

Computable randomness and martingales a la probability theory

Jason Rute

www.math.cmu.edu/~jrute

Mathematical Sciences Department
Carnegie Mellon University

November 13, 2012
Penn State Logic Seminar

Something provocative

- Martingales are an essential tool in computability theory...
...but the martingales we use are “outdated”.
- Algorithmic randomness is “effective probability theory”...
...but most tools seem to rely on “bit-wise” thinking.
- We often ask what computability says about “classical math”...
...but what does “classical math” tell us about computability.
- Infinitary methods have revolutionized finitary combinatorics...
...so can they revolutionize computability theory?
- Computability theorists study information and knowledge...
...and so do probabilists. What can we learn from them?

This is a talk about martingales

- This is a talk about martingales.
- But what is a martingale?

What is a martingale?

The computability theorist's answer

- **Notation.** Let 2^* denote the set of finite binary strings (words).
- **Definition.** A **martingale** d is a **function** $d: 2^* \rightarrow \mathbb{R}_{\geq 0}$ such that for all $w \in 2^*$

$$\frac{1}{2}d(w0) + \frac{1}{2}d(w1) = d(w).$$

- **Interpretation.** A martingale is a strategy for betting on coin flips.
 w encodes the flips you have seen so far.
 $d(w)$ is how much capital you have after those flips.
- **Observations.**
 - We are implicitly working in $2^{\mathbb{N}}$ under the fair-coin measure.
 - We are assuming finitely-many states, each with a non-zero probability, on each bet.

What is a martingale good for?

The computability theorist's answer

Martingales can be used to **characterize algorithmic randomness**.

- **Main idea.** A string $x \in 2^{\mathbb{N}}$ is “random” if one cannot win unbounded money betting on it with a “computable strategy”.
- **Definition/Example.** A string $x \in 2^{\mathbb{N}}$ is **computably random** if there is no computable martingale d such that

$$\sup_n d(x \upharpoonright n) = \infty.$$

What is known so far?

The computability theorist's answer

Table : Randomness notions defined by betting strategies

	monotone	selection	permutation	injection
total	CR \cup	= CR \cup	= CR \cup	\subset TIR \cup
partial	PCR \cup	= PCR \cup	\subset PPR \cup	\subset PIR \cup
l.s.c.	MLR	= MLR	= MLR	= MLR

	adapted	balanced	process
total	$\dots \subset$ KLR \parallel	\subseteq MLR \parallel	$=$ MLR \parallel
partial	$\dots \subseteq$ KLR \cup	\subseteq MLR \parallel	$=$ MLR \parallel
l.s.c.	$\dots =$ MLR	$=$ MLR	$=$ MLR

Notes on “What is known so far”

The rows refer to the computability of the martingale (computable, partial computable, and lower-semi-computable (right c.e.)).

The columns refer to the bit-selection processes. The first four are the “non-adaptive” strategies. Before seeing any information, they choose which bits to bet on.

- “monotone” means going through every bit in order
- “selection” means going through a computable subset of the bits in order. This is easily the same as monotone, since one can just bet no money on the other bits.
- “permutation” means going through every bit in some computable ordering
- “injection” means going through the bits in some order, never looking at the same bit twice.

The last three columns are strategies that take into account information the gambler has already seen to determine what to bet on next.

- “adaptive” means choosing the next bit to bet on after looking at other bits
- “balanced” means not betting on bits, but instead on clopen sets that are half the measure of set of known information. (This is closer to martingales in probability theory.)
- “process” means the same as balanced, but the sets do not have to be half the measure of the previous. They just need to be a subset of the previous information.

More notes on “What is known so far”

The randomness notions are as follows:

- CR is computable randomness.
- PCR is partial computable randomness.
- KLR is Kolmogorov-Loveland randomness.
- MLR is Martin-Löf randomness.
- The others are named for their position in this table.

References:

- For background and older facts see [?].
- For permutation, injective, and adapted see [?].
- For martingale processes see [?] and [?].
- For balanced strategies, see the upcoming paper of Tomislav Petrovic. These strategies are also mentioned in [?], independently of Petrovic.

