Von Neumann's biased coin revisited

Benoit Monin - LIAFA - University of Paris VII

Join work with Laurent Bienvenu - CNRS & University of Paris VII

29 June 2012

Von Neumann's coin trick

Section 1

Von Neumann's coin trick

Von Neumann's coin trick

I want to play Head or Tail

Suppose that you want to play a fair game of "head or tail", but all you have at your disposal is a biased coin, and you don't know the bias.

How to achieve this ?

An easy but nice solution is to group the bits two by two, then you replace 01 by 0, replace 10 by 1 and you discard blocks 00 and 11.

Von Neumann's coin trick

I want to play Head or Tail

Suppose that you want to play a fair game of "head or tail", but all you have at your disposal is a biased coin, and you don't know the bias.

How to achieve this ?

An easy but nice solution is to group the bits two by two, then you replace 01 by 0, replace 10 by 1 and you discard blocks 00 and 11.

Von Neumann's coin trick example

Example

The biased coin : $P(head) = p$ and $P(tail) = 1 - p$

Von Neumann's coin trick example

Example

The biased coin : $P(head) = p$ and $P(tail) = 1 - p$
The first results : 1101111001011011011111100

Von Neumann's coin trick example

Example

The biased coin : $P(head) = p$ and $P(tail) = 1 - p$
The first results : 11011110010110110111100
The trick :

$\underbrace{11}$	$\underbrace{01}$	$\underbrace{11}$	$\underbrace{10}$	$\underbrace{01}$	$\underbrace{01}$	$\underbrace{10}$	$\underbrace{11}$	$\underbrace{01}$	$\underbrace{11}$	$\underbrace{11}$	$\underbrace{00}$
$\underbrace{p^2}$	$\underbrace{p(1-p)}$	$\underbrace{p^2}$	$\underbrace{p(1-p)}$	$\underbrace{p(1-p)}$	$\underbrace{p(1-p)}$	$\underbrace{p(1-p)}$	$\underbrace{p^2}$	$\underbrace{p(1-p)}$	$\underbrace{p^2}$	$\underbrace{p^2}$	$\underbrace{(1-p)^2}$
–	0	–	1	0	0	1	–	0	–	–	–

Von Neumann's coin trick example

Example

The biased coin : $P(head) = p$ and $P(tail) = 1 - p$
The first results : 11011110010110110111100
The trick :

$\underbrace{11}$	$\underbrace{01}$	$\underbrace{11}$	$\underbrace{10}$	$\underbrace{01}$	$\underbrace{01}$	$\underbrace{10}$	$\underbrace{11}$	$\underbrace{01}$	$\underbrace{11}$	$\underbrace{11}$	$\underbrace{00}$
$\underbrace{p^2}$	$\underbrace{p(1-p)}$	$\underbrace{p^2}$	$\underbrace{p(1-p)}$	$\underbrace{p(1-p)}$	$\underbrace{p(1-p)}$	$\underbrace{p(1-p)}$	$\underbrace{p^2}$	$\underbrace{p(1-p)}$	$\underbrace{p^2}$	$\underbrace{p^2}$	$\underbrace{(1-p)^2}$
–	0	–	1	0	0	1	–	0	–	–	–

The fair coin tossing : 010010

Von Neumann's coin trick example

Nice things about von Neumann's trick :

- We have a computable extraction procedure.
- It works even if the measure is not computable.
- It is uniform for all Bernoulli measures (except trivial ones) and all of their random elements.

Von Neumann's coin trick example

Nice things about von Neumann's trick :

- We have a computable extraction procedure.
- It works even if the measure is not computable.
- It is uniform for all Bernoulli measures (except trivial ones) and all of their random elements.

Von Neumann's coin trick example

Nice things about von Neumann's trick :

- We have a computable extraction procedure.
- It works even if the measure is not computable.
- It is uniform for all Bernoulli measures (except trivial ones) and all of their random elements.

Von Neumann's coin trick example

Nice things about von Neumann's trick :

- We have a computable extraction procedure.
- It works even if the measure is not computable.
- It is uniform for all Bernoulli measures (except trivial ones) and all of their random elements.

A more general framework

On a more abstract level, the situation is the following :

- We have access to a random sequence for a given measure μ which we do not know.
- However, we do know that μ belongs to some particular class C.
- Based on this information we are able to build a computable procedure which works for all $\mu \in C$.

A more general framework

On a more abstract level, the situation is the following :

- We have access to a random sequence for a given measure μ which we do not know.
- However, we do know that μ belongs to some particular class C.
- Based on this information we are able to build a computable procedure which works for all $\mu \in C$.

A more general framework

On a more abstract level, the situation is the following :

- We have access to a random sequence for a given measure μ which we do not know.
- However, we do know that μ belongs to some particular class C.
- Based on this information we are able to build a computable procedure which works for all $\mu \in C$.

A more general framework

On a more abstract level, the situation is the following :

- We have access to a random sequence for a given measure μ which we do not know.
- However, we do know that μ belongs to some particular class C.
- Based on this information we are able to build a computable procedure which works for all $\mu \in C$.

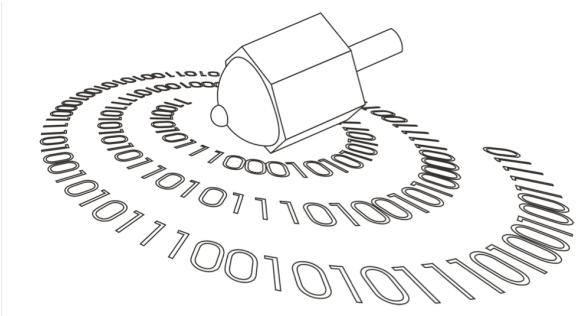
A more general framework

On a more abstract level, the situation is the following :

- We have access to a random sequence for a given measure μ which we do not know.
- However, we do know that μ belongs to some particular class C.
- Based on this information we are able to build a computable procedure which works for all $\mu \in C$.

For which other class C can such an extraction procedure be built ?

Algorithmic randomness



Section 2

Algorithmic randomness

Algorithmic randomness

Algorithmic randomness :

What does it mean for a string to be random ?

Are

c :0000000000000010000000001000000000010000000000001 . . .

or

π :00100100001111110110101010001000100001011010001100 . . .

random ?

Algorithmic randomness

Algorithmic randomness :

What does it mean for a string to be random ?

Intuition

A sequence of 2^ω should be random if it belongs to the smallest set of measure 1.

