Algorithmic Randomness via Probabilistic Algorithms

Sam Buss
Joint work with Mia Minnes

UC San Diego

The Constructive in Logic and Applications
Honoring Sergei Artemov's 60th Birthday
April 24, 2012

Motivation: Algorithmic Randomness

Algorithmic Randomness:
What does it mean for $X \in \{0,1\}^\infty$ to be algorithmically random?

Three classic paradigms, which often yield equivalent definitions:

- Unpredictability: *No effective betting strategy succeeds by betting on the bits of a random object.* [Schnorr '71]
- Typical-ness: *A random object avoids effective measure* 0 *sets.* [Levin'73, Schnorr'73]
- Incompressibility: (Kolmogorov Complexity) *Finite portions of a random object cannot be concisely described effectively.* [Martin-Löf '66]

Different notions of "effective" give rise to different notions of randomness.

We shall discuss only the *Unpredictability* paradigm.
This paradigm is the most closely tied to algorithms and betting strategies.

Betting strategies

Let $X \in \{0,1\}^\infty$. A betting strategy A satisfies:

- A sees the bits $X(i)$ of X sequentially,
- A decides how much to bet that the next bit of X is 0 or 1,
- For $\sigma \in \{0,1\}^*$ an initial segment of X, A's current winnings are given by a capital function $C = d(\sigma)$ where d is a martingale:

$$d(\lambda) \neq 0 \qquad \text{and} \qquad d(\sigma) = \frac{d(\sigma 0) + d(\sigma 1)}{2}.$$

- A succeeds against X if $\lim_n d(X \upharpoonright n) = \infty$.

The bets made by A are specified by a stake function $q = q(\sigma)$, such that $q \in [0,2]$ and means that A bets $\$(q-1)C$ that the next bit is 0.

Therefore, $q(\sigma) = d(\sigma 0)/d(\sigma)$: the new capital C after the bet becomes

$$C + (q-1)C = qC \qquad \text{if next bit is 0,}$$

$$C - (q-1)C = (2-q)C \qquad \text{if next bit is 1.}$$

Effective betting strategies and algorithmic randomness

X is ...

- Computable random if for each computable martingale d,

$$\lim_n d(X \upharpoonright n) \neq \infty.$$

- Partial computable random if for each partial computable martingale d,

$$\lim_n d(X \upharpoonright n) \neq \infty.$$

- Martin-Löf (ML) random if for each computably enumerable martingale d,

$$\lim_n d(X \upharpoonright n) \neq \infty.$$

Note: each limit can be replaced by limsup.

For computable and partial computable, the martingale is w.l.o.g. rational-valued.

A "c.e." function outputs a real value α by enumerating the rationals less than α.

Notions of algorithmic randomness

$$\boxed{\text{ML-random}}$$
$$\Downarrow \quad \not\Uparrow$$
$$\boxed{\text{partial computable random}}$$
$$\Downarrow \quad \not\Uparrow$$
$$\boxed{\text{computable random}}$$

Separations: [Nies, Stephan, Terwijn '05, Merkle '08, ...]

Schnorr's Critique

ML-randomness is a (the?) central notion in algorithmic randomness.

- Strongest of the natural notions of randomness based on effective computability.
- Elegant characterizations in all three paradigms.
- "Well-behaved" and tractable mathematical theory, including universal objects.

Schnorr's Critique

ML-randomness is a (the?) central notion in algorithmic randomness.

- Strongest of the natural notions of randomness based on effective computability.
- Elegant characterizations in all three paradigms.
- "Well-behaved" and tractable mathematical theory, including universal objects.

BUT

Schnorr's critique:

- ML-randomness is defined in terms of computably enumerable objects rather than computable ones.
- "Left c.e." property for a martingale is somewhat unnatural.

Schnorr's Critique

ML-randomness is a (the?) central notion in algorithmic randomness.

- Strongest of the natural notions of randomness based on effective computability.
- Elegant characterizations in all three paradigms.
- "Well-behaved" and tractable mathematical theory, including universal objects.

BUT

Schnorr's critique:

- ML-randomness is defined in terms of computably enumerable objects rather than computable ones.
- "Left c.e." property for a martingale is somewhat unnatural.

