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.
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: \( P(\text{head}) = p \) and \( P(\text{tail}) = 1 - p \)
Von Neumann’s coin trick example

Example

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

The first results: 110111100101101101111100
Von Neumann’s coin trick example

Example

The biased coin: $P(\text{head}) = p$ and $P(\text{tail}) = 1 - p$

The first results: 110111100101101101111100

The trick:

\[
\begin{array}{cccccccccccc}
\underline{11} & \underline{01} & \underline{11} & \underline{10} & \underline{01} & \underline{01} & \underline{10} & \underline{11} & \underline{01} & \underline{11} & \underline{11} & \underline{00} \\
- & 0 & - & 1 & 0 & 0 & 1 & - & 0 & - & - & -
\end{array}
\]
**Example**

The biased coin: \( P(\text{head}) = p \) and \( P(\text{tail}) = 1 - p \)

The first results: 110111100101101101111100

The trick:

\[
\begin{align*}
11 & \quad 01 & \quad 11 & \quad 10 & \quad 01 & \quad 01 & \quad 10 & \quad 11 & \quad 01 & \quad 11 & \quad 11 & \quad 00 \\
p^2 & \quad p(1-p) & \quad p^2 & \quad p(1-p) & \quad p(1-p) & \quad p(1-p) & \quad p^2 & \quad p(1-p) & \quad p^2 & \quad p^2 & \quad (1-p)^2 \\
- & \quad 0 & \quad - & \quad 1 & \quad 0 & \quad 0 & \quad 1 & \quad - & \quad 0 & \quad - & \quad - & \quad -
\end{align*}
\]

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

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

- We have access to a random sequence for a given measure $\mu$ which we do not know.
- However, we do know that $\mu$ 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:

What does it mean for a string to be random?

Are

c :0000000000000001000000000001000000000000100000000000001...

or

\( \pi :0010010000111111011010100100010001000101101001100 \ldots \)

random?
Algorithmic randomness

Algorithmic randomness:

What does it mean for a string to be random?

Intuition

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

Definition (Martin-Löf)

A sequence of $2^\omega$ is Martin-Löf random if it belongs to the smallest $\Sigma^0_2$ set, effectively of measure 1.
Algorithmic randomness:

What does it mean for a string to be random?

**Intuition**
A sequence of $2^\omega$ should be random if it belongs to the smallest set of measure 1.

**Definition (Martin-Löf)**
A sequence of $2^\omega$ is **Martin-Löf random** if it belongs to the smallest $\Sigma^0_2$ set, effectively of measure 1.
Algorithmic randomness

Definition (Martin-Löf test)

A $\Pi^0_2$ subset of $2^\omega$ if 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 \( \Pi^0_2 \) subset of \( 2^\omega \) if 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 :

<table>
<thead>
<tr>
<th>Measure</th>
<th>f(., 1)</th>
<th>f(., 2)</th>
<th>f(., 3)</th>
<th>f(., 4)</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda(f(1, \mathbb{N})) \leq \frac{1}{2}$</td>
<td>$s_{1,1}$</td>
<td>$s_{1,2}$</td>
<td>$s_{1,3}$</td>
<td>$s_{1,4}$</td>
<td>...</td>
</tr>
<tr>
<td>$\lambda(f(2, \mathbb{N})) \leq \frac{1}{4}$</td>
<td>$s_{2,1}$</td>
<td>$s_{2,2}$</td>
<td>$s_{2,3}$</td>
<td>$s_{2,4}$</td>
<td>...</td>
</tr>
<tr>
<td>$\lambda(f(3, \mathbb{N})) \leq \frac{1}{8}$</td>
<td>$s_{3,1}$</td>
<td>$s_{3,2}$</td>
<td>$s_{3,3}$</td>
<td>$s_{3,4}$</td>
<td>...</td>
</tr>
<tr>
<td>$\lambda(f(4, \mathbb{N})) \leq \frac{1}{16}$</td>
<td>$s_{4,1}$</td>
<td>$s_{4,2}$</td>
<td>$s_{4,3}$</td>
<td>$s_{4,4}$</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Algorithmic randomness : universal Martin-Löf test

