Probabilistic vs Deterministic Gamblers

Laurent Bienvenu
LaBRI, CNRS & Université de Bordeaux, France

Valentino Delle Rose
Dipartimento di Ingegneria Informatica e Scienze Matematiche, University of Siena, Italy

Tomasz Steifer
Institute of Fundamental Technological Research, Polish Academy of Sciences, Warszawa, Poland

Abstract

Can a probabilistic gambler get arbitrarily rich when all deterministic gamblers fail? We study this problem in the context of algorithmic randomness, introducing a new notion – almost everywhere computable randomness. A binary sequence X is a.e. computably random if there is no probabilistic computable strategy which is total and succeeds on X for positive measure of oracles. Using the fireworks technique we construct a sequence which is partial computably random but not a.e. computably random. We also prove the separation between a.e. computable randomness and partial computable randomness, which happens exactly in the uniformly almost everywhere dominating Turing degrees.

2012 ACM Subject Classification Theory of computation → Computability

Keywords and phrases Algorithmic randomness, Martingales, Probabilistic computation, Almost everywhere domination

Digital Object Identifier 10.4230/LIPIcs.STACS.2022.11

Funding *Tomasz Steifer*: Supported by the National Science Centre Poland grant no. 2018/31/B/HS1/04018 and the 2021 French Government Scholarship.

1 Introduction

What does it mean for an infinite binary sequence X to be random? This may seem like a strange question at first since in classical probability theory, any infinite binary sequence drawn at random (with respect to the uniform distribution) has probability 0 to occur. Yet, the theory of algorithmic randomness gives us a way answer it from a computability perspective: X is random if it does not possess any property of measure 0 which can be computably tested. There are many ways to formalize this, and hence many possible definitions of random sequence. One of the main approaches is the so-called unpredictability paradigm. We may say that a sequence X is unpredictable if no computable gambling strategy (or martingale) betting on the values of the bits of X and being rewarded fairly for its predictions can become arbitrarily rich during the course of the (infinite) game. The main two notions of randomness derived from this point of view are computable randomness and partial computable randomness, depending on whether we allow total computable or partial computable martingales. But in either case, the martingales considered are deterministic.

In this paper, we ask: do we get a stronger notion of randomness if we ask that X defeats not just all deterministically computable martingales, but also all probabilistically computable martingales? Usually, in computability theory, allowing probabilistic computations does not make a difference. This is in large part due to the foundational result that if a set $A \subset \mathbb{N}$ (or function $f : \mathbb{N} \to \mathbb{N}$, etc.) can be obtained by a probabilistic computation with positive probability, then it can in fact be obtained via a deterministic computation [5]. Yet this result is not necessarily an obstacle here as for a given X, different runs of the probabilistic algorithm are allowed to produce different martingales, as long as with positive probability,

39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022).
Editors: Petra Berenbrink and Benjamin Monmege; Article No. 11; pp. 11:1–11:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the martingale output by the probabilistic algorithm defeats X. And indeed, the main result of our paper is that probabilistic martingales *do* in fact perform better than deterministic ones!

We should note that probabilistic martingales were already considered by Buss and Minnes [4]. However, the applicability of their results for our purpose is limited. In particular, they studied two cases: probabilistic martingales which are total almost surely and probabilistic martingales which may be partial but nevertheless almost surely succeed on a given sequence. It is fairly easy to show that these cases reduce to computable and partial computable martingales respectively. The results of this paper are different and require more involved proofs.

1.1 Notation

The set of all infinite binary sequences is denoted by $2^{\mathbb{N}}$, while the set of finite binary strings is $2^{<\mathbb{N}}$. The truncation of x to the first n bits is $x \upharpoonright n$, while length of a string σ is written by $|\sigma|$. We write $\tau \prec x$ when τ is a prefix of some x (which might be a sequence or a string). The empty string is denoted by ϵ, the concatenation of two strings σ and τ by $\sigma^\frown\tau$. We are working with the product topology on $2^{\mathbb{N}}$, i.e., the topology generated by cylinder sets $[\sigma] = \{X \in 2^{\mathbb{N}} : \sigma \prec X\}$. This means that open sets are of the form $\bigcup_{\sigma \in A}[\sigma]$ where A is any set of strings. When A is computably enumerable (c.e.), the set $\bigcup_{\sigma \in A}[\sigma]$ is called *effectively open*. In this topology, the clopen sets are exactly the finite unions of cylinders.

We further equip $2^{\mathbb{N}}$ with the uniform measure μ, which is the measure where each bit of the sequence is equal to $1/2$ independently of the values of other bits. Formally, μ is the unique probability measure on the σ-algebra generated by cylinders for which $\mu([\sigma]) = 2^{-|\sigma|}$ for all σ.

As is common in computability theory, we sometimes identify sequences and strings with subsets of $\mathbb{N}$ (via characteristic function of the set) or paths in the full infinite binary tree. In particular, we say that σ is on the left of τ if σ is lesser than τ with respect to the lexicographical order.