Main idea of this talk

- Certain non-monotonic strategies can be used to characterize computable randomness.
- The main idea is that the strategy needs to know both
 - the bits it is betting on, and
 - the bits it is not betting on.
- This can be made formal by using “filtrations”.

- Certain transformations do preserve computable randomness.
- The main idea is that the map must choose both
 - the bits to use, and
 - the bits to not use.
- This can be made formal by using “measure-preserving transformations” and “factor maps”.

What is a martingale?

The probabilist's answer

- Fix a probability space (Ω, \mathcal{A}, P) .
- **Definition.** A **filtration** $\mathcal{F} = \{\mathcal{F}_n\}$ is a sequence of σ -algebras such that $\mathcal{F}_n \subseteq \mathcal{F}_{n+1} \subseteq \mathcal{A}$ for all n .
- Each \mathcal{F}_n represents the **information** known at time n .
- **Definition.** A **martingale** $M = \{M_n\}$ is a sequence of integrable functions $M_n : \Omega \rightarrow \mathbb{R}$ such that for each $n \in \mathbb{N}$,
 - M_n is \mathcal{F}_n -measurable, and
 - $\mathbb{E}[M_{n+1} \mid \mathcal{F}_n] = M_n$ a.s.
- A martingale represents the position of a process at time n .
- It is “fair” in that the expectation of the future is the present.

A translation between probability and computability

- Let (Ω, \mathcal{A}, P) be the fair-coin Borel probability measure on $2^{\mathbb{N}}$.
- Let $\mathcal{F} = \{\mathcal{F}_n\}$ be the filtration defined by $\mathcal{F}_n := \sigma\{[w] \mid w \in 2^n\}$.
This corresponds to information in the first n coin-flips.

- Given a martingale $d : 2^* \rightarrow \mathbb{R}$,

$$M_n(x) := d(x \upharpoonright n)$$

defines a martingale wrt the filtration \mathcal{F} .

- Given a martingale $M = \{M_n\}$ wrt the filtration \mathcal{F} ,

$$d(x \upharpoonright n) := M_n(x)$$

defines a well-defined “computability-theoretic” martingale (although it may not be nonnegative).

What is a martingale good for?

The probabilist's answer

Many things!

Used in

- Probability
- Finance
- Analysis
- Combinatorics
- Differential equations
- Dynamical Systems

Can be used to prove

- Lebesgue Differentiation Theorem
- Law of Large Numbers
- De Finetti's Theorem

What is known so far?

The probabilist's answer

A lot! ...but an important result is this:

■ Doob's martingale convergence theorem.

Let M be a martingale.

Assume $\sup_n \|M_n\|_{L^1} < \infty$ (M is L^1 -**bounded**).

Then M_n converges a.e. as $n \rightarrow \infty$.

■ Remarks.

- The L^1 -bounded property is important:
Consider a random walk on the integers.
- If M is nonnegative (as in computability theory), then

$$\sup_n \|M_n\|_{L^1} = \|M_0\|_{L^1} < \infty$$

and hence it is L^1 -bounded.

More on filtrations

- One kind of filtration is where \mathcal{F} is given by a sequence of increasingly fine partitions $\mathcal{P} = \{\mathcal{P}_n\}$.
- **Example.** In the case of coin-flipping, $\mathcal{P}_n = \{[w] \mid w \in 2^n\}$.
- In this case each M_n takes on finitely-many values.

- Every filtration \mathcal{F} has a limit σ -algebra $\mathcal{F}_\infty := \sigma(\bigcup_n \mathcal{F}_n)$.
- **Example.** In the case of coin-flipping, \mathcal{F}_∞ is the Borel σ -algebra on $2^{\mathbb{N}}$.