Universal test :

<table>
<thead>
<tr>
<th>Measure</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{1,i}^n \right) \leq \frac{1}{2}$</td>
<td>$(s_{1,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{2,i}^n \right) \leq \frac{1}{4}$</td>
<td>$(s_{2,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{3,i}^n \right) \leq \frac{1}{8}$</td>
<td>$(s_{3,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{4,i}^n \right) \leq \frac{1}{16}$</td>
<td>$(s_{4,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Algorithmic randomness: universal Martin-Löf test

Universal test:

<table>
<thead>
<tr>
<th>Measure</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{1,i}^n \right) \leq \frac{1}{2}$</td>
<td>$(s_{1,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{2,i}^n \right) \leq \frac{1}{4}$</td>
<td>$(s_{2,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{3,i}^n \right) \leq \frac{1}{8}$</td>
<td>$(s_{3,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{4,i}^n \right) \leq \frac{1}{16}$</td>
<td>$(s_{4,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
</tbody>
</table>
Algorithmic randomness: universal Martin-Löf test

Universal test:

<table>
<thead>
<tr>
<th>Measure</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s^n_{1,i} \right) \leq \frac{1}{2} )</td>
<td>((s^1_{1,i})_{i \in \mathbb{N}})</td>
<td>((s^2_{1,i})_{i \in \mathbb{N}})</td>
<td>((s^3_{1,i})_{i \in \mathbb{N}})</td>
<td>((s^4_{1,i})_{i \in \mathbb{N}})</td>
<td>...</td>
</tr>
<tr>
<td>( \forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s^n_{2,i} \right) \leq \frac{1}{4} )</td>
<td>((s^1_{2,i})_{i \in \mathbb{N}})</td>
<td>((s^2_{2,i})_{i \in \mathbb{N}})</td>
<td>((s^3_{2,i})_{i \in \mathbb{N}})</td>
<td>((s^4_{2,i})_{i \in \mathbb{N}})</td>
<td>...</td>
</tr>
<tr>
<td>( \forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s^n_{3,i} \right) \leq \frac{1}{8} )</td>
<td>((s^1_{3,i})_{i \in \mathbb{N}})</td>
<td>((s^2_{3,i})_{i \in \mathbb{N}})</td>
<td>((s^3_{3,i})_{i \in \mathbb{N}})</td>
<td>((s^4_{3,i})_{i \in \mathbb{N}})</td>
<td>...</td>
</tr>
<tr>
<td>( \forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s^n_{4,i} \right) \leq \frac{1}{16} )</td>
<td>((s^1_{4,i})_{i \in \mathbb{N}})</td>
<td>((s^2_{4,i})_{i \in \mathbb{N}})</td>
<td>((s^3_{4,i})_{i \in \mathbb{N}})</td>
<td>((s^4_{4,i})_{i \in \mathbb{N}})</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Algorithmic randomness: universal Martin-Löf test

Universal test:

<table>
<thead>
<tr>
<th>Measure</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{1,i}^n \right) \leq \frac{1}{2}$</td>
<td>$(s_{1,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{2,i}^n \right) \leq \frac{1}{4}$</td>
<td>$(s_{2,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{3,i}^n \right) \leq \frac{1}{8}$</td>
<td>$(s_{3,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{4,i}^n \right) \leq \frac{1}{16}$</td>
<td>$(s_{4,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Algorithmic randomness: universal Martin-Löf test

Universal test:

<table>
<thead>
<tr>
<th>Measure</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{1,i}^n \right) \leq \frac{1}{2}$</td>
<td>$(s_{1,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{2,i}^n \right) \leq \frac{1}{4}$</td>
<td>$(s_{2,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{3,i}^n \right) \leq \frac{1}{8}$</td>
<td>$(s_{3,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{4,i}^n \right) \leq \frac{1}{16}$</td>
<td>$(s_{4,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{4,i}^4)_{i \in \mathbb{N}}$</td>
<td>...</td>
</tr>
</tbody>
</table>
Algorithmic randomness: universal Martin-Löf test

