Random reals and infinite time Turing machines

Philipp Schlicht, Universität Bonn

joint work with Merlin Carl, Universität Konstanz

11. September 2016

Random sequences

When is an infinite sequence random?

In other words, we would like to formalize the properties of a sequence obtained by infinitely many tosses of an unbiased coin.

The intuition: an object is random if it satisfies no exceptional properties.

Example

- *Every second digit is* 0.
- *In the limit, there are at least twice as many* 0*s as* 1*s.*

The above sets are null classes.

We can formalize 'exceptional property' by null classes.

Random sequences

Using algorithmic tools, we introduce effective null classes, also called tests.

To be random in an algorithmic sense, a real merely has to avoid these effective null classes, that is, pass those tests.

Definition

- A Martin-Löf test is a uniformly computably enumerable sequence

  $$\langle U_n \mid n \in \omega \rangle$$

  of open subsets of the Cantor space 2^ω such that

  $$\mu(U_n) \leqslant 2^{-n}$$

  for all n.
- A real x is Martin-Löf random if x passes each ML-test, in the sense that x is not in all of the U_n.

Incompressibility

When is an infinite sequence random?

A different answer is: when its initial segments are incompressible.

Definition

- A partial computable function on finite words is *prefix-free* if there are no s, t in its domain with $s \sqsubseteq t$.
- Let
$$\langle M_n \mid n \in \omega \rangle$$
be an effective listing of all prefix-free machines. We define a universal prefix free machine U by
$$U(0^n \sigma) = M_d(\sigma).$$
- Given a string τ, the prefix-free descriptive string complexity $K(\tau)$ is the length of a shortest U-description of x:
$$K(\tau) = \min\{|\sigma| : U(\sigma) = \tau\}.$$

Incompressibility

Informally, a finite string σ is *compressible* if $K(\sigma) \ll |\sigma|$

ML-random sequences can be characterized by their initial segment complexity.

Theorem (Levin-Schnorr 1973)

The following are equivalent.

- *x is ML-random.*
- $\exists b \, \forall n \; K(x \upharpoonright n) \geqslant n - b$.

Hypercomputation

The field *hypercomputation* (higher recursion theory) studies notions of computability beyond Turing computability.

- Π^1_1 sets are a higher analogue of computably enumerable sets, where the steps of an effective enumeration are computable ordinals.
- Hyperarithmetical (i.e. Δ^1_1) sets are a higher analogue of computable sets.

Satz (Gandy, Spector)

The following are equivalent for any subset A of the Cantor space 2^ω.

1. *A is Π^1_1.*
2. *There is a Σ_1-formula φ such that*

$$x \in A \Longleftrightarrow L_{\omega_1^x}[x] \vDash \varphi(x)$$

for all x.

Higher randomness

Already Martin-Löf criticized the classical randomness notions as too weak.

Hjorth and Nies (2007), Yu and Bienvenu, Greenberg and Monin (2015) studied randomness notions at the level of Π^1_1.

Higher randomness

These notions satisfy variants of desirable features of the classical randomness notions, for instance the following.

Theorem (van Lambalgen)

$x \oplus y$ is ML-random if and only x is ML-random and y is ML-random relative to x.

In this situation, we say that x and y are *mutually random*.

We will focus on the property: *Mutual randoms do not share common information.*

This is false for ML-random, but holds for many higher randomness notions.

Higher randomness

Question

Do notions of randomness beyond Π^1_1 have similar desirable properties as the classical randomness notions?

On the level of Σ^1_2, many properties of randomness are independent.

Therefore, we study randomness notions between Π^1_1 and Σ^1_2, defined by infinite Turing machines.

Infinite time Turing machines

Infinite time Turing machines were introduced by Hamkins and Kidder (Hamkins-Lewis 2000).

Hardware:

- tape of length ω
- read/write head.

Software:

- finite alphabet A
- finite set S of states, including some end states
- transition function $A \times S \times \{\text{succ}, \lim\} \to A \times S \times \{\text{left}, \text{right}\}$

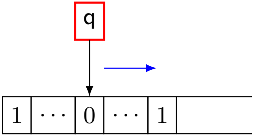
Infinite time Turing machines

We can assume that the letters and states are natural numbers.

The machine runs through steps of the computation at every ordinal time.

