# Interactions of computability and randomness

André Nies

University of Auckland

LSE, June 2015

# A two-way interaction:

Randomness

interacts with

Computability

# Part 1: Studying randomness via computability

**Main idea**

Mathematical notions of randomness can be defined and studied using algorithmic methods.

- In contrast to the setting of probability theory, it makes sense to say that an individual object is random.
- There is no single "best" notion of algorithmic randomness. Rather, randomness notions form a hierarchy.
- Randomness of a real $z \in [0,1]$ in a specific sense is equivalent to differentiability at $z$ of an appropriate kind of computable functions $f\colon [0,1] \to \mathbb{R}$. Extensions to functions $f$ defined on $[0,1]^n$.

# Part 2: Studying computational lowness via randomness

Intuitively speaking, an object (such as a real, a set of natural numbers, or a function) is low if it is close to being computable.

**Main idea**

Lowness properties can be defined and studied via randomness.

For instance, in a sense to be specified,

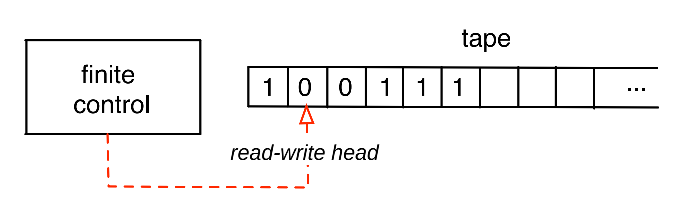
being close to computable is equivalent to being far from random.

# Part I

## Studying randomness via computability

# Basics of computability theory 1

A Turing machine in action looks like this:




The finite control holds a Turing program.

A function $F \colon \mathbb{N} \to \mathbb{N}$ is called computable if there is a Turing program which, beginning with $n$ in binary on the tape, ends with $F(n)$ in binary on the tape:

$$n \longrightarrow \boxed{\text{Turing program}} \longrightarrow F(n)$$

# Computable reals

In the definition of computable function, $\mathbb{N}$ can be replaced by domains that are effectively encoded by natural numbers, such as the rationals $\mathbb{Q}$.

- A real $r \in \mathbb{R}$ is computable if there is a computable sequence $(q_n)_{n\in\mathbb{N}}$ of rational numbers such that $|r - q_n| < 2^{-n-1}$ for each $n$.
- Examples of computable reals are $\sqrt{2}, \pi, e, \ldots$

# Randomness via probability theory

Imagine we toss a fair coin repeatedly. This is modelled as follows.

- We have a sequence $(X_n)_{n \in \mathbb{N}}$ of $0, 1$-valued "random variables" on a probability space $(M, \mathcal{B}, P)$.
- The $X_n$ are independent. We have $P[X_n = 0] = 1/2$ for each $n$.
- Each element $w$ of the space determines a sequence of coin tosses, where the $n$-th bit is $X_n(w)$.
- To say that a property holds for a "random" sequence means that the property holds with probability $1$. Thus, the exceptions form a null set. Random sequences are typical.
- An example of such a property is the law of large numbers: for a random $w$, we have $\frac{1}{n} \sum_{i<n} X_n(w) \to 1/2$.

# The probability spaces

In the following, the probability space will be either

- Cantor space $\{0,1\}^{\mathbb{N}}$ with the product measure, where $\{0,1\}$ is equipped with the measure such that both $0, 1$ have probability $1/2$, or
- the unit interval $[0,1]$ of reals, with Lebesgue measure.

$Z \in \{0,1\}^{\mathbb{N}}$ is an infinite sequence of bits. We identify $Z$ with a subset of $\mathbb{N}$ (and call $Z$ a set).
The two spaces are equivalent (outside a co-countable set) via the binary expansion of reals.

# Algorithmic randomness notions

## The idea in algorithmic randomness

$z$ is random $\iff$ $z$ avoids each algorithmic null set.

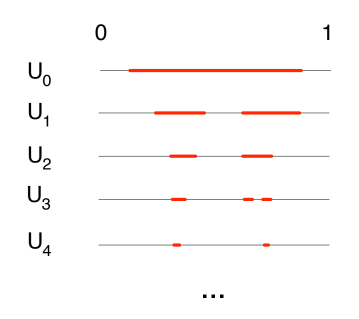
- We have to specify what we mean by an algorithmic null set.
- For instance, having more than 3/4 zeros in arbitrarily long initial segments will be an algorithmic null set in the sense of Martin-Löf.