- Every martingale M has a minimal filtration \mathcal{F} where $\mathcal{F}_n := \sigma\{M_0, \dots, M_n\}$.
- So M is a martingale (wrt some filtration) if and only if

$$\mathbb{E}[M_{n+1} \mid M_0, \dots, M_n] = M_n.$$

σ -algebras

- Fix (Ω, \mathcal{A}, P) .
- We consider two sub- σ -algebras $\mathcal{F}, \mathcal{G} \subseteq \mathcal{A}$ to be **a.e. equivalent** if every set $A \in \mathcal{F}$ is a.e. equal to some set $B \in \mathcal{G}$, and vice-versa.
- A σ -algebra $\mathcal{F} \subseteq \mathcal{A}$ is can be represented (up to a.e. equivalence) in multiple ways.
 - 1 By a countable sequence of sets $\{A_0, A_1, \dots\}$ in \mathcal{F} , such that $\mathcal{F} = \sigma\{A_0, A_1, \dots\}$ a.e.
 - 2 By a continuous linear operator on L^1 (or L^2) given by $f \mapsto \mathbb{E}[f \mid \mathcal{F}]$ a.e.
 - 3 By a measure preserving map $T: (\Omega, \mathcal{A}, P) \rightarrow (\Omega', \mathcal{A}', P')$ (i.e. $P(T^{-1}(B)) = P'(B)$ for all $B \in \mathcal{A}'$), such that $\mathcal{F} = \sigma(T) := \sigma\{T^{-1}(A) \mid A \in \mathcal{A}'\}$. Call T a **factor map**.

Morphisms, isomorphisms, and factor maps

- A **morphism** $T: (\Omega, \mathcal{A}, P) \rightarrow (\Omega', \mathcal{A}', P')$ is a measure preserving map.
- An **isomorphism** is a pair of morphisms

$$T: (\Omega, \mathcal{A}, P) \rightarrow (\Omega', \mathcal{A}', P') \quad \text{and} \quad S: (\Omega', \mathcal{A}', P') \rightarrow (\Omega, \mathcal{A}, P)$$

such that

$$S \circ T = \text{id}_{\Omega} \quad (P\text{-a.e.}) \quad T \circ S = \text{id}_{\Omega'} \quad (P'\text{-a.e.})$$

- **Remark.** A morphism is the same as a factor map, but I am using the factor map to code a σ -algebra.
- **Remark.** With an isomorphism $T: (\Omega, \mathcal{A}, P) \rightarrow (\Omega', \mathcal{A}', P')$, the corresponding σ -algebra is just \mathcal{A} .

The main results

- Fix a **computable Polish space** Ω and a **computable Borel probability measure** P .
- **Theorem (R.)**. A point $x \in \Omega$ is **P -computably random** if and only if $\tilde{M}_n(x)$ is Cauchy as $n \rightarrow \infty$ for every (M, \mathcal{F}) where
 - 1 \mathcal{F} is a **computably enumerable filtration**,
 - 2 M is an **L^1 -computable martingale** wrt \mathcal{F} ,
 - 3 $\sup_n \|M_n\|_{L^1}$ is finite and computable, and
 - 4 \mathcal{F}_∞ is a **computable σ -algebra**.
- **Corollary (R.)**. Computable randomness is preserved by **effective factor maps** and **effectively measurable isomorphisms** (but not by **effectively measurable morphisms**).

Talk Outline

- 1 Define computable/effective versions of the following:
 - Borel probability measures
 - measurable functions, measurable sets, L^1 -functions
 - martingales
 - σ -algebras
 - morphisms, isomorphisms, factor maps
 - filtrations
- 2 Sketch the proof of the Main Theorem (on the fair-coin measure)
- 3 Sketch the proof of the Main Corollary.
- 4 Talk about related ideas and future work.

Computable Polish spaces and computable Borel probability measures

- **Definition.** A **computable Polish space** (or **computable metric space**) is a triple $(\Omega, \rho, \mathcal{S})$ such that
 - 1 ρ is a metric on Ω ,
 - 2 $\mathcal{S} = \{s_1, s_2, \dots\}$ is a countable dense subset of Ω , and
 - 3 $\rho(s_i, s_j)$ is computable uniformly from i, j for all $s_i, s_j \in \mathcal{S}$.