Universal test:

<table>
<thead>
<tr>
<th>Measure</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{1,i}^n \right) \leq \frac{1}{2}$</td>
<td>$(s_{1,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{1,i}^4)_{i \in \mathbb{N}}$</td>
<td>…</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{2,i}^n \right) \leq \frac{1}{4}$</td>
<td>$(s_{2,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{2,i}^4)_{i \in \mathbb{N}}$</td>
<td>…</td>
</tr>
<tr>
<td>$\forall n \lambda \left( \bigcup_{i \in \mathbb{N}} s_{3,i}^n \right) \leq \frac{1}{8}$</td>
<td>$(s_{3,i}^1)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^2)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^3)_{i \in \mathbb{N}}$</td>
<td>$(s_{3,i}^4)_{i \in \mathbb{N}}$</td>
<td>…</td>
</tr>
</tbody>
</table>

$\exists m \ x \in \bigcap_{n} \bigcup_{i} s_{n,i}^m \leftrightarrow x \in \bigcap_{k} \bigcup_{n,i} s_{n+k,i}^n$
Algorithmic randomness for other measures

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 $\mu$ be a computable measure. A sequence of $2^\omega$ is **Martin-Löf random for the measure** $\mu$ if it belongs to the smallest $\Sigma^0_2$ set, effectively of $\mu$ measure 1.
Algorithmic randomness for other measures

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 $\mu$ be a computable measure. A sequence of $2^\omega$ is **Martin-Löf random for the measure $\mu$** if it belongs to the smallest $\Sigma^0_2$ set, effectively of $\mu$ measure 1.
Algorithmic randomness for other measures

For non-computable measure

- When the measure is not computable, we cannot necessarily obtain universal Martin-Löf test for the measure...
- 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.
- A good solution can be to adopt an analytical point of view.
Algorithmic randomness for other measures

For non-computable measure

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

- 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.

- A good solution can be to adopt an analytical point of view.
Algorithmic randomness for other measures

**For non-computable measure**

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

- 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.

- A good solution can be to adopt an analytical point of view.
Constructive topological spaces

Definition (constructive topological space)

The notion of **constructive topological space** can be used to generalize the well known computability notions in $2^\omega$ or $\mathbb{R}$. Formally it is given by $(X, \mathcal{O}(X), \nu)$ where:

- $X$ is a $T_0$ second countable topological space.
- $\mathcal{O}(X)$ is a countable basis for the topology and $\nu : \mathbb{N} \to \mathcal{O}(X)$ is an enumeration of basic open sets.

The $T_0$ separation axiom means that each point of the space is uniquely determined by basic open sets containing it.
Définition (Computable functions)

Let $X$ and $Y$ be two constructive topological spaces. A function $f : X \to Y$ is said to be **computable** if for all $x$, one can uniformly obtain from any enumeration of basic open sets containing $x$, an effective enumeration of basic open sets containing $f(x)$. 
Constructive topological spaces

\[ x \in O_i \]

\[ \begin{array}{cccccccc}
O_0 & O_1 & O_2 & O_3 & O_4 & O_5 & O_6 & O_7 & O_8 & \ldots \\
\end{array} \]

\[ f \]

\[ f(x) \in U_i \]

\[ \begin{array}{cccccccc}
U_0 & U_1 & U_2 & U_3 & U_4 & U_5 & U_6 & U_7 & U_8 & \ldots \\
\end{array} \]
Computable function between constructive topological spaces

Fact
A function $f : X \to Y$ is computable iff for all basic open sets $U \subseteq Y$, $f^{-1}(U)$ is an effectively open set of $X$, uniformly in $U$.

It makes computability an effective version of continuity.
Lower semi-computable functions

Definition (lower semi-computable function)

Let $(X, \mathcal{O}(X), \nu)$ be a constructive topological space. A function $f : X \to \overline{\mathbb{R}}^+$ is said to be lower semi-computable if equivalently:

- $f$ is the sum of an effective sequence of computable functions.
- $f^{-1}(]r; +\infty[)$ is an effectively open set.

