Computability, Randomness and Ergodic Theory on Metric Spaces

(Calculabilité, aléatoire et théorie ergodique sur les espaces métriques)

Mathieu Hoyrup

ENS

June 17, 2008

What is randomness?

What is randomness?

1. What does randomness look like?

What is randomness?

1. What does randomness look like?
2. How does randomness appear?

What is randomness?

1. What does randomness look like?
2. How does randomness appear?

1. Probability theory

What properties should random sequences satisfy?

What is randomness?

1. What does randomness look like?
2. How does randomness appear?

1. Probability theory

What properties should random sequences satisfy?

Strong law of large numbers

In random sequences, number of 0's $=$ number of 1's.

What is randomness?

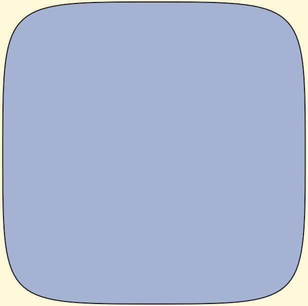
1. What does randomness look like?
2. How does randomness appear?

2. Ergodic theory

In deterministic dynamical systems, as unpredictability.

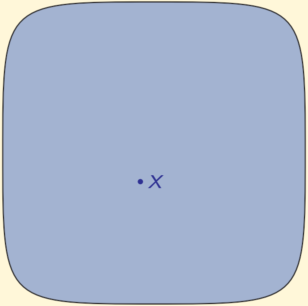
Deterministic dynamical system

- Space X,
- Transformation $T : X \to X$.



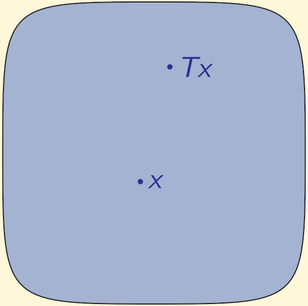
Deterministic dynamical system

- Space X,
- Transformation $T : X \to X$,



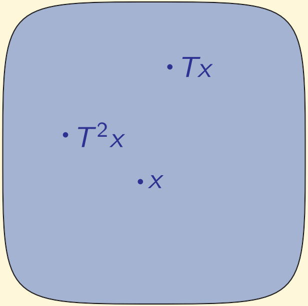
Deterministic dynamical system

- Space X,
- Transformation $T : X \to X$,



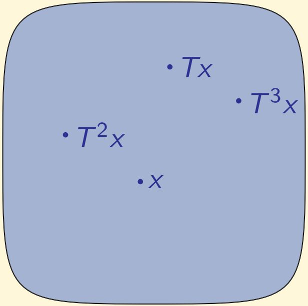
Deterministic dynamical system

- Space X,
- Transformation $T : X \to X$,



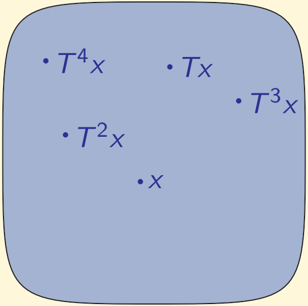
Deterministic dynamical system

- Space X,
- Transformation $T : X \to X$,



Deterministic dynamical system

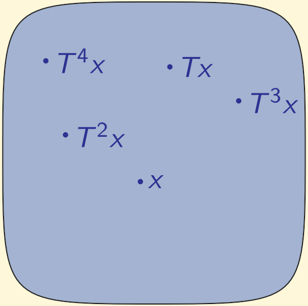
- Space X,
- Transformation $T : X \to X$,



Deterministic dynamical system

...observed with sharp eyes

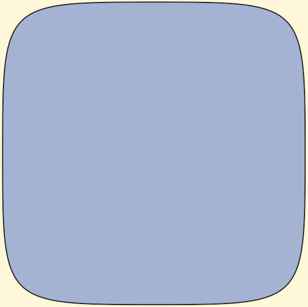
- Space X,
- Transformation $T : X \to X$,
- Laplace's demon.



Deterministic dynamical system

...observed with finite precision

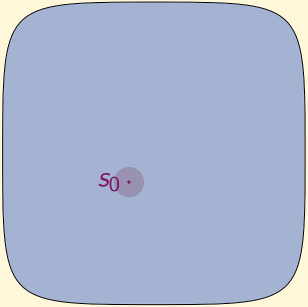
- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.



Deterministic dynamical system

...observed with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.

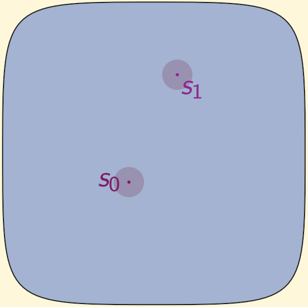


s_0

Deterministic dynamical system

...observed with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.

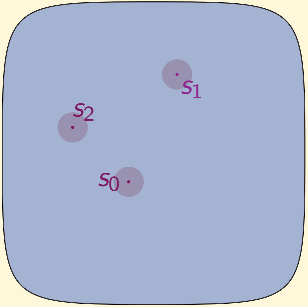


s_0, s_1

Deterministic dynamical system

...observed with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.

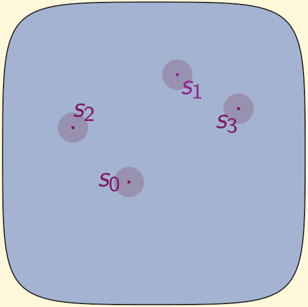


s_0, s_1, s_2

Deterministic dynamical system

...observed with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.

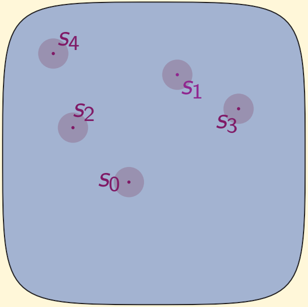


s_0, s_1, s_2, s_3

Deterministic dynamical system

...observed with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.



$s_0, s_1, s_2, s_3, s_4, \ldots$

- Space X,
- Transformation $T : X \to X$,
- Invariant measure μ.

What is randomness?

1. What does randomness look like?

1. Probability theory

What properties should random sequences satisfy?

Algorithmic randomness (Martin-Löf, 1966)

$$\{0,1\}^{\mathbb{N}} = R_\mu \uplus N_\mu$$

$$0000000000000000000000\ldots$$

$$0011100111001110011001\ldots$$

$$0101000101101011100110\ldots$$

What is randomness?

1. What does randomness look like?

1. Probability theory

What properties should random sequences satisfy?

Algorithmic randomness (Martin-Löf, 1966)

$$\{0,1\}^{\mathbb{N}} = R_\mu \uplus N_\mu$$

$0000000000000000000000\ldots$ non-random

$0011100111001110011100l\ldots$

$0101000101101101110011 0\ldots$

What is randomness?