- **Definition.** A **computable Borel probability measure** P on Ω is a one such that the map $f \mapsto \mathbb{E}[f]$ is computable on bounded continuous functions.

- **Example.** If Ω is $2^{\mathbb{N}}$ then a Borel probability measure P is computable if and only if $P([w])$ is uniformly computable for all $w \in 2^*$.

Matrix of “computable” sets and functions

	Computable <i>Continuous</i>	A.E. Comp. <i>A.E. Cont.</i>	Nearly Comp. <i>measurable</i>	Eff. Meas. <i>meas. (mod 0)</i>
Set	decidable <i>clopen</i>	a.e. decid. μ -cont.	nearly decid. <i>measurable</i>	eff. meas. <i>meas. (mod 0)</i>
Function	computable <i>continuous</i>	a.e. comp. <i>a.e. cont.</i>	nearly comp. <i>measurable</i>	eff. meas. <i>meas. (mod 0)</i>
Integrable Function	computable ² <i>continuous</i> ³	a.e. comp. ^{2 4} <i>a.e. cont.</i> ^{3 4}	nearly comp. ² <i>measurable</i> ³	L^1 -comp. L^1 (mod 0)

²And the L^1 norm is computable.

³And integrable.

⁴For bounded functions on an interval, this is equivalent to being effectively Riemann integrable (and *Riemann integrable*).

Effectively measurable sets and maps

- Fix a computable Borel probability space (Ω, P) .
- Let Ω' be a computable Polish space with metric ρ .
- Consider the following (pseudo) metrics:

Borel sets A, B

$$d_1(A, B) = P(A \Delta B)$$

Integrable functions f, g

$$d_2(f, g) = \mathbb{E}[|f - g|]$$

Borel-meas. functions f, g

$$d_3(f, g) = \mathbb{E}[\min\{|f - g|, 1\}]$$

Borel-meas. maps $T, S: \Omega \rightarrow \Omega'$

$$d_4(T, S) = \mathbb{E}[\min\{\rho(T, S), 1\}]$$

- **Definition.** Define
 - effectively measurable sets
 - L^1 -computable functions
 - effectively measurable functions
 - effectively measurable maps

as those effectively approximable in the corresponding metric.

Useful facts about effectively measurable maps

- Effectively measurable objects are only defined up to P -a.e. equiv.
- Some set A is eff. measurable iff $\mathbf{1}_A$ is eff. measurable.
- Some function f is L^1 -computable iff f is effectively measurable and the L^1 -norm of f is computable.
- For every effectively measurable map $T: (\Omega, P) \rightarrow \Omega'$, there is a unique computable measure Q on Ω' (the distribution measure or the push forward measure) such that $T: (\Omega, P) \rightarrow (\Omega', Q)$ is measure-preserving.
- Further, the map $B \mapsto T^{-1}(B)$ is a computable map from Q -effectively measurable sets to P -effectively measurable sets.

Nearly computable sets and functions

- Say a function \tilde{f} is **nearly computable** if for each $\varepsilon > 0$, one can effectively find a computable function f_ε such that

$$P \left\{ x \mid \tilde{f}(x) = f_\varepsilon(x) \right\} \geq 1 - \varepsilon.$$

- Say a set \tilde{A} is **nearly decidable** if $\mathbf{1}_{\tilde{A}}$ is nearly computable.
- **Example.**

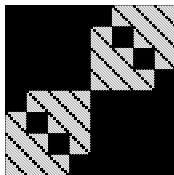


Figure : Nearly decidable set on $[0, 1]^2$.

Nearly computable sets and functions

- Nearly computable objects are defined pointwise, whereas effectively measurable objects are equivalence classes.
- Nearly computable functions are defined on Schnorr randoms.
- Nearly computable functions have been studied elsewhere.
 - Representative functions (MLR) of Pathak [?].
 - Representative functions (SR) of Pathak, Rojas, and Simpson [?].
 - Layerwise computable functions of Hoyrup and Rojas [?].
 - Schnorr layerwise computable functions of Miyabe [?].
 - Implicit in the work of Yu [?] on reverse mathematics.
 - Similar ideas are found in Edalat [?] on computable analysis.