It makes lower semi-computability an effective version of lower semi-continuity.
Fact

The space of measures on $2^\omega$ denoted by $\mathcal{M}(2^\omega)$ can be structured as a constructive metric space.

A basic open set in the space of measure:

$\mathcal{M}(2^\omega) \subseteq [0, 1]^\mathbb{N}$
Algorithmic randomness for different measures

Fact

The space of measures on $2^\omega$ can be structured as a constructive metric space.

- 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 product topology on $[0, 1]^\mathbb{N}$.
- A measure is computable iff the set of basic open sets containing it is effectively enumerable.
Algorithmic randomness for different measures

Integrable tests

To define what it means for a point $x \in 2^\omega$ to be $\mu$-MLR, we define the notion of uniform integrable test:

**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:

$$\int t(x, \mu) \ d\mu(x) \leq 1 \ (\text{for all } \mu)$$

**Definition**

We say that $x$ passes an integrable test with the measure $\mu$ if $t(x, \mu) < +\infty$. 
Algorithmic randomness for different measures

### Integrable tests

To define what it means for a point $x \in 2^\omega$ to be $\mu$-MLR, we define the notion of uniform integrable test:

**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:

$$\int t(x, \mu) \, d\mu(x) \leq 1 \quad (\text{for all } \mu)$$

**Definition**

We say that $x$ passes an integrable test with the measure $\mu$ if $t(x, \mu) < +\infty$. 
Algorithmic randomness for different measures

Integrable tests

To define what it means for a point $x \in 2^\omega$ to be $\mu$-MLR, we define the notion of uniform integrable test:

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:

$$\int t(x, \mu) \, d\mu(x) \leq 1 \text{ (for all } \mu)$$

Definition

We say that $x$ passes an integrable test with the measure $\mu$ if $t(x, \mu) < +\infty$. 
Algorithmic randomness for different measures

Integrable tests

As it turns out, there exists a universal uniform integrable test.

Theorem (Levin-Gács-Hoyrup-Rojás)

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

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

Integrable tests

As it turns out, there exists a universal uniform integrable test.

Theorem (Levin-Gács-Hoyrup-Rojás)

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

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

Integrable tests

As it turns out, there exists a universal uniform integrable test.

Theorem (Levin-Gács-Hoyrup-Rojás)

There exists a universal uniform integrable test \( u \) which dominates every other integrable tests up to a multiplicative constant.

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

In fact, being computable in a measure $\mu$ (with the analytical point of view) is equivalent to being computable from all binary oracles representing the measure.

Theorem (Day, Miller)

$u(x, \mu) < +\infty$ iff there exists an oracle representation $A$ of $\mu$ such that $x$ is in the smallest $A-\Sigma^0_2$ set $A$-effectively of measure 1.
Algorithmic randomness for different measures

In fact, being computable in a measure $\mu$ (with the analytical point of view) is equivalent to being computable from all binary oracles representing the measure.

Theorem (Day, Miller)

$u(x, \mu) < +\infty$ iff there exists an oracle representation $A$ of $\mu$ such that $x$ is in the smallest $A-\Sigma^0_2$ set $A$-effectively of measure 1.
Randomness extraction

Section 3

Randomness extraction
Reformulation of the problem

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

For all $x$ such that $u(x, \mu) < \infty$ 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 $\mu$ is a computable measure on $2^\omega$, then there is a computable $f : 2^\omega \to 2^\omega$ such that $f(x)$ is random for all $\mu$-random $x$ which are not atoms of $\mu$ ($x$ is an atom of $\mu$ 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 $\mu$-random without being an atom of $\mu$. 
The Levin-Kautz conversion procedure

**Theorem (Levin-Kautz)**

If $\mu$ is a computable measure on $2^\omega$, then there is a computable $f : 2^\omega \to 2^\omega$ such that $f(x)$ is random for all $\mu$-random $x$ which are not atoms of $\mu$ ($x$ is an atom of $\mu$ 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 $\mu$-random without being an atom of $\mu$. 
Suppose a measure $\mu$ 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 $\mu$. 
Guessing the measure