Definition (Martin-Löf)

A sequence of 2^ω is **Martin-Löf random** if it belongs to the smallest Σ^0_2 set, effectively of measure 1.

Algorithmic randomness

Algorithmic randomness :

What does it mean for a string to be random ?

Intuition

A sequence of 2^ω should be random if it belongs to the smallest set of measure 1.

Definition (Martin-Löf)

A sequence of 2^ω is **Martin-Löf random** if it belongs to the smallest Σ^0_2 set, effectively of measure 1.

Algorithmic randomness

Definition (Martin-Löf test)

A Π^0_2 subset of 2^ω is a Martin-Löf test if it is effectively of measure 0, which means that the n-th open set of the intersection should be of measure less than 2^{-n}.

Definition (Martin-Löf test)

There is a largest Martin-Löf test. A sequence is not Martin-Löf random if it belongs to the largest Martin-Löf test.

Algorithmic randomness

Definition (Martin-Löf test)

A Π^0_2 subset of 2^ω is a Martin-Löf test if it is effectively of measure 0, which means that the n-th open set of the intersection should be of measure less than 2^{-n}.

Definition (Martin-Löf test)

There is a largest Martin-Löf test. A sequence is not Martin-Löf random if it belongs to the largest Martin-Löf test.

Algorithmic randomness : Martin-Löf test example

Illustration of a test :

Measure	$\mathbf{f(., 1)}$	$\mathbf{f(., 2)}$	$\mathbf{f(., 3)}$	$\mathbf{f(., 4)}$	$\dots$
$\lambda(f(1,\mathbb{N})) \leqslant \frac{1}{2}$	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	$\dots$
$\lambda(f(2,\mathbb{N})) \leqslant \frac{1}{4}$	$\sigma_{2,1}$	$\sigma_{2,2}$	$\sigma_{2,3}$	$\sigma_{2,4}$	$\dots$
$\lambda(f(3,\mathbb{N})) \leqslant \frac{1}{8}$	$\sigma_{3,1}$	$\sigma_{3,2}$	$\sigma_{3,3}$	$\sigma_{3,4}$	$\dots$
$\lambda(f(4,\mathbb{N})) \leqslant \frac{1}{16}$	$\sigma_{4,1}$	$\sigma_{4,2}$	$\sigma_{4,3}$	$\sigma_{4,4}$	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

Algorithmic randomness : universal Martin-Löf test

Universal test :

Measure	Test 1	Test 2	Test 3	Test 4	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{1,i}\right) \leqslant \frac{1}{2}$	$(\sigma^1_{1,i})_{i\in\mathbb{N}}$	$(\sigma^2_{1,i})_{i\in\mathbb{N}}$	$(\sigma^3_{1,i})_{i\in\mathbb{N}}$	$(\sigma^4_{1,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{2,i}\right) \leqslant \frac{1}{4}$	$(\sigma^1_{2,i})_{i\in\mathbb{N}}$	$(\sigma^2_{2,i})_{i\in\mathbb{N}}$	$(\sigma^3_{2,i})_{i\in\mathbb{N}}$	$(\sigma^4_{2,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{3,i}\right) \leqslant \frac{1}{8}$	$(\sigma^1_{3,i})_{i\in\mathbb{N}}$	$(\sigma^2_{3,i})_{i\in\mathbb{N}}$	$(\sigma^3_{3,i})_{i\in\mathbb{N}}$	$(\sigma^4_{3,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{4,i}\right) \leqslant \frac{1}{16}$	$(\sigma^1_{4,i})_{i\in\mathbb{N}}$	$(\sigma^2_{4,i})_{i\in\mathbb{N}}$	$(\sigma^3_{4,i})_{i\in\mathbb{N}}$	$(\sigma^4_{4,i})_{i\in\mathbb{N}}$	...
...	...	...	...	...	...

Algorithmic randomness : universal Martin-Löf test

Universal test :

Measure	Test 1	Test 2	Test 3	Test 4	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{1,i}\right) \leqslant \frac{1}{2}$	$(\sigma^1_{1,i})_{i\in\mathbb{N}}$	$(\sigma^2_{1,i})_{i\in\mathbb{N}}$	$(\sigma^3_{1,i})_{i\in\mathbb{N}}$	$(\sigma^4_{1,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{2,i}\right) \leqslant \frac{1}{4}$	$(\sigma^1_{2,i})_{i\in\mathbb{N}}$	$(\sigma^2_{2,i})_{i\in\mathbb{N}}$	$(\sigma^3_{2,i})_{i\in\mathbb{N}}$	$(\sigma^4_{2,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{3,i}\right) \leqslant \frac{1}{8}$	$(\sigma^1_{3,i})_{i\in\mathbb{N}}$	$(\sigma^2_{3,i})_{i\in\mathbb{N}}$	$(\sigma^3_{3,i})_{i\in\mathbb{N}}$	$(\sigma^4_{3,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{4,i}\right) \leqslant \frac{1}{16}$	$(\sigma^1_{4,i})_{i\in\mathbb{N}}$	$(\sigma^2_{4,i})_{i\in\mathbb{N}}$	$(\sigma^3_{4,i})_{i\in\mathbb{N}}$	$(\sigma^4_{4,i})_{i\in\mathbb{N}}$	...
...	...	...	...	...	...

Algorithmic randomness : universal Martin-Löf test

Universal test :

Measure	Test 1	Test 2	Test 3	Test 4	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{1,i}\right) \leqslant \frac{1}{2}$	$(\sigma^1_{1,i})_{i\in\mathbb{N}}$	$(\sigma^2_{1,i})_{i\in\mathbb{N}}$	$(\sigma^3_{1,i})_{i\in\mathbb{N}}$	$(\sigma^4_{1,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{2,i}\right) \leqslant \frac{1}{4}$	$(\sigma^1_{2,i})_{i\in\mathbb{N}}$	$(\sigma^2_{2,i})_{i\in\mathbb{N}}$	$(\sigma^3_{2,i})_{i\in\mathbb{N}}$	$(\sigma^4_{2,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{3,i}\right) \leqslant \frac{1}{8}$	$(\sigma^1_{3,i})_{i\in\mathbb{N}}$	$(\sigma^2_{3,i})_{i\in\mathbb{N}}$	$(\sigma^3_{3,i})_{i\in\mathbb{N}}$	$(\sigma^4_{3,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{4,i}\right) \leqslant \frac{1}{16}$	$(\sigma^1_{4,i})_{i\in\mathbb{N}}$	$(\sigma^2_{4,i})_{i\in\mathbb{N}}$	$(\sigma^3_{4,i})_{i\in\mathbb{N}}$	$(\sigma^4_{4,i})_{i\in\mathbb{N}}$	...
...	...	...	...	...	...