Littlewood's Three Principles

For nearly computable structures

- **Principle 1.** Given an effectively measurable set A , there is a unique (up to Schnorr randomness) nearly decidable set \tilde{A} such that $\tilde{A} = A$ a.e.
- **Principle 2.** Given an effectively measurable map f , there is a unique (up to Schnorr randomness) nearly computable map \tilde{f} such that $\tilde{f} = f$ a.e.
- **Principle 3.** Given a computable sequence of effectively measurable functions (f_n) which are effectively a.e. Cauchy, then
 - the limit g is effectively measurable, and
 - $\tilde{f}_n(x) \rightarrow \tilde{g}(x)$ on Schnorr randoms x .

L^1 -computable martingales

- **Definition.** Take a martingale M (wrt some filtration). We say M is an L^1 -**computable martingale** if $M = (M_n)$ is a computable sequence of L^1 -computable functions.
- Computability theoretic martingales are L^1 -computable.
- Non-monotonic (adapted) martingales and martingale processes are L^1 -computable.
- For an L^1 -computable martingale, we have that $\tilde{M}_n(x)$ is well-defined on Schnorr randoms x and hence on computable randoms.

Computable σ -algebras

- Let \mathcal{A} be a sub- σ -algebra of the Borel sets.
- Say \mathcal{A} is a **computable σ -algebra** if the operator $f \mapsto \mathbb{E}[f \mid \mathcal{A}]$ is a computable operator in L^1 .
- (This is the same as saying it is a computable operator in L^2 .)
- Say that \mathcal{A} is a **lower semicomputable σ -algebra** if there is an enumeration of effectively measurable sets B_0, B_1, \dots which generates \mathcal{A} a.e.
- **Lemma.** All computable σ -algebras are lower semicomputable.
- **Lemma.** If f is an effectively measurable map, then $\sigma(f)$ is lower-semicomputable.

Computable morphisms and isomorphisms

- Take a measure preserving map $T: (\Omega, P) \rightarrow (\Omega', P')$.
- Say that T is an **effectively measurable morphism** if T is effectively measurable.
- Say that T is an **effective factor map** if T is effectively measurable and the factor σ -algebra $\sigma(T)$ is computable.
- Say T is an **effectively measurable isomorphism** if T is effectively measurable and has an effectively measurable inverse.

- All effectively measurable isomorphisms are effective factor maps.

Computable partitions and filtrations

- Say a filtration \mathcal{F} is **computable** (resp. **lower semicomputable**) if it is a computable sequence of computable (resp. lower semicomputable) σ -algebras \mathcal{F}_n .
- The limit F_∞ of a computable filtration is lower-semicomputable.
- If M is a computable martingale (wrt some filtration), then the minimal filtration $\mathcal{F}_n = \sigma(M_0, \dots, M_n)$ is lower semi-computable.
- Say a finite partition $\mathcal{P} = \{A_0, \dots, A_{n-1}\}$ of the space is **computable** if each set A_i is effectively measurable.
- A computable partition generates a computable σ -algebra.
- A computable sequence of partitions gives a **computable partition filtration**.
- The coin-flipping filtration is a computable filtration partition.

The main results (restated)

- Fix a computable Polish space Ω and a computable Borel probability measure.

- **Theorem (R.).** A point $x \in \Omega$ is P -computably random if and only if $\tilde{M}_n(x)$ is Cauchy as $n \rightarrow \infty$ for every (M, \mathcal{F}) where
 - 1 \mathcal{F} is a computably enumerable filtration,
 - 2 M is an L^1 -computable martingale wrt \mathcal{F} ,
 - 3 $\sup_n \|M_n\|_{L^1}$ is finite and computable, and
 - 4 \mathcal{F}_∞ is a computable σ -algebra.