1. What does randomness look like?

1. Probability theory

What properties should random sequences satisfy?

Algorithmic randomness (Martin-Löf, 1966)

$$\{0,1\}^{\mathbb{N}} = R_\mu \uplus N_\mu$$

$000000000000000000000000\dots$ non-random

$001110011100111001110011\dots$ non-random

$010100010110110111001101\dots$

What is randomness?

1. What does randomness look like?

1. Probability theory

What properties should random sequences satisfy?

Algorithmic randomness (Martin-Löf, 1966)

$$\{0,1\}^{\mathbb{N}} = R_\mu \uplus N_\mu$$

$000000000000000000000\ldots$ non-random

$001110011100111001110\ldots$ non-random

$010100010110110111001\ldots$ random

What is randomness?

1. What does randomness look like?
2. How does randomness appear?

2. Ergodic theory

In deterministic dynamical systems, as unpredictability.

Algorithmic complexity of orbits (Kolmogorov, 1965 – Brudno, 1978)

"A system is unpredictable

$$\Longleftrightarrow$$

its orbits are algorithmically unpredictable"

Contributions

Computability, Randomness and Ergodic Theory on Metric Spaces.

- Study of algorithmic randomness on general spaces,
- Development of algorithmic probability theory,
- Contributions to algorithmic complexity of orbits, relations with algorithmic randomness.

1. Computability/Semi-computability
2. Algorithmic randomness
   - Random sequences
   - Random points in metric spaces
3. Computability on probability spaces
   - Computability theory is topological
   - Definitions
   - Existence of almost decidable sets
4. Complexity of dynamical systems
   - Classical setting
   - Orbit complexity
   - Topological relations

Fast convergence: $q_i \twoheadrightarrow x$ means $d(q_i, x) < 2^{-i}$.

Computable real $x \in \mathbb{R}$

$$\boxed{\text{algorithm}} \xrightarrow{q_i \twoheadrightarrow x}$$

Computability/Semi-computability

on $\mathbb{R}$

Fast convergence: $q_i \twoheadrightarrow x$ means $d(q_i, x) < 2^{-i}$.

Computable real $x \in \mathbb{R}$

$$\boxed{\text{algorithm}} \xrightarrow{q_i \twoheadrightarrow x}$$

Computable function $f : \mathbb{R} \to \mathbb{R}$

$$\xrightarrow{q_i \twoheadrightarrow x} \boxed{\text{algorithm}} \xrightarrow{q'_i \twoheadrightarrow f(x)}$$

Examples

- Computable real numbers: $\sqrt{2}, \pi, e,$etc.
- Computable real functions: $\sqrt{x}, \cos, \ln,$etc.

Lower convergence: $q_i \nearrow x$ means $q_i \leq q_{i+1}, q_i \to x$.

Lower semi-computable real $x \in \mathbb{R}$

$$\boxed{\text{algorithm}} \xrightarrow{q_i \nearrow x}$$

Lower semi-computable function $f : \mathbb{R} \to \mathbb{R}$

$$\xrightarrow{q_i \twoheadrightarrow x} \boxed{\text{algorithm}} \xrightarrow{q'_i \nearrow f(x)}$$

Example

- Lower semi-computable real function: $\mathbf{1}_{(0,1)}$

Computability/Semi-computability

on $\mathbb{R}$

$$\begin{aligned}\mathbb{R}_c &= \{\text{computable real numbers}\}\\ \mathbb{R}_{sc} &= \{\text{semi-computable real numbers}\}\end{aligned}$$

Both $\mathbb{R}_c$ and $\mathbb{R}_{sc}$ are countable. But...

Computability

$\mathbb{R}_c$ is "effectively uncountable"

Semi-computability

$\mathbb{R}_{sc}$ is "effectively countable"

$q_i \twoheadrightarrow x$ means $d(q_i, x) < 2^{-i}$.

$q_i \nearrow x$ means $q_i \leq q_{i+1}, q_i \to x$.

$q_i \twoheadrightarrow x$ means $d(q_i, x) < 2^{-i}$.

$q_i \nearrow x$ means $q_i \leq q_{i+1}, q_i \to x$.

Computable metric space

to express computability

Enumerative lattice

to express semi-computability

Computability/Semi-computability

Abstract structures

$q_i \twoheadrightarrow x$ means $d(q_i, x) < 2^{-i}$.

$q_i \nearrow x$ means $q_i \leq q_{i+1}, q_i \to x$.

Computable metric space

to express computability

- $\mathbb{R}^n$, euclidean distance,
- $C([0,1])$,
  uniform distance $\|.\|_\infty$,
- Compact subsets of $\mathbb{R}$,
  Hausdorff distance,

Enumerative lattice

to express semi-computability

- $\overline{\mathbb{R}}$, order $\leq$,
- $\mathcal{P}(\mathbb{N})$, order $\subseteq$,

X computable metric space:

- τ_X, order $\subseteq$,
- $LC(X, \overline{\mathbb{R}})$[a], order $\leq$.

[a] called $\mathcal{C}(X, \overline{\mathbb{R}})$ in the thesis.

$q_i \twoheadrightarrow x$ means $d(q_i, x) < 2^{-i}$.

$q_i \nearrow x$ means $q_i \leq q_{i+1}, q_i \to x$.

Computable metric space

to express computability

- $\mathbb{R}^n$, euclidean distance,
- $C([0,1])$,
  uniform distance $\|.\|_\infty$,
- Compact subsets of $\mathbb{R}$,
  Hausdorff distance,
- $\mathcal{M}(X)$,
  Prokhorov distance.

Enumerative lattice

to express semi-computability

- $\overline{\mathbb{R}}$, order $\leq$,
- $\mathcal{P}(\mathbb{N})$, order $\subseteq$,

X computable metric space:

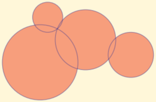
- τ_X, order $\subseteq$,
- $LC(X, \overline{\mathbb{R}})$[a], order $\leq$.

[a] called $\mathcal{C}(X, \overline{\mathbb{R}})$ in the thesis.

Theorem (2.1.4.1)

Let $\mu \in \mathcal{M}(X)$ be a probability measure.

μ *is computable* $\iff$ *all* $\mu(B_1 \cup \ldots \cup B_n)$ *are lower semi-computable.*



On $\mathbb{R}$

μ is computable $\iff$ all $\mu(q_1, q_2)$ are lower semi-computable.

On $\{0,1\}^{\mathbb{N}}$

μ is computable $\iff$ all $\mu([w])$ are computable ($w \in \{0,1\}^*$).