# Algorithmic null sets in the sense of Martin-Löf (1966)

An open set $U \subseteq [0,1]$ is called computably enumerable if there is an effective list $I_0, I_1, \ldots$ of open intervals with rational endpoints such that $U = \bigcup_r I_r$.

A sequence $(U_n)_{n \in \mathbb{N}}$ of open sets is called a Martin-Löf test if

- the $U_n$ are computably enumerable, where the listing procedure has $n$ as a parameter, and
- $U_n$ has measure at most $2^{-n}$ for each $n$.



## Definition

We call $\bigcap_n U_n$ an algorithmic null set in the sense of Martin-Löf.

A real $r$ is Martin-Löf random if $r \notin \bigcap_n U_n$ for each ML-test $(U_n)_{n\in\mathbb{N}}$.

No computable real $r$ is ML-random:
if $(q_n)_{n\in\mathbb{N}}$ is a computable sequence of rational numbers such that $|r - q_n| < 2^{-n-1}$ for each $n$, let

$$U_n = (q_n - 2^{-n-1}, q_n + 2^{-n-1}).$$

Then $(U_n)_{n\in\mathbb{N}}$ is a ML-test such that $r \in \bigcap_n U_n$.

# Functions of bounded variation...

A function $f\colon [0,1] \to \mathbb{R}$ is of bounded variation if it doesn't "wiggle" too much:

$$V(f) = \sup \sum_{i=1}^{n-1} |f(t_{i+1}) - f(t_i)| < \infty,$$

where the sup is taken over all collections $t_1 \le t_2 \le \ldots \le t_n$ in $[0,1]$.

Examples:

- nondecreasing functions,
- Lipschitz functions
- $x^2 \sin(1/x)$

On the other hand $x \sin(1/x)$ is not of bounded variation.

# ... are differentiable outside a null set

A function $f$ of bounded variation is differentiable at a "random" real:

**Theorem (Lebesgue,1904, using Jordan, 1879)**

*Let $f\colon [0,1] \to \mathbb{R}$ be of bounded variation. Then*

$f'(r)$ *exists for each* $r$ *outside a null set (depending on* $f$*).*

# Complexity of the exception set

**Theorem (Demuth 1975/Brattka, Miller, Nies, TAMS, 2015)**

*Let $z \in [0,1]$. Then*

*$z$ is Martin-Löf random $\Longleftrightarrow$*

*$f'(z)$ exists, for each function $f$ of bounded variation such that $f(q)$ is a computable real, uniformly for each rational $q$.*

- The implication "$\Rightarrow$" is an algorithmic version of the classical theorem.
- For the implication "$\Leftarrow$", one builds a computable function $f$ of bounded variation that is only differentiable at the Martin-Löf random reals.

How about the smaller class of nondecreasing functions?

# Randomness via betting strategies

Computable betting strategies (also called martingales) are certain computable functions $M$ from binary strings to the non-negative reals.

- Let $Z$ be a sequence of bits. When the player has seen the string $\sigma$ of the first $n$ bits of $Z$, she can bet $q$ on what the next bit $Z(n)$ is. Need $0 \leq q \leq M(\sigma)$
- If she is right, she gets $q$. Otherwise she loses $q$. Fairness means that
$$M(\sigma 0) + M(\sigma 1) = 2M(\sigma)$$
for each string $\sigma$.
- She wins on $Z$ if $M$ is unbounded along $Z$. We call a set $Z$ computably random if no computable betting strategy wins on $Z$.

Martin-Löf random $\Rightarrow$ computably random, but not conversely.