- **Corollary (R.).** Computable randomness is preserved by effective factor maps and effectively measurable isomorphisms (but not by effectively measurable morphisms).

Proof of the Main Theorem

- I will prove the main theorem when
 - $\Omega = 2^{\mathbb{N}}$ and
 - P is the fair-coin measure.
- The proof is the same for other computable probability spaces.
- The definition of (P, Ω) -computable randomness is mentioned along the way.

Step 1: Four simplifying assumptions

- Fix a filtration \mathcal{F} such that
 - 1 \mathcal{F} is a **computable partition filtration**.
 - 2 \mathcal{F}_∞ is the **the Borel σ -algebra**.
- **Lemma (R.)**. A point $x \in 2^{\mathbb{N}}$ is computably random iff

$$\sup_{n \rightarrow \infty} \tilde{M}_n(x) < \infty$$

for every **nonnegative**, L^1 -computable martingale M wrt \mathcal{F} .

- **Proof Sketch.**
 - Note the lemma is true when \mathcal{F} is the coin-flipping filtration.
 - Assume \mathcal{F} is a different filtration.
 - Then “move” M to a martingale on the coin-flipping filtration which succeeds on the same points.
- **Remark.** This lemma be used as the definition of computable randomness for any computable probability space (Ω, P) (after showing it is invariant under the choice of filtration).

Step 2: Add in unused information to \mathcal{F}

- Let f_0, f_1, \dots be a computable dense sequence of L^1 -computable functions.
- Let $g_n := f_n - \mathbb{E}[f_n \mid \mathcal{F}_\infty]$ for each n . (Note \mathcal{F}_∞ is computable.)
- Let $\mathcal{G} := \sigma\{g_0, g_1, g_2, \dots\}$.
- \mathcal{G} is all the information independent of \mathcal{F} .

- Let $\mathcal{F}'_n := \sigma(\mathcal{G} \cup \mathcal{F}_n)$ for each n .
- \mathcal{F}' is still a lower-semicomputable filtration.

- \mathcal{G} and M_{n+1} are independent.
- Hence $E[M_{n+1} \mid \mathcal{F}'_n] = E[M_{n+1} \mid \mathcal{F}_n \cup \mathcal{G}] = E[M_{n+1} \mid \mathcal{F}_n] = M_n$.
- Hence M is still a martingale wrt \mathcal{F}' .

- $\mathcal{F}'_\infty = \sigma(\mathcal{F}_\infty \cup \mathcal{G})$, that is the Borel σ -algebra.

Step 3: Reduce \mathcal{F} to a partition filtration

- Let $\mathcal{F}_n = \sigma\{A_1^n, A_2^n, \dots\}$.
- **Levy 0-1 Law.** $\mathbb{E}[M_n \mid A_1^n, \dots, A_k^n] \xrightarrow{L^1} M_n$ as $k \rightarrow \infty$.
- Pick large enough k .
- Let $\mathcal{F}_n := \sigma\{A_1^n, \dots, A_k^n\}$.
- Let $M'_n := \mathbb{E}[M_n \mid \mathcal{F}'_n] = \mathbb{E}[M_n \mid A_1^n, \dots, A_k^n]$.
- Make sure these hold:
 - $\|M'_n - M_n\|_{L^1} < 2^{-n}$
 - $\mathcal{F}'_n \subseteq \mathcal{F}_n$ for each n .
 - $\mathcal{F}'_\infty = \mathcal{F}_\infty$.
- Then (M', \mathcal{F}') is still a martingale, and $\sup_n \|M'_n\|_{L^1} = \sup_n \|M_n\|_{L^1}$.
- Also $|\tilde{M}'_n(x) - \tilde{M}_n(x)| \rightarrow 0$ on Schnorr randoms.
- So $\tilde{M}_n(x)$ is Cauchy if and only if $\tilde{M}'_n(x)$ is Cauchy for all computable randoms x .