Computability/Semi-computability

Abstract structures

Computable metric space

to express computability

Enumerative lattice

to express semi-computability

The set of computable objects is "effectively uncountable" (in general)

The set of semi-computable objects is "effectively countable"

Algorithmically random sequences

$00000000000000000000000\ldots$ non-random

$00111001110011100111001\ldots$ non-random

$01010001011011011100110\ldots$ maybe random

Algorithmically random sequences

Martin-Löf, 1966

μ (computable) probablity measure on $\{0,1\}^{\mathbb{N}}$

Definition (Martin-Löf, 1966)

A sequence ω is μ-random if it withstands all μ-tests.

Algorithmically random sequences

μ (computable) probablity measure on $\{0,1\}^{\mathbb{N}}$

Definition (Martin-Löf, 1966)

A μ-test is a function $t : \{0,1\}^{\mathbb{N}} \to [0,+\infty]$ such that:

1. $\int t \, \mathrm{d}\mu < \infty$,
2. t is lower semi-computable.

Definition (Martin-Löf, 1966)

A sequence ω is μ-random if it withstands all μ-tests.

μ (computable) probablity measure on $\{0,1\}^{\mathbb{N}}$

Definition (Martin-Löf, 1966)

A μ-test is a function $t : \{0,1\}^{\mathbb{N}} \to [0,+\infty]$ such that:

1. $\int t \, \mathrm{d}\mu < \infty$,
2. t is lower semi-computable.

A sequence ω withstands the test if $t(\omega) < \infty$.

Definition (Martin-Löf, 1966)

A sequence ω is μ-random if it withstands all μ-tests.

Definition

A property P is testable if there is a μ-test t such that:

$$t(\omega) < \infty \implies P(\omega) \text{ holds.}$$

$$P(\omega) \text{ holds for } \mu\text{-almost every sequence } \omega$$

becomes

$$P(\omega) \text{ holds for every } \mu\text{-random sequence } \omega.$$

Examples

Strong law of large numbers, law of the iterated logarithm, etc.

Theorem (Martin-Löf, 1966)

There is a universal μ-test $\mathbf{t}$:

$$\omega \text{ is } \mu\text{-random} \iff \omega \text{ withstands } \mathbf{t}.$$

This test can be expressed in terms of Kolmogorov complexity.

Algorithmically random sequences

Binary sequences

- Kolmogorov,
- Martin-Löf, 1966,
- Levin,
- Chaitin,
- Schnorr,
- Gács,
- V'yugin,
- Vovk,
- Asarin,
- Van Lambalgen,
- Downey,
- Hirschfeldt,
- Li,
- Vitanyi,
- Miller,

More general objects

- Asarin, 1986.
  Random functions (Brownian motion).
- Barmpalias et al., 2007.
  Random closed subsets of $\{0,1\}^{\mathbb{N}}$.

Abstract spaces

- Weihrauch, Hertling, 1998.
  Topological spaces.
- Gács, 2005.
  Metric spaces.

Definition (Martin-Löf, 1966)

A μ-test is a function $t : \{0,1\}^{\mathbb{N}} \to [0, +\infty]$ satisfying:

1. $\int t \, \mathrm{d}\mu < +\infty$,
2. t is lower semi-computable.

A sequence ω withstands the test t if $t(\omega) < \infty$.

Definition (Martin-Löf, 1966)

A sequence ω is μ-random if it withstands all μ-tests t.

Algorithmic randomness: extensions

Martin-Löf, 1966 $\rightsquigarrow$ Gács, 2005

First extension: space $\{0,1\}^{\mathbb{N}}$ $\rightsquigarrow$ computable metric space X
sequence ω $\rightsquigarrow$ point x

Definition (Martin-Löf, 1966)

A μ-test is a function $t : \{0,1\}^{\mathbb{N}} \longrightarrow [0,+\infty]$ satisfying:

1. $\int t \, \mathrm{d}\mu < +\infty$,
2. t is lower semi-computable.

A sequence ω withstands the test t if $t(\omega) < \infty$.

Definition (Martin-Löf, 1966)

A sequence ω is μ-random if it withstands all μ-tests t.

Algorithmic randomness: extensions

Martin-Löf, 1966 $\rightsquigarrow$ Gács, 2005

First extension: space $\{0,1\}^{\mathbb{N}}$ $\rightsquigarrow$ computable metric space X
sequence ω $\rightsquigarrow$ point x

Definition (Martin-Löf, 1966 – Gács, 2005)

A μ-test is a function $t : X \to [0, +\infty]$ satisfying:

1. $\int t \, \mathrm{d}\mu < +\infty$,
2. t is lower semi-computable.

A point x withstands the test t if $t(x) < \infty$.

Definition (Martin-Löf, 1966 – Gács, 2005)

A point x is μ-random if it withstands all μ-tests t.

Second extension: μ-test t $\rightsquigarrow$ uniform test T

Definition (Martin-Löf, 1966 – Gács, 2005)

A μ-test is a function $t : X \to [0, +\infty]$ such that:

1. $\int t \, \mathrm{d}\mu < +\infty$,
2. t is lower semi-computable.

A point x withstands the test t if $t(x) < \infty$.

Definition (Martin-Löf, 1966 – Gács, 2005)

A point x is μ-random if it withstands all μ-tests t.

Second extension: μ-test t $\rightsquigarrow$ uniform test T

Definition (Martin-Löf, 1966 – Gács, 2005)

A uniform test is a function $T : \mathcal{M}(X) \times X \to [0, +\infty]$ such that:

1. $\int t \, \mathrm{d}\mu < +\infty$,
2. t is lower semi-computable.

A point x withstands the test t if $t(x) < \infty$.

Definition (Martin-Löf, 1966 – Gács, 2005)

A point x is μ-random if it withstands all μ-tests t.

Second extension: μ-test t $\rightsquigarrow$ uniform test T

Definition (Martin-Löf, 1966 – Gács, 2005)

A uniform test is a function $T : \mathcal{M}(X) \times X \to [0, +\infty]$ such that:

1. $\int T_\mu \, \mathrm{d}\mu < \infty$ for each $\mu \in \mathcal{M}(X)$ (where $T_\mu(.) = T(\mu, .)$),
2. t is lower semi-computable.

A point x withstands the test t if $t(x) < \infty$.

Definition (Martin-Löf, 1966 – Gács, 2005)

A point x is μ-random if it withstands all μ-tests t.

Second extension: μ-test t $\rightsquigarrow$ uniform test T

Definition (Martin-Löf, 1966 – Gács, 2005)

A uniform test is a function $T : \mathcal{M}(X) \times X \to [0, +\infty]$ such that:

1. $\int T_\mu \, \mathrm{d}\mu < \infty$ for each $\mu \in \mathcal{M}(X)$ (where $T_\mu(.) = T(\mu, .)$),
2. T is lower semi-computable.

A point x withstands the test t if $t(x) < \infty$.

Definition (Martin-Löf, 1966 – Gács, 2005)

A point x is μ-random if it withstands all μ-tests t.