Algorithmic randomness : universal Martin-Löf test

Universal test :

Measure	Test 1	Test 2	Test 3	Test 4	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{1,i}\right) \leqslant \frac{1}{2}$	$(\sigma^1_{1,i})_{i\in\mathbb{N}}$	$(\sigma^2_{1,i})_{i\in\mathbb{N}}$	$(\sigma^3_{1,i})_{i\in\mathbb{N}}$	$(\sigma^4_{1,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{2,i}\right) \leqslant \frac{1}{4}$	$(\sigma^1_{2,i})_{i\in\mathbb{N}}$	$(\sigma^2_{2,i})_{i\in\mathbb{N}}$	$(\sigma^3_{2,i})_{i\in\mathbb{N}}$	$(\sigma^4_{2,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{3,i}\right) \leqslant \frac{1}{8}$	$(\sigma^1_{3,i})_{i\in\mathbb{N}}$	$(\sigma^2_{3,i})_{i\in\mathbb{N}}$	$(\sigma^3_{3,i})_{i\in\mathbb{N}}$	$(\sigma^4_{3,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{4,i}\right) \leqslant \frac{1}{16}$	$(\sigma^1_{4,i})_{i\in\mathbb{N}}$	$(\sigma^2_{4,i})_{i\in\mathbb{N}}$	$(\sigma^3_{4,i})_{i\in\mathbb{N}}$	$(\sigma^4_{4,i})_{i\in\mathbb{N}}$	...
...	...	...	...	...	...

Algorithmic randomness : universal Martin-Löf test

Universal test :

Measure	Test 1	Test 2	Test 3	Test 4	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{1,i}\right) \leqslant \frac{1}{2}$	$(\sigma^1_{1,i})_{i\in\mathbb{N}}$	$(\sigma^2_{1,i})_{i\in\mathbb{N}}$	$(\sigma^3_{1,i})_{i\in\mathbb{N}}$	$(\sigma^4_{1,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{2,i}\right) \leqslant \frac{1}{4}$	$(\sigma^1_{2,i})_{i\in\mathbb{N}}$	$(\sigma^2_{2,i})_{i\in\mathbb{N}}$	$(\sigma^3_{2,i})_{i\in\mathbb{N}}$	$(\sigma^4_{2,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{3,i}\right) \leqslant \frac{1}{8}$	$(\sigma^1_{3,i})_{i\in\mathbb{N}}$	$(\sigma^2_{3,i})_{i\in\mathbb{N}}$	$(\sigma^3_{3,i})_{i\in\mathbb{N}}$	$(\sigma^4_{3,i})_{i\in\mathbb{N}}$	...
$\forall n\ \lambda\left(\bigcup_{i\in\mathbb{N}} \sigma^n_{4,i}\right) \leqslant \frac{1}{16}$	$(\sigma^1_{4,i})_{i\in\mathbb{N}}$	$(\sigma^2_{4,i})_{i\in\mathbb{N}}$	$(\sigma^3_{4,i})_{i\in\mathbb{N}}$	$(\sigma^4_{4,i})_{i\in\mathbb{N}}$	...
...	...	...	...	...	...

Algorithmic randomness : Integrable test

Switch to analysis

To test the randomness of sequences, we can equivalently use a more analytical notion.

Intuition

We can define $\mathbf{t} : 2^\omega \to \overline{\mathbb{R}^+}$ to be on a string x : The smallest n such that x does not belong to the n-th open set of the universal test.

Algorithmic randomness : Integrable test

Switch to analysis

To test the randomness of sequences, we can equivalently use a more analytical notion.

Intuition

We can define $\mathbf{t} : 2^\omega \to \overline{\mathbb{R}^+}$ to be on a string x : The smallest n such that x does not belong to the n-th open set of the universal test.

Algorithmic randomness : Integrable test

Illustration $\mathbf{t}(x) = 4$:

Num	1	2	3	4	5	6	7	8	...
1	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	$\sigma_{1,5}$	$\sigma_{1,6}$	$\sigma_{1,7}$	$\sigma_{1,8}$	...
2	$\sigma_{2,1}$	$\sigma_{2,2}$	$\sigma_{2,3}$	$\sigma_{2,4}$	$\sigma_{2,5}$	$\sigma_{2,6}$	$\sigma_{2,7}$	$\sigma_{2,8}$	...
3	$\sigma_{3,1}$	$\sigma_{3,2}$	$\sigma_{3,3}$	$\sigma_{3,4}$	$\sigma_{3,5}$	$\sigma_{3,6}$	$\sigma_{3,7}$	$\sigma_{3,8}$	...
4	$\sigma_{4,1}$	$\sigma_{4,2}$	$\sigma_{4,3}$	$\sigma_{4,4}$	$\sigma_{4,5}$	$\sigma_{4,6}$	$\sigma_{4,7}$	$\sigma_{4,8}$	...
...	...	...	...	...	...	...	...	...	...

Algorithmic randomness : Integrable test

Illustration $\mathbf{t}(x) = 4$:

Num	1	2	3	4	5	6	7	8	...
1	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	$\sigma_{1,5}$	$\sigma_{1,6}$	$\sigma_{1,7}$	$\sigma_{1,8}$	...
2	$\sigma_{2,1}$	$\sigma_{2,2}$	$\sigma_{2,3}$	$\sigma_{2,4}$	$\sigma_{2,5}$	$\sigma_{2,6}$	$\sigma_{2,7}$	$\sigma_{2,8}$	...
3	$\sigma_{3,1}$	$\sigma_{3,2}$	$\sigma_{3,3}$	$\sigma_{3,4}$	$\sigma_{3,5}$	$\sigma_{3,6}$	$\sigma_{3,7}$	$\sigma_{3,8}$	...
4	$\sigma_{4,1}$	$\sigma_{4,2}$	$\sigma_{4,3}$	$\sigma_{4,4}$	$\sigma_{4,5}$	$\sigma_{4,6}$	$\sigma_{4,7}$	$\sigma_{4,8}$	...
...	...	...	...	...	...	...	...	...	...

