus z) := h^z(x) \oplus z$ where

- h^z selects positions of x -bits **used** in (copied into) $h^z(x)$
- all but at most one position k^z are **used** in $h^z(x)$.

Blueprint for two-to-one functions

- if $\lim_s k_s^z = \infty$ all x -positions are used in $h^z(x)$
- if $\lim_s k_s^z = k^z$ all x -positions **except k^z** are used in $h^z(x)$.

Given $E_s^z \in \Sigma_1^0$ we update the unused k_s^z if

- either $k_s^z \in \emptyset'_s$ which we call a **\emptyset' -permission**
- or $E_s^z(k_s^z)$ which we call a **z -permission**

Goal: given inversion g , for each n produce z, s with $k_s^z = n$ and $\neg E_s^z(k_s^z)$.

Theorem. There is a total poly-time computable $f : 2^\omega \rightarrow 2^\omega$ such that:

- f is a two-to-one random-preserving surjection
- every $g : \subseteq 2^\omega \rightarrow 2^\omega$ that inverts f computes \emptyset'

Let $f(x \oplus z) := h^z(x) \oplus z$ and

$$k_{s+1}^z := \begin{cases} s + 1 & \text{if } k_s^z \in \emptyset'_s \text{ or } z(\langle k_s^z, s \rangle) = 1 \\ k_s^z & \text{otherwise.} \end{cases}$$

To select the next x -bit used in $f(x \oplus z)$ let $h^z(x; s) := x(p_s^z)$ where:

$$p_s^z := \begin{cases} s + 1 & \text{if } k_{s+1}^z = k_s^z \\ k_s^z & \text{otherwise.} \end{cases}$$

Given inversion g , for each n produce z, s with $k_s^z = n$ and $\neg E_s^z(k_s^z)$.

Use $g(f(0^\omega \oplus z))$ to determine if $n \in \emptyset'$.

Theorem. There is a total poly-time computable $f : 2^\omega \rightarrow 2^\omega$ with:

- f is a two-to-one random-preserving surjection
- for each partial $g \not\leq_T \emptyset'$, with positive probability, g fails to invert f and the latter holds for the restriction of f in any cylinder σ .

- $f(x \oplus z) := h^z(x) \oplus z$ and $h^z(x; s) := x(p_s^z)$
- **Issue:** inversions can only be assumed almost total.
- **Solution:** work with incomplete randoms.

Let z^n be the n th column of z and U a member of a universal test.

Updates of k^z coincide with those of d^z and are due to one of:

- $d_s^z \in \emptyset'_s$ which we call \emptyset' -permission of k_s^z at $s + 1$
- $z^{d_s^z} \in U_s$ which we call z -permission of k_s^z at $s + 1$.

Let k_s^z be the non-decreasing counter with

$$k_{s+1}^z := \begin{cases} s + 1 & \text{if } d_s^z \in \emptyset'_s \text{ or } z^{d_s^z} \in U_s \\ k_s^z & \text{otherwise} \end{cases}$$

where $d_s^z := |\{t < s : k_{t+1}^z \neq k_t^z\}|$ counts the updates of k^z and

$$p_s^z := \begin{cases} s + 1 & \text{if } k_{s+1}^z = k_s^z \\ k_s^z & \text{otherwise} \end{cases}$$

$h^z(x; s) := x(p_s^z)$ selects the next x -bit in $f(x \oplus z) := h^z(x) \oplus z$.

Theorem. There is a total poly-time computable $f : 2^\omega \rightarrow 2^\omega$ with:

- f is a two-to-one random-preserving surjection
- for each partial $g \not\leq_T \emptyset'$, with positive probability, g fails to invert f

For each $r \not\leq_T \emptyset'$ there exist y, w such that

- $y \oplus w$ is weakly r -random and y is random
- no column w^n of w is random and $r \oplus y \oplus w \not\leq_T \emptyset'$.

Effectively in $y \oplus w$ and n we can define z and s such that

- $d_s^z = n$ and $(\lim_t k_t^z < \infty \iff \lim_t k_t^z = k_s^z \iff n \notin \emptyset')$
- $y \oplus z$ is weakly r -random.

Given a.e. inversion g , use $g(f(x \oplus z))$ to determine if $n \in \emptyset'$.

--

Unresolved problems

Oracles for inversion

We know that $\mathbf{0}'$ characterizes the strength of total oneway functions.

Question. Which oracles x can probabilistically invert every

1. random-preserving partial computable **function** f ?
2. random-preserving partial computable **injection** f ?

- for (1) $x \geq_T \emptyset''$ is sufficient and $x \geq_T \emptyset'$ is necessary
- for (2) computing an almost everywhere dominating (m_i) suffices.

Question. Is there a partial computable oneway injection?

Making positive maps injective

Question

Which oracles x can compute an injective restriction for each random-preserving partial computable function?

We know that $\mathbf{0}'$ suffices.

Theorem

If f is partial computable and random-preserving there is h which

- is an injective restriction of f and $h \leq_T \mathbf{0}'$
- has $\Pi_1^0(\emptyset'')$ domain and positive $\Pi_1^0(\emptyset'')$ range.

--

Thank you for your attention!

Thanks to Leonid Levin and Yu Liang and his students

References

- Zermelo-Fraenkel Axioms, Internal Classes, External Sets - Levin [ArXiv 2209.07497](#)
- Computable one-way functions on the reals - [Arxiv 2406.15817](#)
- Complexity of inversion of functions on the reals - [Arxiv 2412.07592](#)
- Collision-resistant hash-shuffles on the reals - [Arxiv 2501.02604](#)