# Upper and lower derivatives

Let $f\colon [0,1] \to \mathbb{R}$. We define

$$\begin{aligned} \overline{D}f(z) &= \limsup_{h\to 0} \frac{f(z+h)-f(z)}{h} \\ \underline{D}f(z) &= \liminf_{h\to 0} \frac{f(z+h)-f(z)}{h} \end{aligned}$$

Then

$$f'(z) \text{ exists} \iff \overline{D}f(z) \text{ equals } \underline{D}f(z) \text{ and is finite.}$$

# Computable randomness and differentiability

**Theorem (Brattka, Miller, Nies, TAMS, 2015)**

*Let* $r \in [0,1]$. *Then*

$r$ *(written in binary) is computably random* $\Longleftrightarrow$

$g'(r)$ *exists, for each nondecreasing function* $g$ *that is uniformly computable on the rationals.*

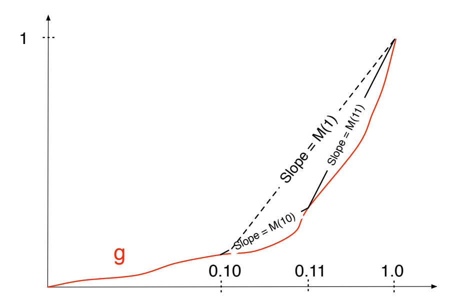
- Classically, we cannot distinguish the exception sets for nondecreasing functions from the more general exception sets for functions of bounded variation.
- Algorithmic randomness provides a finer view: to ensure a computable function of bounded variation is differentiable at $z$, one needs the stronger notion of Martin-Löf randomness of $z$.

# Nondecreasing functions versus betting strategies

$r$ computably random $\Rightarrow g'(r)$ exists.

We prove the contraposition. In the simplest case, suppose that the lower derivative $\underline{D}g(r)$ equals $+\infty$. Then the following computable betting strategy $M$ succeeds on $r$: for a binary string $\sigma$, $M(\sigma)$ is the slope of $g$ between the points $0.\sigma$ and $0.\sigma + 2^{-|\sigma|}$.

This is clearly a betting strategy:
the picture shows,
for instance, that
$2M(1) = M(10) + M(11)$.



# Polynomial time randomness

## Definition

- A martingale $M\colon 2^{<\omega} \to \mathbb{R}$ is called polynomial time if for a string $\sigma$, one can compute $n$ bits of the real $M(\sigma)$ in time polynomial in $|\sigma| + n$.
- A real $z$ is polynomial time random if no polynomial time martingale succeeds on its binary expansion.

Such a real exists in all time classes properly containing $P$, such as $\mathrm{DTIME}(n^{\log n})$.

# Polynomial time functions $g\colon (0,1) \to \mathbb{R}$

- A sequence of rationals $(p_i)_{i \in \mathbb{N}}$ is called a Cauchy name if $\forall k > i \, |p_i - p_k| \le 2^{-i}$
- In the efficient setting, one uses a compact set of Cauchy names to represent reals.
- A sequence $(a_i)_{i \in \mathbb{N}}$, where $a_i \in \{-1, 0, 1\}$, $a_0 = 0, a_1 = 1$, determines the real $\sum_{i \in \mathbb{N}} a_i 2^{-i} \in (0,1)$.
- A function $g\colon (0,1) \to \mathbb{R}$ is called polynomial time if there is a polynomial time oracle Turing machine turning every such Cauchy name for $x$ into a Cauchy name for $g(x)$.

Functions such as $e^x$, $x^2$, $\sin x$ are polynomial time.

# Turning a martingale test into a function

For a martingale $M$, the corresponding measure $\mu_M$ is given by

$$\mu_M([\sigma]) = 2^{-|\sigma|} M(\sigma).$$

- $M$ has the savings property if $M(\sigma) \geq M(\tau) - 2$ whenever $\sigma \succeq \tau$. Such martingales are sufficient for computable and polynomial time randomness.
- This implies $M(\sigma) = O(|\sigma|)$, so $M$ grows slowly.
- In particular, $\mu_M$ has no atoms.

Let $g_M(x) = \mu_M[0, x)$ be the (nondecreasing) distribution function.
If $M$ succeeds on $z$ then $\overline{D} g_M(z) = \infty$, so $g'_M(z)$ fails to exist.
If $M$ is computable and has the savings property, then $f$ is computable.
If $M$ is in fact polynomial time, then $g_M$ is polynomial time (Figueira and N, 2013).

# Characterising polynomial time randomness via differentiability

**Theorem (N., STACS 2014)**

A real $z$ is polynomial time random $\iff$
$g'(z)$ exists for every nondecreasing polynomial time function $g$.

We can develop the theory of martingales with bases $b$ other than $2$, and define polynomial time randomness in base $b$.

We get the same connections with nondecreasing functions.