Algorithmic randomness : Integrable test

Illustration $\mathbf{t}(x) = 4$:

Num	1	2	3	4	5	6	7	8	...
1	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	$\sigma_{1,5}$	$\sigma_{1,6}$	$\sigma_{1,7}$	$\sigma_{1,8}$	...
2	$\sigma_{2,1}$	$\sigma_{2,2}$	$\sigma_{2,3}$	$\sigma_{2,4}$	$\sigma_{2,5}$	$\sigma_{2,6}$	$\sigma_{2,7}$	$\sigma_{2,8}$	...
3	$\sigma_{3,1}$	$\sigma_{3,2}$	$\sigma_{3,3}$	$\sigma_{3,4}$	$\sigma_{3,5}$	$\sigma_{3,6}$	$\sigma_{3,7}$	$\sigma_{3,8}$	...
4	$\sigma_{4,1}$	$\sigma_{4,2}$	$\sigma_{4,3}$	$\sigma_{4,4}$	$\sigma_{4,5}$	$\sigma_{4,6}$	$\sigma_{4,7}$	$\sigma_{4,8}$	...
...	...	...	...	...	...	...	...	...	...

Algorithmic randomness : Integrable test

Illustration $\mathbf{t}(x) = 4$:

Num	1	2	3	4	5	6	7	8	...
1	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	$\sigma_{1,5}$	$\sigma_{1,6}$	$\sigma_{1,7}$	$\sigma_{1,8}$	...
2	$\sigma_{2,1}$	$\sigma_{2,2}$	$\sigma_{2,3}$	$\sigma_{2,4}$	$\sigma_{2,5}$	$\sigma_{2,6}$	$\sigma_{2,7}$	$\sigma_{2,8}$	...
3	$\sigma_{3,1}$	$\sigma_{3,2}$	$\sigma_{3,3}$	$\sigma_{3,4}$	$\sigma_{3,5}$	$\sigma_{3,6}$	$\sigma_{3,7}$	$\sigma_{3,8}$	...
4	$\sigma_{4,1}$	$\sigma_{4,2}$	$\sigma_{4,3}$	$\sigma_{4,4}$	$\sigma_{4,5}$	$\sigma_{4,6}$	$\sigma_{4,7}$	$\sigma_{4,8}$	...
...	...	...	...	...	...	...	...	...	...

Properties of $\mathbf{t}$

We have that :

- $\mathbf{t}$ is computabily approximable from below, uniformily in x ($\mathbf{t}$ is lower semi-computable).
- $\int \mathbf{t}(x)dx$ is finite (as the $\sum_n (n+1)2^{-n}$ is finite).

Definition (integrable test)

Such a function is called an **integrable test**. There is a universal integrable test.

Randomness

We have that x is random iff $\mathbf{t}(x)$ is finite for all integrable tests iff $\mathbf{t}(x)$ is finite for the universal integrable test.

Properties of $\mathbf{t}$

We have that :

- $\mathbf{t}$ is computabily approximable from below, uniformily in x ($\mathbf{t}$ is lower semi-computable).
- $\int \mathbf{t}(x)dx$ is finite (as the $\sum_n (n+1)2^{-n}$ is finite).

Definition (integrable test)

Such a function is called an **integrable test**. There is a universal integrable test.

Randomness

We have that x is random iff $\mathbf{t}(x)$ is finite for all integrable tests iff $\mathbf{t}(x)$ is finite for the universal integrable test.

Properties of $\mathbf{t}$

We have that :

- $\mathbf{t}$ is computabily approximable from below, uniformily in x ($\mathbf{t}$ is lower semi-computable).
- $\int \mathbf{t}(x)dx$ is finite (as the $\sum_n (n+1)2^{-n}$ is finite).

Definition (integrable test)

Such a function is called an **integrable test**. There is a universal integrable test.

Randomness

We have that x is random iff $\mathbf{t}(x)$ is finite for all integrable tests iff $\mathbf{t}(x)$ is finite for the universal integrable test.

Properties of $\mathbf{t}$

We have that :

- $\mathbf{t}$ is computabily approximable from below, uniformily in x ($\mathbf{t}$ is lower semi-computable).
- $\int \mathbf{t}(x)dx$ is finite (as the $\sum_n (n+1)2^{-n}$ is finite).

Definition (integrable test)

Such a function is called an **integrable test**. There is a universal integrable test.

Randomness

We have that x is random iff $\mathbf{t}(x)$ is finite for all integrable tests iff $\mathbf{t}(x)$ is finite for the universal integrable test.

Properties of $\mathbf{t}$

We have that :

- $\mathbf{t}$ is computabily approximable from below, uniformily in x ($\mathbf{t}$ is lower semi-computable).
- $\int \mathbf{t}(x)dx$ is finite (as the $\sum_n (n+1)2^{-n}$ is finite).

Definition (integrable test)

Such a function is called an **integrable test**. There is a universal integrable test.

Randomness

We have that x is random iff $\mathbf{t}(x)$ is finite for all integrable tests iff $\mathbf{t}(x)$ is finite for the universal integrable test.

Algorithmic randomness for other measures

The space of probability measures on 2^ω will be denoted by $\mathcal{M}(2^\omega)$.

Question

What if we want to define random sequences obtained by flipping a biased coin ? The definition generalizes itself pretty well as long as the measure is computable.

Definition (Martin-Löf randomness for computable measure)

Let μ be a computable measure. A sequence of 2^ω is **Martin-Löf random for the measure μ** if it belongs to the smallest Σ^0_2 set, effectively of μ measure 1.

Algorithmic randomness for other measures

The space of probability measures on 2^ω will be denoted by $\mathcal{M}(2^\omega)$.

Question

What if we want to define random sequences obtained by flipping a biased coin ? The definition generalizes itself pretty well as long as the measure is computable.

Definition (Martin-Löf randomness for computable measure)

Let μ be a computable measure. A sequence of 2^ω is **Martin-Löf random for the measure μ** if it belongs to the smallest Σ^0_2 set, effectively of μ measure 1.

Algorithmic randomness for other measures

The space of probability measures on 2^ω will be denoted by $\mathcal{M}(2^\omega)$.

Question

What if we want to define random sequences obtained by flipping a biased coin ? The definition generalizes itself pretty well as long as the measure is computable.

Definition (Martin-Löf randomness for computable measure)

Let μ be a computable measure. A sequence of 2^ω is **Martin-Löf random for the measure μ** if it belongs to the smallest Σ^0_2 set, effectively of μ measure 1.