Second extension: μ-test t $\rightsquigarrow$ uniform test T

Definition (Martin-Löf, 1966 – Gács, 2005)

A uniform test is a function $T : \mathcal{M}(X) \times X \to [0, +\infty]$ such that:

1. $\int T_\mu \, \mathrm{d}\mu < \infty$ for each $\mu \in \mathcal{M}(X)$ (where $T_\mu(.) = T(\mu, .)$),
2. T is lower semi-computable.

A point x withstands the test T_μ if $T_\mu(x) < \infty$.

Definition (Martin-Löf, 1966 – Gács, 2005)

A point x is μ-random if it withstands all μ-tests t.

Second extension: μ-test t $\rightsquigarrow$ uniform test T

Definition (Martin-Löf, 1966 – Gács, 2005)

A uniform test is a function $T : \mathcal{M}(X) \times X \to [0, +\infty]$ such that:

1. $\int T_\mu \, \mathrm{d}\mu < \infty$ for each $\mu \in \mathcal{M}(X)$ (where $T_\mu(.) = T(\mu, .)$),
2. T is lower semi-computable.

A point x withstands the test T_μ if $T_\mu(x) < \infty$.

Definition (Martin-Löf, 1966 – Gács, 2005)

A point x is μ-random if it withstands all μ-tests T_μ.

Computable metric space X.

Theorem (Gács, 2005)

There is a universal uniform test $\mathbf{T} : \mathcal{M}(X) \times X \to [0, +\infty]$*:*

$$\text{a point } x \text{ is } \mu\text{-random} \iff x \text{ passes the test } \mathbf{T}_\mu.$$

Computable metric space X.

Theorem (Gács, 2005)

Suppose X satisfies the Boolean inclusion property.
There is a universal uniform test $\mathbf{T} : \mathcal{M}(X) \times X \to [0, +\infty]$:

$$\text{a point } x \text{ is } \mu\text{-random} \iff x \text{ passes the test } \mathbf{T}_\mu.$$

Computable metric space X.

Theorem (Gács, 2005)

Suppose X satisfies the Boolean inclusion property.
There is a universal uniform test $\mathbf{T} : \mathcal{M}(X) \times X \to [0, +\infty]$:

$$\text{a point } x \text{ is } \mu\text{-random} \iff x \text{ passes the test } \mathbf{T}_\mu.$$

Theorem

The Boolean inclusion property is not necessary.
Every computable metric space admits a universal uniform test.

Algorithmic randomness

Martin-Löf, 1966 $\rightsquigarrow$ Gács, 2005

Computable metric space X.

Theorem (Gács, 2005)

Suppose X satisfies the Boolean inclusion property.
There is a universal uniform test $\mathbf{T} : \mathcal{M}(X) \times X \to [0, +\infty]$:

$$\text{a point } x \text{ is } \mu\text{-random} \iff x \text{ passes the test } \mathbf{T}_\mu.$$

Theorem

The Boolean inclusion property is not necessary.
Every computable metric space admits a universal uniform test.

Let μ be a probability measure on X, and $t : X \to [0, +\infty]$ a μ-test:

Algorithmic randomness

Martin-Löf, 1966 $\rightsquigarrow$ Gács, 2005

Computable metric space X.

Theorem (Gács, 2005)

Suppose X satisfies the Boolean inclusion property.
There is a universal uniform test $\mathbf{T} : \mathcal{M}(X) \times X \to [0, +\infty]$:

$$\text{a point } x \text{ is } \mu\text{-random} \iff x \text{ passes the test } \mathbf{T}_\mu.$$

Theorem

The Boolean inclusion property is not necessary.
Every computable metric space admits a universal uniform test.

Let μ be a probability measure on X, and $t : X \to [0, +\infty]$ a μ-test:

Theorem (μ-tests versus uniform tests)

There is a uniform test $T : \mathcal{M}(X) \times X \to [0, +\infty]$ such that $T_\mu = t$.

Applications of algorithmic randomness

Probabilistic statement:

$$P(\omega) \text{ holds for } \mu\text{-almost every sequence } \omega$$

becomes

$$P(\omega) \text{ holds for every } \mu\text{-random sequence } \omega.$$

Examples

Strong law of large numbers, law of the iterated logarithm, etc.

Applications of algorithmic randomness

Probabilistic statement:

$$P(x) \text{ holds for } \mu\text{-almost every point } x$$

becomes

$$P(x) \text{ holds for every } \mu\text{-random point } x.$$

Examples

Birkhoff ergodic theorem, Shannon-McMillan-Breiman theorem, convergence of random variables, etc.

Computability theory is topological

Theorem

On $\mathbb{R}$, every computable function is continuous.

Computability theory is topological

Theorem

On $\mathbb{R}$, every computable function is continuous.

Theorem

On $\mathbb{R}$, nothing is decidable.

Computability theory is topological

Theorem

On $\mathbb{R}$, every computable function is continuous.

Theorem

On $\mathbb{R}$, nothing is decidable.

Theorem

Representing real numbers by their binary expansion is not suitable.

$$\xrightarrow{bin(x)} \boxed{algorithm} \xrightarrow{bin(f(x))}$$

It makes the function $x \mapsto 3x$ non-computable.

Computability theory is topological

Theorem

On $\mathbb{R}$, every computable function is continuous.

Theorem

On $\mathbb{R}$, nothing is decidable.

Theorem

Representing real numbers by their binary expansion is not suitable.

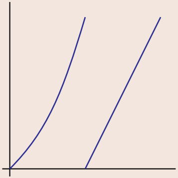
$$\xrightarrow{\mathit{bin}(x)} \boxed{\mathit{algorithm}} \xrightarrow{\mathit{bin}(f(x))}$$

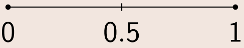
It makes the function $x \mapsto 3x$ non-computable.

Proof.

$[0,1]$ and $\{0,1\}^{\mathbb{N}}$ are not homeomorphic. $\square$

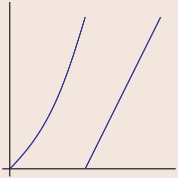
Computability theory is topological

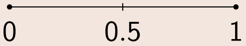




$[0,1]$ and $\{0,1\}^{\mathbb{N}}$, with the Lebesgue measure, are isomorphic.

Computability theory is topological





$[0,1]$ and $\{0,1\}^{\mathbb{N}}$, with the Lebesgue measure, are isomorphic.

Computing on probability spaces ? almost-everywhere computability, decidability ?

Algorithmic probability theory

Definition (2.2.0.1)

(X, μ) is a computable probability space if:

- X is a computable metric space,
- μ is a computable probability measure on X.

Algorithmic probability theory

Definition (2.2.0.1)

(X, μ) is a computable probability space if:

- X is a computable metric space,
- μ is a computable probability measure on X.