Goal: Give a computable characterization of ML-randomness...

Probabilistic strategies [B.-Minnes '12]

A probabilistic betting strategy A does the following at each step:

- Computes a probability p of betting
- Computes stake value q for bet (if one is placed)
- Bets on the next bit of X with probability p, or passes ("waits") with probability $1 - p$.

If the algorithm does not bet (passes), then the same bit of X remains available to be bet upon in the next step.

Finite initial segment of a betting game is

$\sigma \in \{0,1\}^*$ - the bits of X seen — and bet upon — so far,

and

$\pi \in \{\mathrm{b}, \mathrm{w}\}^*$ - the history of bet (b) vs. wait (w) moves.

A probabilistic strategy A is specified by two total computable rational-valued functions p_A and q_A:

$$p = p_A(\pi, \sigma) \qquad \text{and} \qquad q = q_A(\pi, \sigma).$$

Probabilistic strategies

The capital at node π after seeing σ is

- $C_A(\lambda, \lambda) = 1$;
- $C_A(\pi w, \sigma) = C(\pi, \sigma)$;
- $C_A(\pi \mathrm{b}, \sigma 0) = C_A(\pi, \sigma) q_A(\pi, \sigma)$;
  $C_A(\pi \mathrm{b}, \sigma 1) = C_A(\pi, \sigma)(2 - q_A(\pi, \sigma))$.

The probability of reaching node π when playing against σ is

- $P_A(\lambda, \lambda) = 1$;
- $P_A(\pi \mathrm{w}, \sigma) = P_A(\pi, \sigma)(1 - p_A(\pi, \sigma))$;
- $P_A(\pi \mathrm{b}, \sigma i) = P_A(\pi, \sigma) p_A(\pi, \sigma)$.

For a fixed $X \in \{0, 1\}^\infty$, P_A defines a measure μ_A^X on the space of possible bet/wait plays, $\{\mathrm{b}, \mathrm{w}\}^\infty$, defined by

$$\mu_A^X([\pi]) = P_A^X(\pi) := P_A(\pi, X \upharpoonright n)), \text{ where } n = |\pi|_\mathrm{b} \ = \ \#\text{b's in } \pi.$$

How to define success for probabilistic strategy?

The outcome of a probabilistic strategy on X is random, depending on the bet / wait choices. Success can be defined as either success with probability one (P1) or success in expectation (Ex):

Def. A is a successful P1-strategy for X if the set of $\Pi \in \{\mathrm{b}, \mathrm{w}\}^\infty$ s.t.

$$\lim_n C_A^X(\Pi \upharpoonright n) = \infty$$

has μ_A^X-measure one.

Def. A is a successful Ex-strategy for X if

$$\lim_n \mathrm{Ex}_A^X(n) = \infty$$

where $\mathrm{Ex}_A^X(n)$ is the expected capital after n-th bet.

- $\mathrm{Ex}_A^X(n) = \sum_{\pi \in R(n)} P_A^X(\pi) C_A^X(\pi)$,
- $R(n) = \{\pi : \pi = \pi'\mathrm{b},\ |\pi|_\mathrm{b} = n\}$.

How to define success?

X is ...

- P1-random if there is no successful P1-strategy for X.
- Ex-random if there is no successful Ex-strategy for X.

We can also require that the strategy must eventually bet:

X is ...

- Weak P1- or Weak Ex-random if no computable probabilistic strategy which $\boxed{\text{always eventually bets with probability one}}$ is a successful P1-strategy (resp. Ex-strategy) for X.
- Locally weak Ex-random if no computable probabilistic strategy which $\boxed{\text{eventually bets on } X \text{ with probability one}}$ is a successful Ex-strategy for X.

How to define success?

X is ...

- P1-random if there is no successful P1-strategy for X.
- Ex-random if there is no successful Ex-strategy for X.

We can also require that the strategy must eventually bet:

X is ...

- Weak P1- or Weak Ex-random if no computable probabilistic strategy which $\boxed{\text{always eventually bets with probability one}}$ is a successful P1-strategy (resp. Ex-strategy) for X.
- Locally weak Ex-random if no computable probabilistic strategy which $\boxed{\text{eventually bets on } X \text{ with probability one}}$ is a successful Ex-strategy for X.