Algorithmic randomness for other measures

Problem

When the measure is not computable, we cannot necessarily obtain universal Martin-Löf test for the measure...

Intuition

A possibility is to add the measure as an oracle to create our test, but a measure can have many different binary representations having different Turing-degrees. So it is not clear which one to choose.

Algorithmic randomness for other measures

Problem

When the measure is not computable, we cannot necessarily obtain universal Martin-Löf test for the measure...

Intuition

A possibility is to add the measure as an oracle to create our test, but a measure can have many different binary representations having different Turing-degrees. So it is not clear which one to choose.

Algorithmic randomness for other measures

Idea

You can take a representation such that any other representation can compute it. (the smallest one).

Theorem (Day, Miller)

Some measures does not have a smallest representation in the Turing degree !

Solution

Instead of using representations, we should extend the notion of computability to the space of measures

Algorithmic randomness for other measures

Idea

You can take a representation such that any other representation can compute it. (the smallest one).

Theorem (Day, Miller)

Some measures does not have a smallest representation in the Turing degree !

Solution

Instead of using representations, we should extend the notion of computability to the space of measures

Algorithmic randomness for other measures

Idea

You can take a representation such that any other representation can compute it. (the smallest one).

Theorem (Day, Miller)

Some measures does not have a smallest representation in the Turing degree !

Solution

Instead of using representations, we should extend the notion of computability to the space of measures

Algorithmic randomness for other measures

Computability in $\mathcal{M}(2^\omega)$

How can we extend the notion of integrable test $\mathbf{t} : \mathcal{M}(2^\omega) \times 2^\omega \to \overline{\mathbb{R}^+}$?

What do we do in $\mathbb{R}$?

In $\mathbb{R}$ we say that a function f is computable if from any fast cauchy sequence converging to x we can output a fast cauchy sequence converging to $f(x)$.

What do we do in $\mathbb{R}$?

Equivalently, f is computable if from any sequence of all intervals with rationals endpoints containing x, f can output the sequence of intervals with rationals endpoints contaning $f(x)$.

Algorithmic randomness for other measures

Computability in $\mathcal{M}(2^\omega)$

How can we extend the notion of integrable test $\mathbf{t} : \mathcal{M}(2^\omega) \times 2^\omega \to \overline{\mathbb{R}^+}$?

What do we do in $\mathbb{R}$?

In $\mathbb{R}$ we say that a function f is computable if from any fast cauchy sequence converging to x we can output a fast cauchy sequence converging to $f(x)$.

What do we do in $\mathbb{R}$?

Equivalently, f is computable if from any sequence of all intervals with rationals endpoints containing x, f can output the sequence of intervals with rationals endpoints contaning $f(x)$.

Algorithmic randomness for other measures

Computability in $\mathcal{M}(2^\omega)$

How can we extend the notion of integrable test $\mathbf{t} : \mathcal{M}(2^\omega) \times 2^\omega \to \overline{\mathbb{R}^+}$?

What do we do in $\mathbb{R}$?

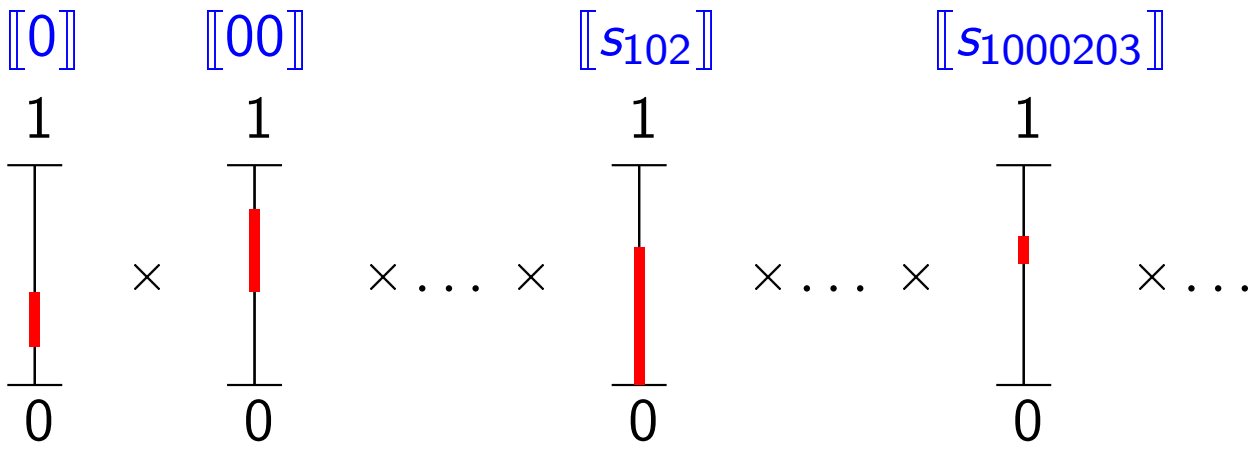
In $\mathbb{R}$ we say that a function f is computable if from any fast cauchy sequence converging to x we can output a fast cauchy sequence converging to $f(x)$.

What do we do in $\mathbb{R}$?

Equivalently, f is computable if from any sequence of all intervals with rationals endpoints containing x, f can output the sequence of intervals with rationals endpoints contaning $f(x)$.

Algorithmic randomness for different measures

A basic open set in the space of measure :



$$\mathcal{M}(2^\omega) \subseteq [0,1]^{\mathbb{N}}$$

Algorithmic randomness for different measures

The space of measures

- The space of measure is a closed subset of $[0,1]^{\mathbb{N}}$.
  $\forall s \in 2^{\omega} \quad \mu(s) = \mu(s0) + \mu(s1)$.
- The topology is the one induced by the product topology on $[0,1]^{\mathbb{N}}$.
- A measure is computable iff the set of basic open sets containing it is effectively enumerable.

Integrable tests

To define what it means for a point $x \in 2^{\omega}$ to be μ-**MLR**, we extend the notion of integrable test.

Algorithmic randomness for different measures

The space of measures

- The space of measure is a closed subset of $[0,1]^{\mathbb{N}}$.
  $\forall s \in 2^{\omega} \quad \mu(s) = \mu(s0) + \mu(s1)$.
- The topology is the one induced by the product topology on $[0,1]^{\mathbb{N}}$.
- A measure is computable iff the set of basic open sets containing it is effectively enumerable.