Definition (attempt...)

$f : X \to Y$ is almost computable if it is computable on a set A satisfying $\mu(A) = 1$.

Algorithmic probability theory

Definition (2.2.0.1)

(X, μ) is a computable probability space if:

- X is a computable metric space,
- μ is a computable probability measure on X.

Definition (attempt...)

$f : X \to Y$ is almost computable if it is computable on a set A satisfying $\mu(A) = 1$.

Theorem (1.6.2.1)

Let $f : X \to Y$ be an almost computable function. There is a function g which coincides with f on A, and is computable on a constructive G_δ-set containing A.

Algorithmic probability theory

Definition (2.2.0.1)

(X, μ) is a computable probability space if:

- X is a computable metric space,
- μ is a computable probability measure on X.

Definition (2.2.0.2)

$f : X \to Y$ is almost computable if it is computable on a constructive G_δ-set A satisfying $\mu(A) = 1$.

Theorem (1.6.2.1)

Let $f : X \to Y$ be an almost computable function. There is a function g which coincides with f on A, and is computable on a constructive G_δ-set containing A.

Definition (2.2.1.2)

A set $A \subseteq X$ is almost decidable if the function $1_A : X \to \{0, 1\}$ is almost computable.

On $\mathbb{R}$

Interval $[x_1, x_2]$ with x_1, x_2 computable and $\mu(x_1) = \mu(x_2) = 0$.

Algorithmic probability theory

Definition (2.2.1.2)

A set $A \subseteq X$ is almost decidable if the function $1_A : X \to \{0,1\}$ is almost computable.

On $\mathbb{R}$

Interval $[x_1, x_2]$ with x_1, x_2 computable and $\mu(x_1) = \mu(x_2) = 0$.

Proposition (2.2.1.1)

If μ is computable and A is almost decidable, then $\mu(A)$ is computable.

Algorithmic probability theory

Definition (2.2.1.2)

A set $A \subseteq X$ is almost decidable if the function $1_A : X \to \{0, 1\}$ is almost computable.

On $\mathbb{R}$

Interval $[x_1, x_2]$ with x_1, x_2 computable and $\mu(x_1) = \mu(x_2) = 0$.

Proposition (2.2.1.1)

If μ is computable and A is almost decidable, then $\mu(A)$ is computable.

Definition (2.2.0.2)

A morphism $f : (X, \mu) \to (Y, \nu)$ is an almost computable function $f : X \to Y$ which maps μ to ν (i.e. $\nu = \mu f^{-1}$).

Algorithmic probability theory and Random points

Theorem

When restricting to random points,

	On (X, μ)	$\rightsquigarrow$	*On* (R_μ, μ)
function	*almost computable*		*computable*
set	*almost decidable*		*decidable*
sequence	*effective a.e. convergence*		*pointwise convergence*

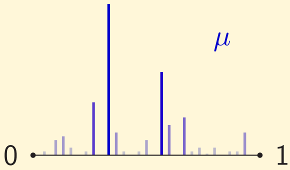
Proposition (3.2.0.8)

- *Morphisms preserve randomness.*
- *Isomorphisms, when restricted to random points, are computable homeomorphisms.*

$X = [0,1]$, μ computable probability measure.

Question

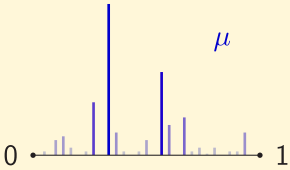
Is it possible that $\mu(\{x\}) \neq 0$ for all computable x ?



$X = [0, 1]$, μ computable probability measure.

Question

Is it possible that $\mu(\{x\}) \neq 0$ for all computable x ?



Theorem (2.2.1.2)

No.

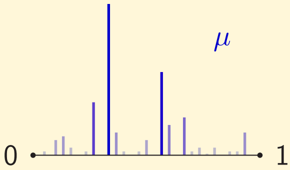
Algorithmic probability theory

Existence of almost decidable sets

$X = [0,1]$, μ computable probability measure.

Question

Is it possible that $\mu(\{x\}) \neq 0$ for all computable x ?



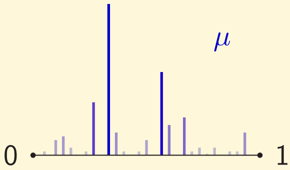
Theorem (2.2.1.2)

No. There is a computable dense sequence of μ-continuity points.

$X = [0,1]$, μ computable probability measure.

Question

Is it possible that $\mu(\{x\}) \neq 0$ for all computable x ?



Theorem (2.2.1.2)

No. There is a computable dense sequence of μ-continuity points.

Proof.

Application of: "$\mathbb{R}_c$ is effectively uncountable". □

(X, μ) computable probability space.

Theorem (2.2.1.2)

There is a basis of almost decidable balls.

(X, μ) computable probability space.

Theorem (2.2.1.2)

There is a basis of almost decidable balls.

Theorem (2.2.1.1)

(X, μ) *admits a generalized binary representation.*

(X, μ) computable probability space.

Theorem (2.2.1.2)

There is a basis of almost decidable balls.

Theorem (2.2.1.1)

(X, μ) *admits a generalized binary representation.*

Theorem (2.2.2.1)

When μ has no mass point, (X, μ) is isomorphic to $(\{0, 1\}^{\mathbb{N}}, \mathit{Lebesgue})$.

(X, μ) computable probability space.

Theorem (2.2.1.2)

There is a basis of almost decidable balls.

Theorem (2.2.1.1)

(X, μ) *admits a generalized binary representation.*

Theorem (2.2.2.1)

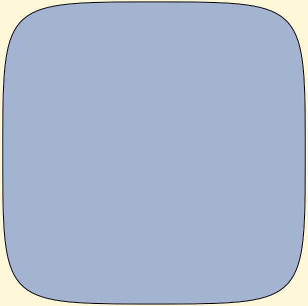
When μ *has no mass point,* (X, μ) *is isomorphic to* $(\{0, 1\}^{\mathbb{N}}, \mathit{Lebesgue})$.

Applications

Transfer of algorithmic randomness, computable symbolic models.

Observing a dynamical system...

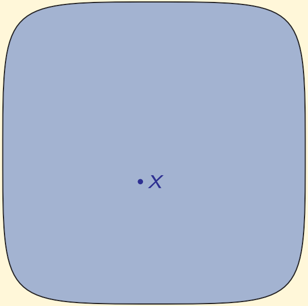
- Space X,
- Transformation $T : X \to X$.



Observing a dynamical system...

...with sharp eyes

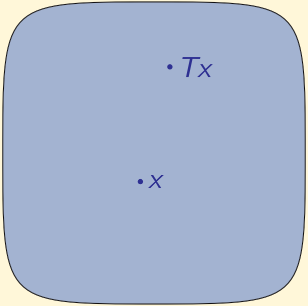
- Space X,
- Transformation $T : X \to X$,
- Laplace's demon.