New characterizations of algorithmic randomness

$$\boxed{\text{ML-random}}$$
$$\Downarrow \quad \not\Uparrow$$
$$\boxed{\text{partial computable random}}$$
$$\Downarrow \quad \not\Uparrow$$
$$\boxed{\text{computable random}}$$

New characterizations of algorithmic randomness

$$\boxed{\text{ML-random} = \text{Ex-random}}$$

$$\Downarrow \qquad \not\Uparrow$$

$$\boxed{\text{partial computable random} = \text{P1-random}}$$

$$\Downarrow \qquad \not\Uparrow$$

$$\boxed{\text{computable random}}$$

Equalities: [B-Minnes '12]

New characterizations of algorithmic randomness

$$\begin{array}{c} \boxed{\text{ML-random} = \text{Ex-random}} \\ \Downarrow \quad \not\Uparrow \\ \boxed{\text{partial computable random} = \text{P1-random} = \text{locally weak Ex-random}} \\ \Downarrow \quad \not\Uparrow \\ \boxed{\text{computable random} = \text{weak P1-random} = \text{weak Ex-random}} \end{array}$$

All definitions are equivalent with $\limsup$ instead of $\lim$.

Equalities: [B-Minnes '12]

Remarks

- The crucial difference between computable randomness and partial computable randomness is that the strategy may stop betting with non-zero probability on inputs other than X.
- The crucial difference between ML-random and (partial) computably random is partly the expectation (Ex) versus probability one (P1) distinction, and but also partly that the strategy for ML-randomness has unknown probability of never betting.

New characterizations of algorithmic randomness

Replacing success probability one (P1) with success probability $\alpha > 0$ does not change the definitions in the (locally) weak cases:

Theorem [B-Minnes, i.p.]

A sequence X is partial computable random if and only if there is no locally-weak probabilistic strategy which is successful against X with probability $\alpha > 0$.

A sequence X is computable random if and only if there is no weak probabilistic strategy which is successful against X with probability $\alpha > 0$.

Proof intuition:

Given a betting strategy A that succeeds on X with probability $\alpha > 0$. W.l.o.g. A uses the "slow but surely savings trick" so that A never loses much of its capital.

Let $q_1 \approx q_2$ be rationals s.t. $q_1 < \alpha \leq q_2$.
Values $C_0 << C_1 << C_2 << \cdots$ will be chosen to be sufficiently large.

A P1 strategy B works as follows:

a. Initially $i = 0$ and C_0 is large enough so that the capital will exceed C_0 with probability $\leq q_2$.
b. B acts like A in choosing p and q values, using the stake value q when an unknown bit of X is available. At the same time, B simulates other possible plays of the betting game by A, dovetailing over all possible moves with the same number of bets.
c. Whenever fraction $\geq q_1$ of the simulated plays by A exceed capital C_i: B chooses one of these at random, "jumps to" that play of A, increments i, computes a new sufficiently large C_i, and returns to b.

Open Problems

- Understanding Ex-randomness. The current definition uses the number of bets ("b" moves) as a stopping criterion to define successive capital values for the increasing expectation. Other natural definitions fail dramatically and unexpectedly — at least in the $\limsup$ case.
  Open: Does the "lim" definition of Ex-random remain equivalent with more general stopping criteria?

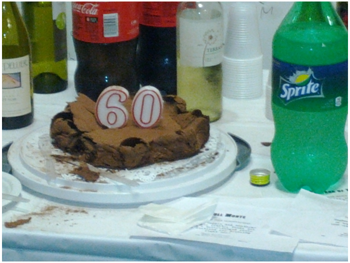
- Kolmogorov-Loveland (KL) randomness is defined by non-monotonic betting strategies, which can bet on bits of X out of sequential order. It is known that ML randomness implies KL randomness. A major open question is whether the notions coincide.

  ML random $\Rightarrow$ KL random $\Rightarrow$ Partial computable random

  Open: What is the strength of a non-monotonic betting strategies under the P1 definition of success? This defines a class of random reals that lies between KL random and ML-random. Is it equal to either of these?

The End

Thank you!



S. Buss, M. Minnes, "Probabilistic Algorithmic Randomness", preprint, 2012.