Integrable tests

To define what it means for a point $x \in 2^{\omega}$ to be μ-**MLR**, we extend the notion of integrable test.

Algorithmic randomness for different measures

The space of measures

- The space of measure is a closed subset of $[0,1]^{\mathbb{N}}$.
  $\forall s \in 2^{\omega} \quad \mu(s) = \mu(s0) + \mu(s1)$.
- The topology is the one induced by the product topology on $[0,1]^{\mathbb{N}}$.
- A measure is computable iff the set of basic open sets containing it is effectively enumerable.

Integrable tests

To define what it means for a point $x \in 2^{\omega}$ to be μ-**MLR**, we extend the notion of integrable test.

Algorithmic randomness for different measures

The space of measures

- The space of measure is a closed subset of $[0,1]^{\mathbb{N}}$.
  $\forall s \in 2^{\omega} \quad \mu(s) = \mu(s0) + \mu(s1)$.
- The topology is the one induced by the product topology on $[0,1]^{\mathbb{N}}$.
- A measure is computable iff the set of basic open sets containing it is effectively enumerable.

Integrable tests

To define what it means for a point $x \in 2^{\omega}$ to be μ-**MLR**, we extend the notion of integrable test.

Algorithmic randomness for different measures

Definition (Uniform tests)

A **uniform integrable test** is a lower semi-computable function $t : 2^\omega \times \mathcal{M}(2^\omega) \to \overline{\mathbb{R}}$ such that : $\int t(x, \mu)\, \mathrm{d}\mu(x)$ is finite for all μ

Theorem (Levin-Gács-Hoyrup-Rojás)

There exists a universal uniform integrable test $\mathbf{u}$ which dominates every other integrable tests up to a multiplicative constant.

Randomness

Intuitively $\mathbf{u}(x, \mu)$ can represent the randomness deficiency of x with respect to the measure μ. We say that x is Martin-löf random for the measure μ iff $\mathbf{u}(x, \mu) < +\infty$. The notion matches the previous one for computable measures.

Algorithmic randomness for different measures

Definition (Uniform tests)

A **uniform integrable test** is a lower semi-computable function $t : 2^\omega \times \mathcal{M}(2^\omega) \to \overline{\mathbb{R}}$ such that : $\int t(x, \mu)\, \mathrm{d}\mu(x)$ is finite for all μ

Theorem (Levin-Gács-Hoyrup-Rojás)

There exists a universal uniform integrable test $\mathbf{u}$ which dominates every other integrable tests up to a multiplicative constant.

Randomness

Intuitively $\mathbf{u}(x, \mu)$ can represent the randomness deficiency of x with respect to the measure μ. We say that x is Martin-löf random for the measure μ iff $\mathbf{u}(x, \mu) < +\infty$. The notion matches the previous one for computable measures.

Algorithmic randomness for different measures

Definition (Uniform tests)

A **uniform integrable test** is a lower semi-computable function $t : 2^\omega \times \mathcal{M}(2^\omega) \to \overline{\mathbb{R}}$ such that : $\int t(x, \mu)\, \mathrm{d}\mu(x)$ is finite for all μ

Theorem (Levin-Gács-Hoyrup-Rojás)

There exists a universal uniform integrable test $\mathbf{u}$ which dominates every other integrable tests up to a multiplicative constant.

Randomness

Intuitively $\mathbf{u}(x, \mu)$ can represent the randomness deficiency of x with respect to the measure μ. We say that x is Martin-löf random for the measure μ iff $\mathbf{u}(x, \mu) < +\infty$. The notion matches the previous one for computable measures.

Randomness extraction

Section 3

Randomness extraction

Reformulation of the problem

Reformulation of the problem

Given a class $\mathcal{C} \subseteq \mathcal{M}(2^\omega)$, is there a computable function $f : 2^\omega \to 2^\omega$ such that :

For all x such that $\mathbf{u}(x, \mu) < \infty$ for *some* $\mu \in \mathcal{C}$,
$f(x)$ is a binary sequence random for the uniform measure ?

A first piece of the puzzle

Randomness can be extracted when the measure is known.

The Levin-Kautz conversion procedure

Theorem (Levin-Kautz)

If μ is a computable measure on 2^ω, then there is a computable $f : 2^\omega \to 2^\omega$ such that $f(x)$ is random for all μ-random x which **are not atoms of** μ (x is an atom of μ if $\mu(\{x\}) > 0$, which implies that x is computable).

It is not hard to see that Levin-Kautz theorem is uniform :

Theorem (Levin-Kautz, extended)

There is a computable $f : 2^\omega \times \mathcal{M}(2^\omega) \to 2^\omega$ such that $f(x, \mu)$ is random whenever x is μ-random without being an atom of μ.

The Levin-Kautz conversion procedure

Theorem (Levin-Kautz)

If μ is a computable measure on 2^ω, then there is a computable $f : 2^\omega \to 2^\omega$ such that $f(x)$ is random for all μ-random x which **are not atoms of** μ (x is an atom of μ if $\mu(\{x\}) > 0$, which implies that x is computable).

It is not hard to see that Levin-Kautz theorem is uniform :

Theorem (Levin-Kautz, extended)

There is a computable $f : 2^\omega \times \mathcal{M}(2^\omega) \to 2^\omega$ such that $f(x, \mu)$ is random whenever x is μ-random without being an atom of μ.

Guessing the measure

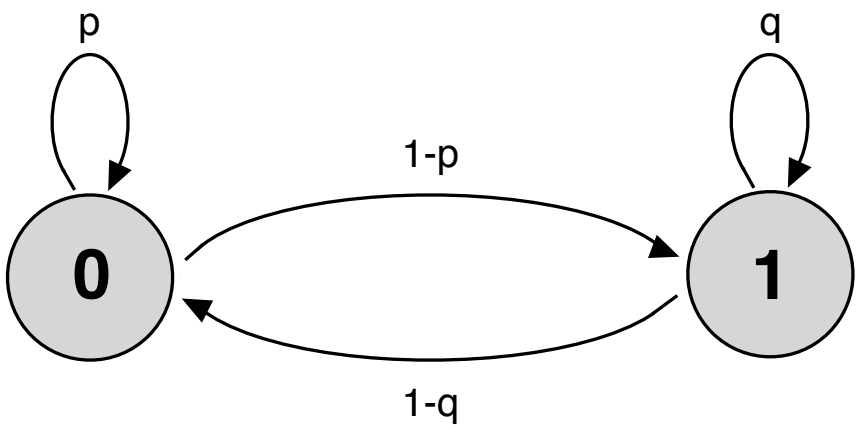
Guessing the measure

Suppose a measure μ was such that it could be *guessed*, *in some uniform way* from any of its random (non-atomic) elements. Then randomness extraction for such measures would be possible on that particular μ.