Observing a dynamical system...

...with sharp eyes

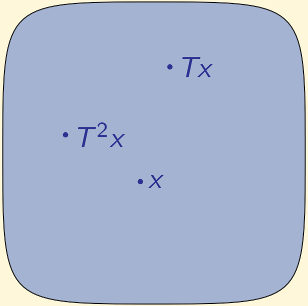
- Space X,
- Transformation $T : X \to X$,
- Laplace's demon.



Observing a dynamical system...

...with sharp eyes

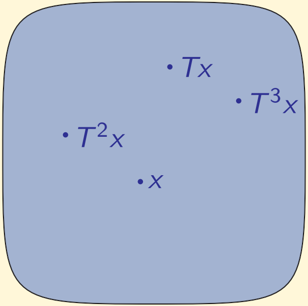
- Space X,
- Transformation $T : X \to X$,
- Laplace's demon.



Observing a dynamical system...

...with sharp eyes

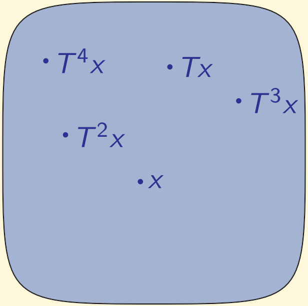
- Space X,
- Transformation $T : X \to X$,
- Laplace's demon.



Observing a dynamical system...

...with sharp eyes

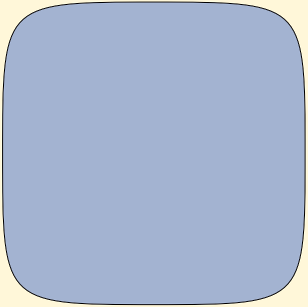
- Space X,
- Transformation $T : X \to X$,
- Laplace's demon.



Observing a dynamical system...

...with finite precision

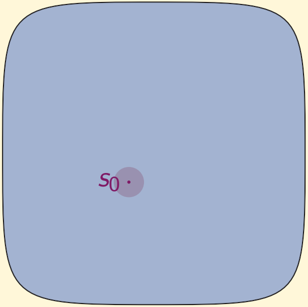
- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.



Observing a dynamical system...

...with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.

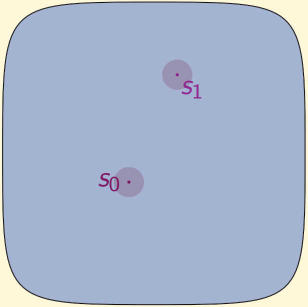


s_0

Observing a dynamical system...

...with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.

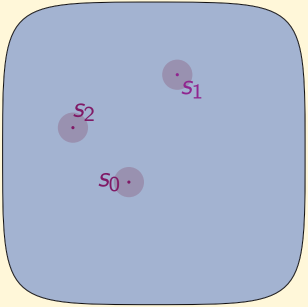


s_0, s_1

Observing a dynamical system...

...with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.

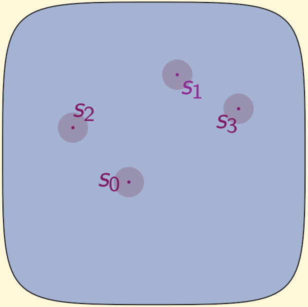


s_0, s_1, s_2

Observing a dynamical system...

...with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.

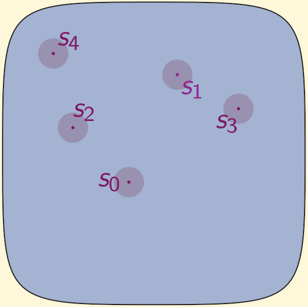


s_0, s_1, s_2, s_3

Observing a dynamical system...

...with finite precision

- Space X,
- Transformation $T : X \to X$,
- Precision $\epsilon > 0$.

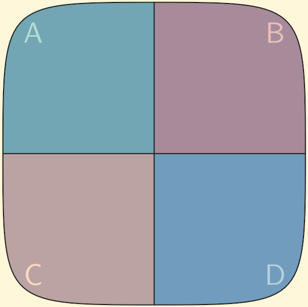


$s_0, s_1, s_2, s_3, s_4, \ldots$

Observing a dynamical system...

...through a partition

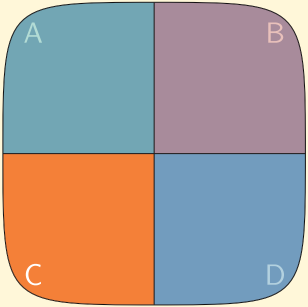
- Space X,
- Transformation $T : X \to X$,
- Partition $P = \{A, B, C, D\}$.



Observing a dynamical system...

...through a partition

- Space X,
- Transformation $T : X \to X$,
- Partition $P = \{A, B, C, D\}$.

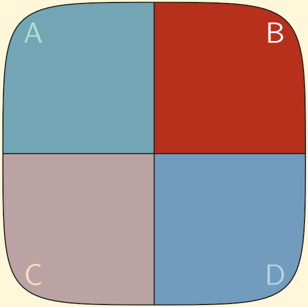


C

Observing a dynamical system...

...through a partition

- Space X,
- Transformation $T : X \to X$,
- Partition $P = \{A, B, C, D\}$.

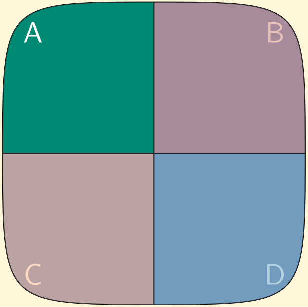


CB

Observing a dynamical system...

...through a partition

- Space X,
- Transformation $T : X \to X$,
- Partition $P = \{A, B, C, D\}$.

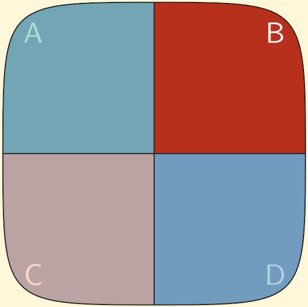


CBA

Observing a dynamical system...

...through a partition

- Space X,
- Transformation $T : X \to X$,
- Partition $P = \{A, B, C, D\}$.

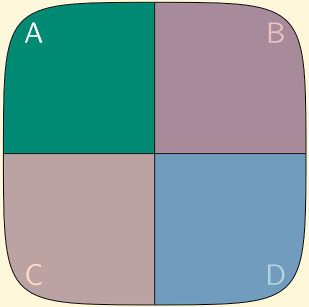


CBAB

Observing a dynamical system...