Step 4: Make M nonnegative.

- **Fact.** For every martingale M such that $\sup_n \|M_n\|_{L^1} < \infty$, there are two nonnegative martingales M^+ and M^- (wrt the same filtration as M), such that $M_n = M_n^+ - M_n^-$.

- **Lemma.** If the martingale (M, \mathcal{F}) satisfies the following:

1 $\sup_n \|M_n\|_{L^1}$ is finite and computable.

2 \mathcal{F} is computable.

then M^+ and M^- are L^1 -computable.

- Also $\tilde{M}_n(x) = \tilde{M}_n^+(x) - \tilde{M}_n^-(x)$ on Schnorr randoms x .
- Hence, if $\tilde{M}^+(x)$ and $\tilde{M}^-(x)$ are Cauchy, then $\tilde{M}(x)$ is Cauchy for computable randoms x .

Step 5: Use Doob's upcrossing trick

- Assume that $\tilde{M}_n(x)$ doesn't converge for some x .
- Then either $\sup \tilde{M}_n(x) = \infty$ or
- x "upcrosses" infinitely often between two rationals α, β .

- Use this information to create a new martingale M' which works as follows:
 - 1 Start betting as M would.
 - 2 If M goes above β , then stop betting until M goes below α .
 - 3 Then bet as M does again.
- "Buy low. Sell high."

- Then M' is a nonnegative martingale on the same filtration such that if $\sup_n \tilde{M}'_n(x) = \infty$.
- We've reduced our martingale to the case in Step 1. QED.

Step 5: Use Doob's upcrossing trick

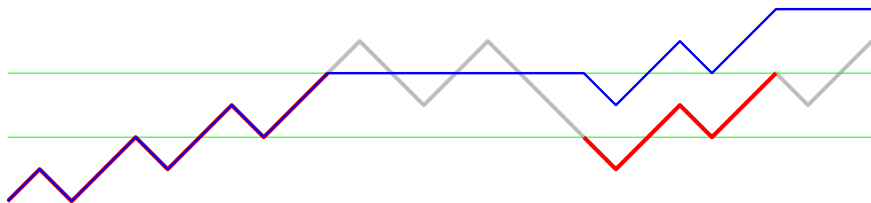


Figure : Upcrossings. Grey is the original martingale, red are the upcrossings, blue is the new martingale.

Quantitative result

- It is also possible to use Doob's upcrossing lemma and the same techniques to get explicit quantitative results.
- Let jumps_ε be the supremum of the number of times M_n "jumps" by ε .
- **Proposition.** (Under the same assumptions) for every $\varepsilon > 0$ there is some $N(\delta)$ and measure μ such that for all $\delta > 0$ and all measurable sets A ,

$$P\left(\left\{\text{jumps}_\varepsilon \geq N(\delta)\right\} \cap A\right) \leq \delta \cdot \mu(A)$$

- This result (which is effective), when combined with "bounded Martin-Löf tests", also proves the Main Theorem.

Proof of Corollary

- **Corollary (R.).** Computable randomness is preserved by effective factor maps
- **Proof.** Take an effective factor map $T: (2^{\mathbb{N}}, P) \rightarrow (2^{\mathbb{N}}, P')$.
- Assume $\tilde{T}(x)$ is not computably random. WTS x is not either.
- Take a computable martingale (M, \mathcal{F}) on $(2^{\mathbb{N}}, P')$ that satisfies the conditions of the theorem but \tilde{M}_n doesn't converge on $\tilde{T}(x)$.
- And the “pull-back” filtration $\mathcal{G} := \{T^{-1}(A) \mid A \in \mathcal{F}\}$.
- **Then, \mathcal{G}_∞ is computable since T is a effective factor map.**
- Take the “pull-back” martingale $N_n := M_n \circ T$.
- So $\tilde{N}_n(x) = \tilde{M}_n(T(x))$ diverges.
- So (N, \mathcal{G}) satisfies all the same conditions of the Theorem.
- So $\tilde{N}_n(x)$ should not converge since x is computably random.