Since the right hand side of the theorem is base invariant, we obtain

**Corollary (Figueira-Nies)**

*Polynomial time randomness of a real is base invariant.*

# Randomness and differentiability in higher dimensions

**Theorem (Galicki, N., Turetsky, 2013)**

*$z \in [0,1]^n$ is in no null effective $G_\delta$ class $\Longleftrightarrow$ every a.e. differentiable computable $f : [0,1]^n \to \mathbb{R}$ is differentiable at $z$.*

# Rademacher's theorem

**Theorem (Rademacher, 1920)**

*Let $f : [0,1]^n \to \mathbb{R}$ be Lipschitz.*
*Then the derivative $Df(z)$ (an element of $\mathbb{R}^n$) exists for almost every vector $z \in [0,1]^n$.*

To define computable randomness of a vector $z \in [0,1]^n$:

- Take the binary expansion of the $n$ components of $z$.
- We can bet on the corresponding sequence of blocks of $n$ bits.

Effective form of Rademacher's Theorem:

**Theorem (Galicki, Turetsky, 2014)**

*$z \in [0,1]^n$ is computably random $\Rightarrow$*
*every computable Lipschitz function $f : [0,1]^n \to \mathbb{R}$ is differentiable at $z$.*

# Monotone functions

The converse fails by the effective form of a result of Dore and Maleva (2011): for $n \geq 2$ there is an effectively closed null class $\mathcal{P} \subseteq [0,1]^n$ that contains a point of differentiability of every computable Lipschitz function.

$f : [0,1]^n \to \mathbb{R}^n$ is monotone if $\langle x - y, f(x) - f(y) \rangle \geq 0$ for each $x, y$.

**Theorem (Galicki)**

*If $z \in [0,1]^n$ is not computably random, then some computable monotone function $f : [0,1]^n \to \mathbb{R}^n$ is not differentiable at $z$.*

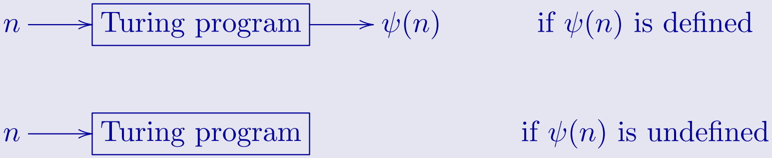
This uses the theory of optimal transport (Monge, Kantorevich, recent books by Villani).
Version for Lipschitz functions in progress.

# Part II

## Studying computability via randomness

# Basics of computability theory 2

A function $\psi\colon \mathbb{N} \to \mathbb{N}$ is partial computable if there is a Turing program which, with $n$ on the input tape, outputs $\psi(n)$ if defined, and loops forever otherwise.




We say that $A \subseteq \mathbb{N}$ is computably enumerable (c.e.) if $A$ is the domain of a partial computable function. Equivalently, one can effectively enumerate the elements of $A$ in some order.

# Basics of computability theory 3

$(W_e)_{e\in\mathbb{N}}$ is an effective listing of all the computably enumerable sets.

The halting problem is a universal computably enumerable set:

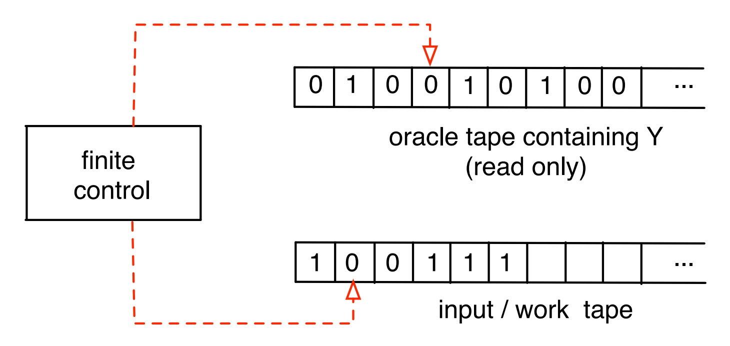
$$\mathcal{H} = \{\langle x, e\rangle \colon\ x \in W_e\}.$$

# Basics of computability theory 4

For sets $X, Y \subseteq \mathbb{N}$, we write

$$X \leq_T Y$$

($X$ is Turing below $Y$) if an "oracle" Turing machine can compute $X$ by asking queries to $Y$ on its oracle tape.