...through a partition

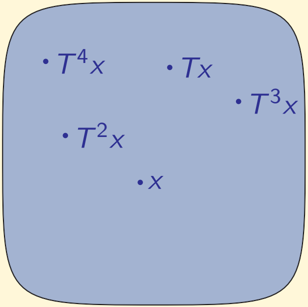
- Space X,
- Transformation $T : X \to X$,
- Partition $P = \{A, B, C, D\}$.



CBABA...

Topological system:

- X compact topological space,
- $T : X \to X$ continuous.



Topological entropy $h(T)$.

Ergodic dynamical system:

- (X, μ) probability space,
- $T : X \to X$ ergodic endomorphism.

Measure-theoretic entropy $h_\mu(T)$.

Complexity of a dynamical system

Topological point of view

Probabilistic point of view

$h(T)$

$h_\mu(T)$

Complexity of a dynamical system

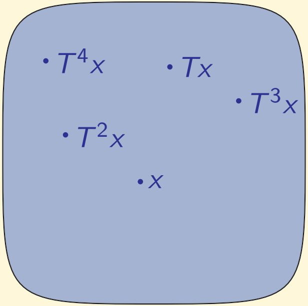
$$h(T) \xleftrightarrow[\text{principle}]{\text{variational}} h_\mu(T)$$

Theorem (Variational principle)

(X, T) topological dynamical system:

$$h(T) = \sup_{\mu \text{ invariant}} h_\mu(T)$$

- Space X,
- Transformation $T : X \to X$.

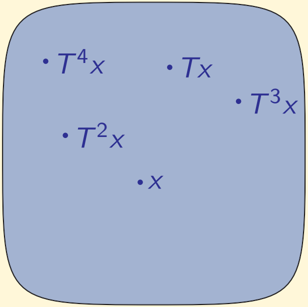


$K(x, T)$: algorithmic complexity of the orbit of x under T.

Observing a dynamical system...

...with finite precision

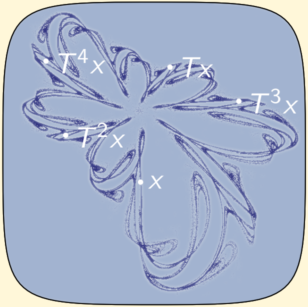
- (X, T) topological system,



$\mathcal{K}_n(x, T)$

...through a partition

- (X, μ, T) ergodic dynamical system

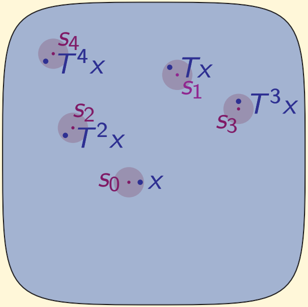


$\mathcal{K}_n(x, T)$

Observing a dynamical system...

...with finite precision

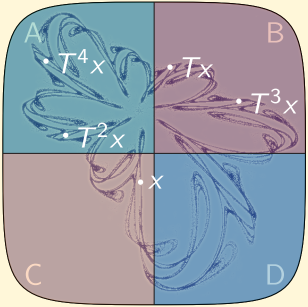
- (X, T) topological system,
- precision $\epsilon > 0$.



$$\mathcal{K}_5(x, T, \epsilon) = K(s_0, s_1, s_2, s_3, s_4)$$

...through a partition

- (X, μ, T) ergodic dynamical system
- partition $P = \{A, B, C, D\}$



$$\mathcal{K}_5(x, T|P) = K(CBABA)$$

Orbit complexity

...with finite precision

(X, T) topological system

$$\overline{\mathcal{K}}(x, T, \epsilon) = \overline{\lim}_n \frac{\mathcal{K}_n(x, T, \epsilon)}{n}$$

$$\underline{\mathcal{K}}(x, T, \epsilon) = \underline{\lim}_n \frac{\mathcal{K}_n(x, T, \epsilon)}{n}$$

...through a partition

(X, μ, T) ergodic system

$$\overline{\mathcal{K}}(x, T|P) = \overline{\lim}_n \frac{\mathcal{K}_n(x, T|P)}{n}$$

$$\underline{\mathcal{K}}(x, T|P) = \underline{\lim}_n \frac{\mathcal{K}_n(x, T|P)}{n}$$

Orbit complexity

...with finite precision

(X, T) topological system

$$\overline{\mathcal{K}}(x, T, \epsilon) = \overline{\lim}_n \frac{\mathcal{K}_n(x, T, \epsilon)}{n}$$

$$\underline{\mathcal{K}}(x, T, \epsilon) = \underline{\lim}_n \frac{\mathcal{K}_n(x, T, \epsilon)}{n}$$

Definition (Galatolo, 2000 – generalizing Brudno, 1983)

$$\overline{\mathcal{K}}(x, T) = \sup_{\epsilon>0} \overline{\mathcal{K}}(x, T, \epsilon)$$

$$\underline{\mathcal{K}}(x, T) = \sup_{\epsilon>0} \underline{\mathcal{K}}(x, T, \epsilon)$$

...through a partition

(X, μ, T) ergodic system

$$\overline{\mathcal{K}}(x, T|P) = \overline{\lim}_n \frac{\mathcal{K}_n(x, T|P)}{n}$$

$$\underline{\mathcal{K}}(x, T|P) = \underline{\lim}_n \frac{\mathcal{K}_n(x, T|P)}{n}$$

Definition (Brudno, 1983 + computable partitions)

$$\overline{\mathcal{K}}_\mu(x, T) = \sup_{P \text{ comp.}} \overline{\mathcal{K}}(x, T|P)$$

$$\underline{\mathcal{K}}_\mu(x, T) = \sup_{P \text{ comp.}} \underline{\mathcal{K}}(x, T|P)$$

Orbit complexity vs entropy

topological context | probabilistic context

$$\begin{tikzcd}[column sep=huge, row sep=huge] h(T) \arrow[r, leftrightarrow, "\text{variational}", "\text{principle}"'] & h_\mu(T) \arrow[d, leftrightarrow] \\ \mathcal{K}(x, T) & \mathcal{K}_\mu(x, T) \end{tikzcd}$$

Orbit complexity vs entropy

topological context | probabilistic context

$$\begin{array}{ccc} h(T) & \xleftrightarrow[\text{principle}]{\text{variational}} & h_\mu(T) \\ & & \Big\updownarrow \text{Brudno} \\ \mathcal{K}(x,T) & & \mathcal{K}_\mu(x,T) \end{array}$$

Theorem (Brudno, 1978)

(X, μ, T) *ergodic dynamical system:*

$$\overline{\mathcal{K}}_\mu(x, T) \quad = \quad h_\mu(T) \qquad \textit{for } \mu\textit{-almost every } x$$