At limits λ

- form the lim inf in each cell
- form the lim inf of the previous states
- move the head to the beginning of the tape



Snapshots of a computation

tape cells →

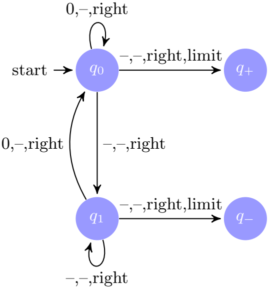
time ↓

time	state	head	0	1	2	3	4	5	⋯
0	0	0	–	–	–	–	–	–	⋯
1	1	1	1	–	–	–	–	–	⋯
2	0	2	1	–	–	–	–	–	⋯
3	1	3	1	–	1	–	–	–	⋯
4	0	4	1	–	1	–	–	–	⋯
5	1	5	1	–	1	–	1	–	⋯
6	0	6	1	–	1	–	1	–	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ω	0	0	1	–	1	–	1	–	⋯
$\omega+1$	1	1	0	–	1	–	1	–	⋯
$\omega+1$	0	2	0	–	1	–	1	–	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\omega\cdot 2$	0	0	0	–	0	–	0	–	⋯
$\omega\cdot 2+1$	1	1	1	–	0	–	0	–	⋯
$\omega\cdot 2+2$	0	2	1	–	0	–	0	–	⋯

Example

Example

Does the letter 0 *appear infinitely often in the input word?*



Strength of infinite time Turing machines

ITTMs can do the following.

- compute the halting problem (for Turing machines)
- test whether a tree is wellfounded, and hence can decide Π^1_1 sets

Writable ordinals

Definition

- x ist *writable* if it can be written, with empty input, by a program which then halts.
- x is *eventually writable* if it can be written and eventually the tape contents is stable.
- x is accidentally writable if it can be written at some time in some computation.

Example

The halting problem for ITTMs is eventually writable.

By coding ordinals by reals, we define the writable ordinals etc.

Writable ordinals

Definition

- λ is the supremum of the writable ordinals.
- ζ is the supremum of the eventually writable ordinals.
- Σ is the supremum of the accidentally writable ordinals.

Then λ is equal to the supremum of the clockable ordinals (halting times).

An important characterization:

Theorem (Welch)

λ, ζ, Σ *is the lexicographically least triple* α, β, γ *with*

$$L_\alpha \prec_{\Sigma_1} L_\beta \prec_{\Sigma_2} L_\gamma.$$

Preservation by random forcing

We distinguish between *random generic* and *random (quasi-generic)*.

Definition

x is random (quasi-generic) over L_α if x avoids every Borel null set with a code in L_α.

Theorem (CS)

λ, ζ and Σ are preserved by random reals over $L_{\Sigma+1}$.

This result is proved via an analysis of a quasi-forcing relation for random reals over admissible sets.

Writable reals from non-null sets

To prove properties of randomness, we need the following analogue to a results of Sacks.

We write $x \leqslant_w y$ ($x \leqslant_{ew} y$, $x \leqslant_{aw} y$) if x is (eventually, accidentally) writable from y.

Theorem (CS)

1. *x is writable if and only if* $\mu(\{y : x \leqslant_w y\}) > 0$
2. *x is eventually writable if and only if* $\mu(\{y : x \leqslant_{ew} y\}) > 0$
3. *x is accidentally writable if and only if* $\mu(\{y : x \leqslant_{aw} y\}) > 0$

This is proved via the preservation of λ, ζ and Σ by sufficiently randoms.

ITTM-random reals

A higher analogue of Π^1_1-random:

Definition
A real x is *ITTM-random* if it avoids every ITTM-semidecidable null set.

Mutual ITTM-randoms have no common information:

Theorem
Suppose that $x \oplus y$ is ITTM-random. If z is writable from x and from y, then z is writable.

This is proved via the previous result about writable reals from non-null sets.

Characterization of ITTM-randoms

By results of Spector and Sacks, the following conditions are equivalent.

- x is Π^1_1-random.
- x is Δ^1_1-random and $\omega_1^x = \omega_1^{\mathrm{CK}}$.

A higher analogue:

Theorem (CS)

The following are equivalent.

- *x is ITTM-random.*
- *x is random over L_Σ and $\Sigma^x = \Sigma$.*

Further results

- similar results for recognizable reals instead of writable reals
- similar results for an ITTM-decidable variant of ITTM-random
- similar results as Hjorth-Nies for a Martin-Löf variant of ITTM-random

Questions

Question

Is $\zeta^x = \zeta$ for every ITTM-random?

Question

Is the set of ITTM-randoms $\mathbf{\Pi^0_3}$?

Question

Is there a concrete description of the set NCR of reals that are not ITTM-random with respect to any continuous measure?

Bibliography

- Merlin Carl, Philipp Schlicht. Randomness via infinite computation and effective descriptive set theory, 2016, in preparation
- Joel David Hamkins, Andy Lewis. Infinite Time Turing Machines. Journal of Symbolic Logic 65(2), 567-604 (2000)
- Greg Hjorth, Andre Nies. Randomness via effective descriptive set theory. Journal of the London Mathematical Society (2), 75(2):495-508 (2007).
- André Nies. Computability and randomness, volume 51 of Oxford Logic Guides. Oxford University Press, Oxford, 2009.
- Philip Welch. Characteristics of discrete transfinite Turing machine models: halting times, stabilization times, and normal form theorems. Theoretical Computer Science, 410 (2009), 426-442