# Prefix-free machines

A partial computable function from binary strings to binary strings is called prefix-free machine if its domain is an anti-chain under the prefix relation of strings.

There is a universal prefix-free machine $\mathbb{U}$: for every prefix-free machine $M$,

$$M(\sigma) = y \text{ implies } \mathbb{U}(\tau) = y,$$

for a string $\tau$ that is only by a constant $d_M$ longer than $\sigma$.

# Descriptive string complexity $K$

- The prefix-free Kolmogorov complexity is the length of a shortest $\mathbb{U}$-description of $y$:
$$K(y) = \min\{|\sigma| \colon \mathbb{U}(\sigma) = y\}.$$
- One can show that $2^{-K(y)}$ is proportional to
$$\lambda\{X \in 2^{\mathbb{N}} \colon \mathbb{U}(\sigma) = y \text{ for some initial segment } \sigma \text{ of } X\},$$
where $\lambda$ denotes product measure in Cantor space $2^{\mathbb{N}}$. Informally, this is the probability that $\mathbb{U}$ prints $y$. This only works with prefix-free machines.

# The Schnorr/Levin 1973 Theorem

We think of a string $\tau$ as random if it is incompressible: $K(\tau) > |\tau| - b$ for some "small" constant $b$.

For an infinite sequence of bits $Z$, let

$$Z\upharpoonright_n = Z(0)\ldots Z(n-1).$$

An infinite sequence of bits $Z$ is Martin-Löf random iff each of its initial segments is random as a string:

**Theorem (Schnorr 1973; Levin 1973)**

*$Z$ is ML-random $\Longleftrightarrow$*

*there is $b \in \mathbb{N}$ such that $\forall n\, [K(Z\upharpoonright_n) > n - b]$.*

Chaitin's halting probability is ML-random:

$$\Omega = \sum\{2^{-|\sigma|} \colon \mathbb{U} \text{ halts on input } \sigma\}.$$

# Definition of $K$-triviality

In the following, we identify a natural number $n$ with its binary representation (as a string). For a string $\tau$, up to additive const we have $K(|\tau|) \leq K(\tau)$, since we can compute $|\tau|$ from $\tau$.

**Definition (going back to Chaitin, 1975)**

An infinite sequence of bits $A$ is $K$-trivial if, for some $b \in \mathbb{N}$,

$$\forall n \, [K(A \upharpoonright_n) \leq K(n) + b],$$

namely, all its initial segments have minimal $K$-complexity.

It is not hard to see that $K(n) \leq 2 \log_2 n + O(1)$.

$$\begin{array}{llcll} Z \text{ is} & \text{random} & \Longleftrightarrow & \forall n \, [K(Z \upharpoonright_n) > n & -O(1)] \\ A \text{ is} & K\text{-trivial} & \Longleftrightarrow & \forall n \, [K(A \upharpoonright_n) \leq K(n) & +O(1)] \end{array}$$

Thus, being $K$-trivial means being far from random.

# Background on the $K$-trivials[1]

- Chaitin (1975) proved that for each constant $b$ there are only $O(2^b)$ $K$-trivials. From this he derived that each $K$-trivial set is Turing below the halting problem $\mathcal{H}$.
- Solovay (1976) built a non-computable $K$-trivial set.
- This was improved to a computably enumerable example by Downey, Hirschfeldt, Nies, and Stephan (2002).
- They also showed that no $K$-trivial set is Turing equivalent to the halting problem $\mathcal{H}$.

[1] Trivium: an introductory curriculum at a medieval university involving the study of grammar, rhetoric, and logic. Compare with Quadrivium.

# Lowness for Martin-Löf randomness

The following specifies a sense in which a set $A$ is computationally weak when used as an oracle: it doesn't derandomize sequences of bits.

**Definition**

$A$ is low for Martin-Löf randomness if every ML-random set $Z$ is already ML-random with oracle $A$.

- This property was introduced by Zambella (1990).
- Kučera and Terwijn (1999) built a c.e. non-computable set of this kind.
- In contrast,

  Low for computably random $\Rightarrow$ computable (Nies, 2005).