Martingales and randomness

Randomness notions can be characterized by L^1 -computable martingales wrt a lower semicomputable filtration as follows.

	Condition on limit	Condition on bounds
Schnorr	– M_∞ is L^1 -computable. ¹	$\sup_n \ M_n\ _{L^2}$ is computable. $\sup_n \ M_n\ _{L^1}$ is computable.
Computable	\mathcal{F}_∞ is computable.	$\sup_n \ M_n\ _{L^1}$ is computable.
Martin-Löf²	– –	$\sup_n \ M_n\ _{L^1}$ is computable. $\sup_n \ M_n\ _{L^1}$ is finite.
Weak 2	M_∞ exists.	–

¹ M_∞ is the pointwise limit of the martingale.

²One direction is due to Merkle, Mihalović, and Slaman [?]. The other is due to Takahashi [?] in the continuous case and by Ed Dean [personal comm.] in the measurable case.

Further directions

- Explore martingales and randomness further.
 - Reverse martingales (related to the Ergodic Theorem).
 - Continuous time martingales, Brownian motion, and stochastic calculus.
 - Martingales along nets.
- Develop more of computable measure theory
 - Compare:
 - conditioning (probability theory)
 - relative computation (computability theory)
 - Develop more reverse mathematics of measure theory.
- Apply to computability theory
 - Use analytic tools to reprove known randomness results.
 - E.g. van Lambalgen's theorem for Schnorr randomness.
 - Use analytic tools to prove new randomness results.
 - Explore "isomorphism degrees" and "morphism degrees".

References I



Laurent Bienvenu, Rupert Hölzl, Thorsten Kräling, and Wolfgang Merkle.

Separations of non-monotonic randomness notions.

J. Logic and Computation, 2010.



Rodney G. Downey and Denis R. Hirschfeldt.

Algorithmic randomness and complexity.

Theory and Applications of Computability. Springer, New York, 2010.



Abbas Edalat.

A computable approach to measure and integration theory.

Inform. and Comput., 207(5):642–659, 2009.

References II



John M. Hitchcock and Jack H. Lutz.

Why computational complexity requires stricter martingales.
Theory Comput. Syst., 39(2):277–296, 2006.



Mathieu Hoyrup and Cristóbal Rojas.

An application of Martin-Löf randomness to effective probability theory.

In *Mathematical theory and computational practice*, volume 5635 of *Lecture Notes in Comput. Sci.*, pages 260–269. Springer, Berlin, 2009.



Wolfgang Merkle, Nenad Mihailović, and Theodore A. Slaman.

Some results on effective randomness.
Theory Comput. Syst., 39(5):707–721, 2006.

References III



Kenshi Miyabe.

L^1 -computability, layerwise computability and Solovay reducibility.

Submitted.



Noopur Pathak.

A computational aspect of the Lebesgue differentiation theorem.

J. Log. Anal., 1:Paper 9, 15, 2009.



Noopur Pathak, Cristóbal Rojas, and Stephen G. Simpson.

Schnorr randomness and the Lebesgue differentiation theorem.

Proceedings of the American Mathematical Society, To appear.

References IV



Jason Rute.

Algorithmic randomness, martingales, and differentiation I.

In preparation. Draft available at

math.cmu.edu/~jrute/preprints/RMD1_paper_draft.pdf.



Jason Rute.

Algorithmic randomness, martingales, and differentiation II.

In preparation.



Jason Rute.

Computable randomness and betting for computable probability spaces.

Submitted. [arXiv:1203.5535](https://arxiv.org/abs/1203.5535).



Jason Rute.

Transformations which preserve computable randomness.

In preparation.

References V



H. Takahashi.

Bayesian approach to a definition of random sequences with respect to parametric models.

In Information Theory Workshop, 2005 IEEE, pages 2180–2184. IEEE, 2005.



Xiaokang Yu.

Lebesgue convergence theorems and reverse mathematics.

Math. Logic Quart., 40(1):1–13, 1994.