Guessing the measure

Example

Suppose μ is represented with a Markov chain of type

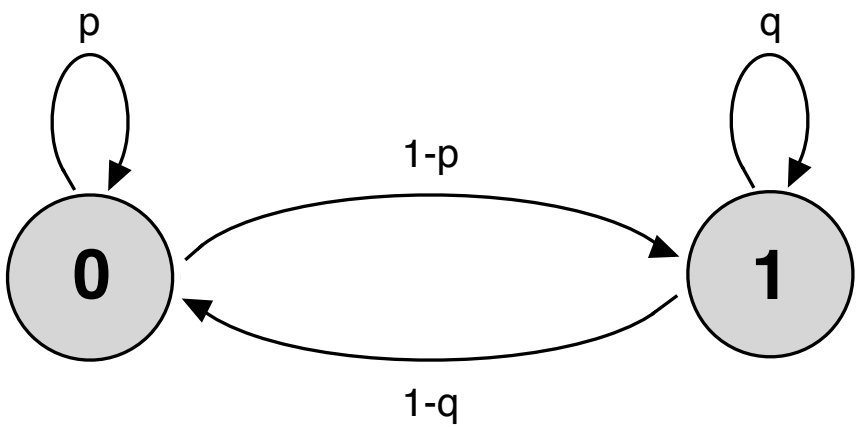


And we get a random $x = 00000000100000001100000000000\ldots$.
Can we deduce anything about p and q after reading finitely many bits? **No! Maybe p is small and only the beginning of the sequence is atypical.**

Guessing the measure

Example

Suppose μ is represented with a Markov chain of type



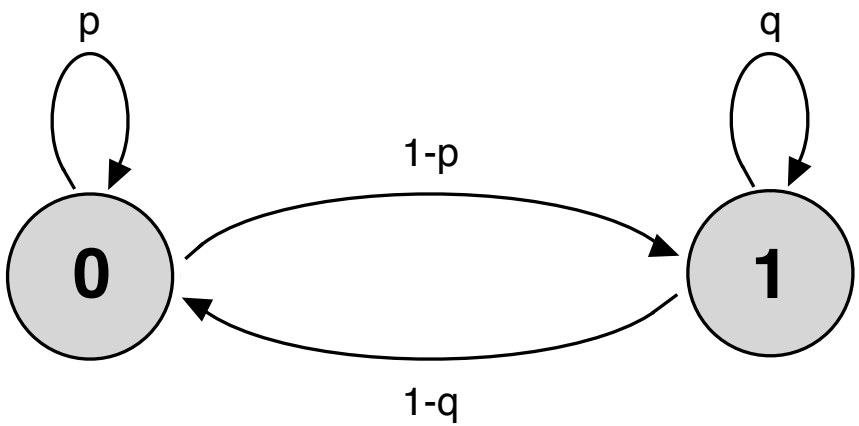
And we get a random $x = 00000000100000001100000000000\ldots$.

Can we deduce anything about p and q after reading finitely many bits? **No! Maybe p is small and only the beginning of the sequence is atypical.**

Guessing the measure

Example

Suppose μ is represented with a Markov chain of type

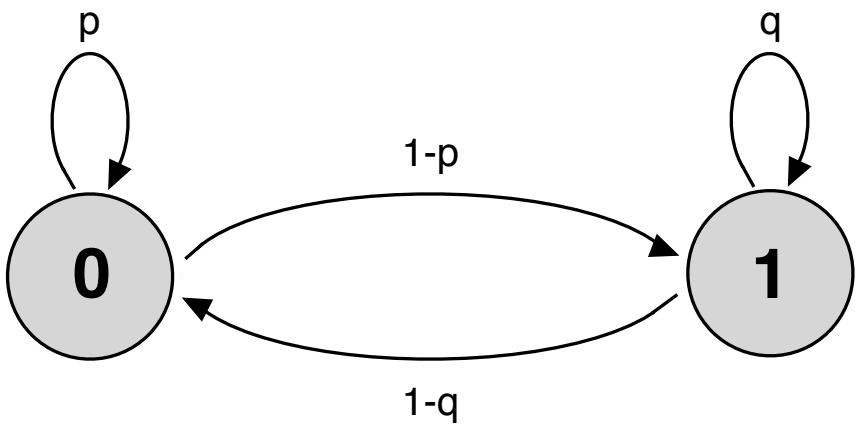


And we get a random $x = 000000001000000011000000000\ldots$.
Can we deduce anything about p and q after reading finitely many bits ? **No ! Maybe p is small and only the beginning of the sequence is atypical.**

Guessing the measure

Example

Suppose μ is represented with a Markov chain of type



And we get a random $x = 00000000100000001100000000000\ldots$.
Can we deduce anything about p and q after reading finitely many bits ? **No ! Maybe p is small and only the beginning of the sequence is atypical.**

Layerwiseness

However...

We could compute p and q if we knew a bound on the randomness deficiency of x with respect to μ !

Layerwise computability (Hoyrup and Rojás)

A function F is μ-layerwise computable over a space x if it is defined on all μ-random reals and it can be uniformly computed modulo an "advice" which is an upper bound on the randomness deficiency $\mathbf{u}(x, \mu)$.

Definition (Bienvenu-Monin)

A measure μ is **(layerwise) learnable** if it can be layerwise computed from its random elements.

Layerwiseness

However...

We could compute p and q if we knew a bound on the randomness deficiency of x with respect to μ !

Layerwise computability (Hoyrup and Rojás)

A function F is μ-layerwise computable over a space x if it is defined on all μ-random reals and it can be uniformly computed modulo an "advice" which is an upper bound on the randomness deficiency $\mathbf{u}(x, \mu)$.

Definition (Bienvenu-Monin)

A measure μ is **(layerwise) learnable** if it can be layerwise computed from its random elements.

Layerwiseness

However...

We could compute p and q if we knew a bound on the randomness deficiency of x with respect to μ !

Layerwise computability (Hoyrup and Rojás)

A function F is μ-layerwise computable over a space x if it is defined on all μ-random reals and it can be uniformly computed modulo an "advice" which is an upper bound on the randomness deficiency $\mathbf{u}(x, \mu)$.