# Far from random = close to computable

- An oracle $A \subseteq \mathbb{N}$ is low for Martin-Löf randomness if every random set is already random with oracle $A$.
- That is, $A$ cannot "derandomize" any random set.
- This means that $A$ is very close to computable.

The following says that far from random = close to computable.

**Theorem (Advances in Mathematics, 2005)**

*Let $A \subseteq \mathbb{N}$. Then*

$$A \text{ is } K\text{-trivial} \iff A \text{ is low for Martin-Löf randomness.}$$

# Lowness paradigms

Paradigms for computational lowness of a set $A$:

- Inertness: $A$ can be computably approximated with a finite total of changes (in the sense of cost functions). This is true for the $K$-trivials
- Oracle weakness: $A$ is not very useful as an oracle (e.g., lowness for Martin-Löf randomness).
- It is easy for oracles to compute $A$. In some sense, “many oracles” compute $A$.

# An instance of the "easy-to-compute" paradigm

$A$ is ML-coverable (Hirschfeldt, Nies, Stephan 2004) if $A \leq_{\mathrm{T}} Y$ for some ML-random $Y$ that is not above the halting problem.

- For c.e. sets $A$, ML-coverable $\Rightarrow$ $K$-trivial (ibd.).
- Frank Stephan 2004 asked whether the converse implication holds. This became a main open question in the area, known as the covering problem.
- It defines $K$-triviality of c.e. sets directly from ML-randomness and Turing reducibility.

# Recent solution of the covering problem (1)

Let $\mathcal{P}$ be a subset of Cantor space $\{0,1\}^{\mathbb{N}}$. The notion of lower density of $\mathcal{P}$ at a point $Y$ goes back to Lebesgue:

$$\underline{\rho}(\mathcal{P} \mid Y) = \inf_m \lambda(\mathcal{P} \cap [Y \upharpoonright_m])/2^{-m}.$$

- This quantity between 0 and 1 tells us "how much" of $\mathcal{P}$ is close to the point $Y$ as we zoom in on $Y$.
- The Lebesgue density theorem says that at almost every point $Y \in \mathcal{P}$, the lower density of $\mathcal{P}$ is $1$.

# Recent solution of the covering problem (2)

Theorem [Bienvenu, Greenberg, Kučera, N. Turetsky, J. European Math. Soc, in press]

Suppose some effectively closed (i.e., $\Pi^0_1$) class $\mathcal{P} \subseteq \{0,1\}^{\mathbb{N}}$ has lower density $< 1$ at some ML-random set $Y \in \mathcal{P}$.

Then $Y$ is Turing above each $K$-trivial set.

Theorem [Day and Miller, Math. Research Letters, in press]

There is an effectively closed class $\mathcal{P}$ and a ML-random set $Y \in \mathcal{P}$ strictly Turing below the halting problem such that $\mathcal{P}$ has lower density $< 1$ at $Y$.

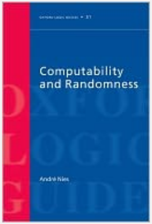
- Thus, there is a single Turing incomplete ML-random $\Delta^0_2$ set $Y$ above all the $K$-trivials!
- BGKNT also showed that this $Y$ must be close to the halting problem.

# Summary

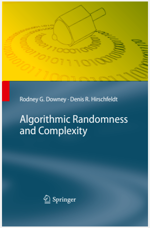
- Randomness can be studied via computability.
- Algorithmic methods lead to a hierarchy of randomness notions for infinite sequences of bist.
- Martin-Löf randomness and computable randomness of reals can be characterized through differentiability of computable functions on the unit interval. In higher dimensions interesting new phenomena.

- Lowness can be studied via randomness.
- Far from random $=$ close to computable.
- Randomness leads to three lowness paradigms: oracle-weakness, inertness, and being easy to compute. Notions introduced via different paradigms often coincide.
- Covering problem recently solved using the analytic notion of density.

# References




My book
"Computability and Randomness",
Oxford University Press, 447 pages, Feb. 2009;
Paperback version Mar. 2012.




Book by Downey and Hirschfeldt:
"Algorithmic Randomness and Complexity",
Springer, $>$ 800 pages, Dec. 2010;

"Randomness and differentiability", with Brattka and Miller,
Transactions AMS, 2015.

Survey "Computing K-trivial sets by incomplete random sets",
7 authors, Bull. Symb Logic, March 2014.