Example

Suppose $\mu$ is represented with a Markov chain of type

\[
\begin{array}{c}
\text{0} \\
\text{p} \quad \text{1-p} \\
\text{1-q} \\
\text{1} \\
\text{q}
\end{array}
\]

And we get a random $x = 0000000100000011000000000000\ldots$. 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 $\mu$ is represented with a Markov chain of type

\[
\begin{align*}
0 & \xrightarrow{p} 0 \\
0 & \xrightarrow{1-p} 1 \\
1 & \xrightarrow{q} 1 \\
1 & \xrightarrow{1-q} 0
\end{align*}
\]

And we get a random $x = 0000000100000001100000000000 \ldots$.

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 $\mu$ is represented with a Markov chain of type

And we get a random $x = 00000000100000001100000000000000\ldots$.
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 $\mu$ is represented with a Markov chain of type

And we get a random $x = 000000001000000001100000000000000000000000000...$. 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 $\mu$!

This is precisely the idea of layerwise computability of Hoyrup and Rojás: a function $F$ is $\mu$-layerwise computable over a space $X$ if it is defined on all $\mu$-random reals and it can be uniformly computed modulo an "advice" which is an upper bound on the randomness deficiency $u(x, \mu)$.

Definition (Bienvenu-Monin)

A measure $\mu$ 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 $\mu$!

This is precisely the idea of *layerwise computability* of Hoyrup and Rojás: a function $F$ is $\mu$-layerwise computable over a space $X$ if it is defined on all $\mu$-random reals and it can be uniformly computed modulo an "advice" which is an upper bound on the randomness deficiency $u(x, \mu)$.

Definition (Bienvenu-Monin)

A measure $\mu$ is *(layerwise) learnable* if it can be layerwise computed from its random elements.
However...

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

This is precisely the idea of *layerwise computability* of Hoyrup and Rojás: a function $F$ is $\mu$-layerwise computable over a space $X$ if it is defined on all $\mu$-random reals and it can be uniformly computed modulo an "advice" which is an upper bound on the randomness deficiency $u(x, \mu)$.

**Definition (Bienvenu-Monin)**

A measure $\mu$ is *(layerwise) learnable* if it can be layerwise computed from its random elements.
A criterion for learnability:

Theorem (Bienvenu-Monin)

If a measure $\mu$ belongs to a class $C$ of measures such that

(i) $C$ is $\Pi^0_1$

(ii) no distinct $\nu_1, \nu_2$ have a random in common $(\star)$

then $\mu$ is learnable.

Surprisingly, the converse holds:

Theorem (Bienvenu-Monin)

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

A criterion for learnability:

Theorem (Bienvenu-Monin)

If a measure $\mu$ belongs to a class $C$ of measures such that

(i) $C$ is $\Pi_1^0$

(ii) no distinct $\nu_1, \nu_2$ have a random in common (★)

then $\mu$ is learnable.

Surprisingly, the converse holds:

Theorem (Bienvenu-Monin)

If a measure $\mu$ is learnable, then it can be embedded into a $\Pi_1^0$ class of measures with the (★) property.
Putting things together

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

Theorem (Bienvenu-Monin)

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

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

(this even extends to $\Sigma^0_2$ classes with the (★) property).
Putting things together

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

**Theorem (Bienvenu-Monin)**

Let $\mathcal{C}$ be a $\Pi^0_1$ class of measures with the (⋆) 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 $\mu$-random for some $\mu \in \mathcal{C}$ and $x$ is not an atom of $\mu$, then $f(x)$ is random for the uniform measure.

(this even extends to $\Sigma^0_2$ classes with the (⋆) property).
source x

(start with $c=1$)

Guess $\mu$ such that $u(x,\mu) < c$

$(x,\mu)$

Does $u(x,\mu) < c$ actually hold?

if not, $c := c+1$

Levin-Kautz conversion

output $y$
Thank you. Questions?