Orbit complexity vs entropy

topological context | probabilistic context

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
h(T) \arrow[r, leftrightarrow, "\text{variational}", "\text{principle}"'] & h_\mu(T) \arrow[d, leftrightarrow, "\begin{array}{c}\text{Brudno}\\ \text{theorem 5.1.4.2}\\ \text{(random)}\end{array}"] \\
\mathcal{K}(x, T) & \mathcal{K}_\mu(x, T)
\end{tikzcd}
$$

Theorem (5.1.4.2)

(X, μ, T) *computable ergodic dynamical system:*

$$\overline{\mathcal{K}}_\mu(x, T) \quad = \quad h_\mu(T) \qquad \textit{for every } \mu\textit{-random } x$$

Orbit complexity vs entropy

topological context   probabilistic context

$$\begin{tikzcd}
h(T) \arrow[rr, leftrightarrow, "\text{variational}", "\text{principle}"'] & & h_\mu(T) \arrow[d, leftrightarrow, "\begin{array}{c}\text{Brudno}\\ \text{theorem 5.1.4.2}\\ \text{(random)}\end{array}"] \\
\mathcal{K}(x,T) & & \mathcal{K}_\mu(x,T)
\end{tikzcd}$$

Theorem (5.1.4.2)

(X, μ, T) *computable ergodic dynamical system:*

$$\underline{\mathcal{K}}_\mu(x,T) \stackrel{?}{=} \overline{\mathcal{K}}_\mu(x,T) = h_\mu(T) \qquad \textit{for every } \mu\textit{-random } x$$

Orbit complexity vs entropy

topological context | probabilistic context

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
h(T) \arrow[r, leftrightarrow, "\text{variational}", "\text{principle}"'] & h_\mu(T) \arrow[dl, leftrightarrow, "\text{Brudno}"', "\text{White}"] \arrow[d, leftrightarrow, "\text{Brudno theorem 5.1.4.2 (random)}"] \\
\mathcal{K}(x, T) & \mathcal{K}_\mu(x, T)
\end{tikzcd}
$$

Theorem (Brudno, 1983 – White, 1993)

(X, T) *topological dynamical system:*

$$\underline{\mathcal{K}}(x, T) \;=\; \overline{\mathcal{K}}(x, T) \;=\; h_\mu(T) \qquad \textit{for } \mu\textit{-almost every } x$$

Orbit complexity vs entropy

topological context — probabilistic context

$$\begin{tikzcd}[column sep=huge, row sep=huge]
h(T) \arrow[r, leftrightarrow, "\text{variational}", "\text{principle}"'] & h_\mu(T) \arrow[d, leftrightarrow, "\text{Brudno}\ \text{theorem 5.1.4.2}\ \text{(random)}"] \\
\mathcal{K}(x,T) \arrow[ur, leftrightarrow, "\text{Brudno}", "\text{White}"'] \arrow[r, leftrightarrow, "\text{theorem 5.3.0.3}", "\text{(random)}"'] & \mathcal{K}_\mu(x,T)
\end{tikzcd}$$

Theorem (5.3.0.3)

(X, μ, T) computable ergodic dynamical system, with X compact:

$$\begin{aligned}
\underline{\mathcal{K}}(x,T) &= \underline{\mathcal{K}}_\mu(x,T) \\
\overline{\mathcal{K}}(x,T) &= \overline{\mathcal{K}}_\mu(x,T)
\end{aligned}$$

for every μ-random x

Orbit complexity vs entropy

Proof.

- Shannon-McMillan-Breiman theorem for random points,
- Birkhoff ergodic theorem for random points,
- both derived from V'yugin's results on $\{0,1\}^{\mathbb{N}}$, using computable partitions.

$\square$

Orbit complexity vs entropy

topological context probabilistic context

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
h(T) \arrow[r, leftrightarrow, "\text{variational}", "\text{principle}"'] & h_\mu(T) \arrow[d, leftrightarrow, "\begin{array}{c}\text{Brudno}\\ \text{theorem 5.1.4.2}\\ \text{(random)}\end{array}"] \\
\mathcal{K}(x, T) \arrow[ur, leftrightarrow, "\text{Brudno}", "\text{White}"'] \arrow[r, leftrightarrow, "\text{theorem 5.3.0.3}", "\text{(random)}"'] & \mathcal{K}_\mu(x, T)
\end{tikzcd}
$$

Orbit complexity vs entropy

topological context | probabilistic context

$$\begin{tikzcd}[column sep=huge, row sep=huge]
h(T) \arrow[r, leftrightarrow, "\text{variational}", "\text{principle}"'] \arrow[d, leftrightarrow, "\text{theorem 5.2.3.1}"'] & h_\mu(T) \arrow[d, leftrightarrow, "\text{Brudno theorem 5.1.4.2 (random)}"] \arrow[dl, leftrightarrow, "\text{Brudno White}"] \\
\mathcal{K}(x,T) \arrow[r, leftrightarrow, "\text{theorem 5.3.0.3}", "\text{(random)}"'] & \mathcal{K}_\mu(x,T)
\end{tikzcd}$$

Theorem (5.2.3.1)

(X, T) *topological system:*

$$\sup_x \underline{\mathcal{K}}(x,T) = \sup_x \overline{\mathcal{K}}(x,T) = h(T)$$

(X, T) topological system.

Upper-bound:

Theorem (Brudno)

For all x,

$$\overline{\mathcal{K}}(x, T) \leq h(T).$$

The set of simple orbits is small.

Theorem (5.2.3.2)

Let $Y_\alpha = \{x : \underline{\mathcal{K}}(x, T) \leq \alpha\}$.

$$h(T, Y_\alpha) \leq \alpha.$$

(X, T) topological system.

Upper-bound:

Theorem (Brudno)

For all x,

$$\overline{\mathcal{K}}(x, T) \leq h(T).$$

The set of simple orbits is small.

Theorem (5.2.3.2)

Let $Y_\alpha = \{x : \underline{\mathcal{K}}(x, T) \leq \alpha\}$.

$$h(T, Y_\alpha) \leq \alpha.$$

Corollary

$$h(T) = \sup_x \underline{\mathcal{K}}(x, T) = \sup_x \overline{\mathcal{K}}(x, T)$$

Conclusion
Contributions

1. Structure dedicated to semi-computability,
2. Framework for computability and probabilities,
3. Integration of algorithmic randomness to general probability theory,
4. Results about algorithmic complexity of orbits, relations with algorithmic randomness.

Conclusion

Further questions

1. Computability and measure
   1. Effective integration theory (Edalat, 2007) and randomness,
   2. Computation models on "physical" spaces,
2. Algorithmic randomness
   1. Particular applications (e.g. Asarin's work on random functions),
   2. Characterization of randomness using Kolmogorov complexity on metric spaces,
3. Dynamical systems
   1. Computability of invariant measures,
   2. Relations between algorithmic complexity and Lyapunov exponents,

Merci

Thanks

Grazie

ممنون

شكرا