Definition (Bienvenu-Monin)

A measure μ is **(layerwise) learnable** if it can be layerwise computed from its random elements.

A criterion

A criterion for learnability :

Theorem (Bienvenu-Monin)

If a measure μ belongs to a class $\mathcal{C}$ of measures such that

(i) $\mathcal{C}$ is Π^0_1

(ii) no distinct ν_1, ν_2 have a random in common $(\star)$

then μ is learnable.

Surprisingly, the converse holds :

Theorem (Bienvenu-Monin)

If a measure μ is learnable, then it can be embedded into a Π^0_1 class of measures with the $(\star)$ property.

A criterion

A criterion for learnability :

Theorem (Bienvenu-Monin)

If a measure μ belongs to a class $\mathcal{C}$ of measures such that

(i) $\mathcal{C}$ is Π^0_1

(ii) no distinct ν_1, ν_2 have a random in common $(\star)$

then μ is learnable.

Surprisingly, the converse holds :

Theorem (Bienvenu-Monin)

If a measure μ is learnable, then it can be embedded into a Π^0_1 class of measures with the $(\star)$ property.

Putting things together

We are ready to present a partial answer to the original question.

Theorem (Bienvenu-Monin)

Let $\mathcal{C}$ be a Π^0_1 class of measures with the $(\star)$ property. Then uniform randomness extraction is possible, i.e., there exists a partial computable function $f : 2^\omega \to 2^\omega$ such that :

if x is μ-random for some $\mu \in \mathcal{C}$ and x is not an atom of μ, then $f(x)$ is random for the uniform measure.

(this even extends to Σ^0_2 classes with the $(\star)$ property).

Putting things together

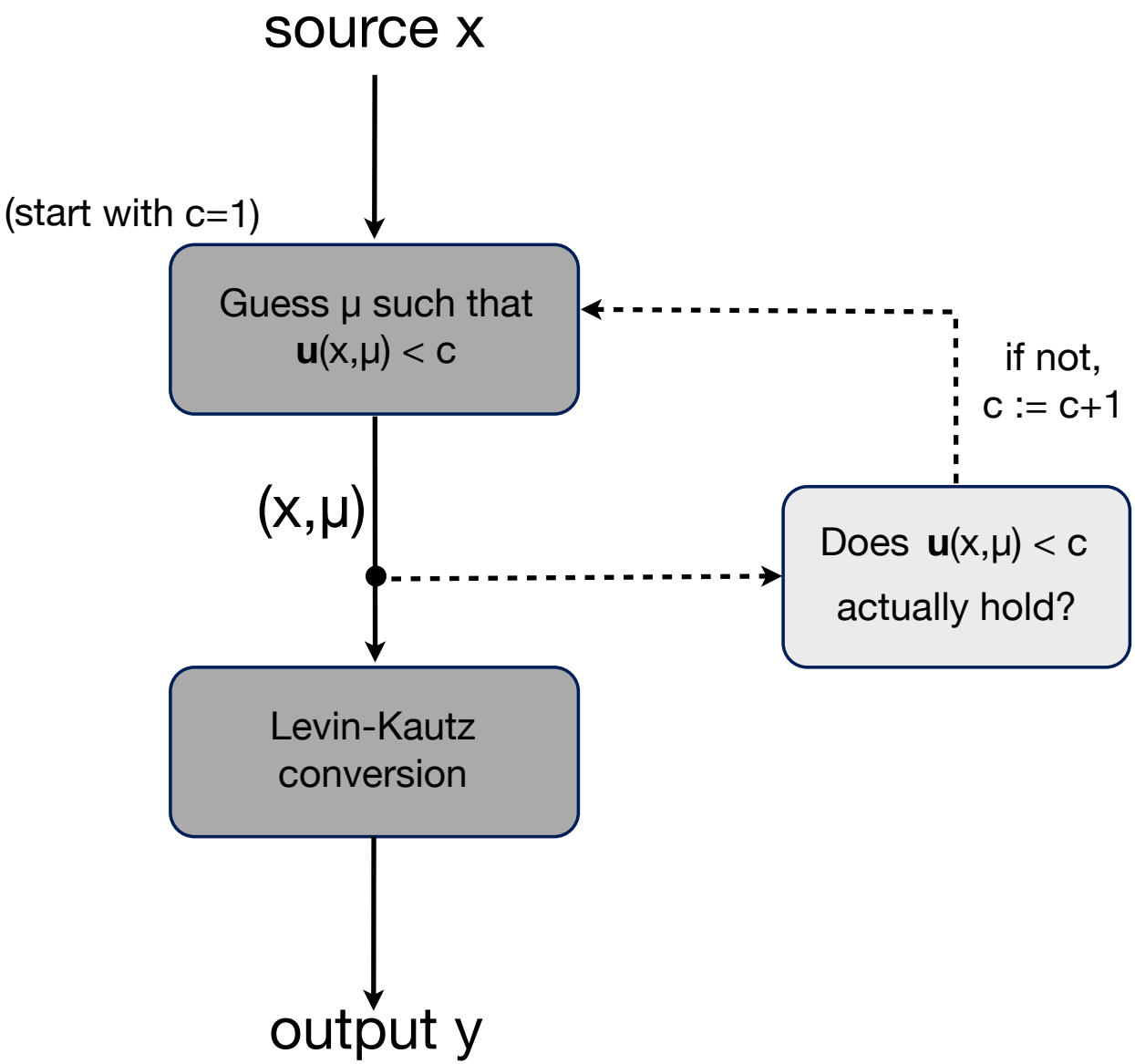
We are ready to present a partial answer to the original question.

Theorem (Bienvenu-Monin)

Let $\mathcal{C}$ be a Π^0_1 class of measures with the $(\star)$ property. Then uniform randomness extraction is possible, i.e., there exists a partial computable function $f : 2^\omega \to 2^\omega$ such that :

if x is μ-random for some $\mu \in \mathcal{C}$ and x is not an atom of μ, then $f(x)$ is random for the uniform measure.

(this even extends to Σ^0_2 classes with the $(\star)$ property).


source x
(start with c=1)
Guess μ such that
u(x,μ) < c
if not,
c := c+1
(x,μ)
Does **u**(x,μ) < c
actually hold?
Levin-Kautz
conversion
output y

Thank you. Questions ?