1.2 Algorithmic randomness

Algorithmic randomness' goal is to assign a meaning to the notion of individual random string or sequence. While for strings we cannot reasonably hope for a clear separation between random and non-random (instead we have a quantitative measure of randomness: Kolmogorov complexity), for infinite binary sequences one can get such a separation. There are in fact many possible definitions. The most important one is called Martin-Löf randomness and is defined as follows. A set $\mathcal{N} \subset 2^{\mathbb{N}}$ is called effectively null if for every n one can cover it by an effectively open set of measure at most $\leq 2^{-n}$, uniformly in n.

▶ **Definition 1.** *A sequence $X \in 2^{\mathbb{N}}$ is called Martin-Löf random if it does not belong to any effectively null set.*

Said otherwise, X is Martin-Löf random if for every sequence $(\mathcal{U}_n)$ of uniformly effectively open sets such that $\mu(\mathcal{U}_n) \leq 2^{-n}$ for all n (such a sequence is known as a *Martin-Löf test*), we have $X \notin \bigcap_n \mathcal{U}_n$.

An effectively null set corresponds to an atypical (= measure 0) property which can in some sense be effectively tested and therefore, a Martin-Löf random sequence is one that withstands all computable statistical tests. The reason Martin-Löf's definition of randomness is considered to be the central one is that it is both well-behaved (Martin-Löf random sequences possess most properties one would expect from "random" sequences, including computability-theoretic properties) and robust, in that one can naturally get to the same notion by different approaches. For example, if we denote by K the prefix-free Kolmogorov complexity function (see for example [11]), then the Levin-Schnorr theorem states that a sequence X is Martin-Löf random if and only if $K(X \restriction n) \geq n - d$ for some d and all n. Informally, this means that Martin-Löf random sequences are exactly the "incompressible" ones.

As discussed above there is, however, another natural paradigm to define randomness (seemingly different from atypicality): unpredictability. We want to say that a sequence X is random if its bits cannot be guessed with better-than-average accuracy. This is formalized via the notion of martingale.

▶ **Definition 2** (martingale). *A function $d : 2^{<\mathbb{N}} \to \mathbb{R}^{>0}$ is called a martingale if for all $\sigma \in 2^{<\mathbb{N}}$:*

$$d(\sigma) = \frac{d(\sigma 0) + d(\sigma 1)}{2}.$$

A martingale d succeeds on a sequence X if

$$\limsup_{n\to\infty} d(X \restriction n) = \infty.$$

A martingale represents the outcome of a gambling strategy in a fair game where the gambler guesses bits one by one by betting some amount of money at each stage, doubling the stake if correct, losing the stake otherwise, debts not being allowed. The quantity $d(\sigma)$ represents the capital of the gambler after having seen σ. Usually in the literature martingales are allowed to take value 0 but not allowing it makes no difference for the definitions that follow and avoids some pathological cases later in the paper.

Armed with the notion of martingale, we can now formulate an important definition of "randomness", known as computable randomness.

▶ **Definition 3.** *A sequence $X \in 2^{\mathbb{N}}$ is called computably random if no computable martingale succeeds on X.*

In the above definition, we consider only martingales that are total computable. We would also like to allow partial computable martingales, but since they are not total functions in general, they are not even martingales in the above sense. To remedy this, one can simply define a partial martingale as a function d taking values in $\mathbb{R}^{>0}$ whose domain is contained in $2^{<\mathbb{N}}$ and closed under the prefix relation (if $d(\sigma)$ is defined, $d(\tau)$ is defined for every prefix τ of σ) and furthermore for every σ, $d(\sigma 0)$ is defined if and only if $d(\sigma 1)$ is defined and in case both are defined, the fairness condition $d(\sigma) = (d(\sigma 0) + d(\sigma 1))/2$ applies. Finally, success is defined in the same way as for martingales: we say that d succeeds on X if $d(X \restriction n)$ is defined for all n and $\limsup_{n\to\infty} d(X \restriction n) = \infty$. We can now get the following strengthening of computable randomness.

▶ **Definition 4.** *A sequence $X \in 2^{\mathbb{N}}$ is called partial computably random if no partial computable martingale succeeds on X.*

It is well-known that partial computable randomness is strictly stronger than computable randomness, but nonetheless strictly weaker than Martin-Löf randomness (see [11]).

Computable randomness and partial computable randomness are pretty robust notions. For example, it makes no difference whether we define success as achieving unbounded capital or as having a capital that tends to infinity.

▶ **Lemma 5** (folklore, see [7]). *For every total (resp. partial) computable martingale d there exists a (resp. partial) computable martingale d' such that d and d' succeed on exactly the same sequences and for every $A \in 2^{\mathbb{N}}$ we have $\limsup_{n\to\infty} d(A \upharpoonright n) = \infty$ iff $\lim_{n\to\infty} d'(A \upharpoonright n) = \infty$. Moreover, an index for d' can be found effectively from an index for d.*

Another important fact is that instead of considering computable real-valued martingales, we can restrict ourselves to rational valued martingales that are computable as functions from $2^{<\mathbb{N}}$ to $\mathbb{Q}$ (which we sometimes refer to as *exactly computable* martingales).

▶ **Lemma 6** (Exact Computation lemma, see [14]). *For every total (resp. partial) computable martingale d, there exists a total (resp. partial) exactly computable martingale d' such that d' succeeds on every sequence on which d succeeds. Moreover, an index for d' can be effectively obtained from an index for d.*

1.3 Probabilistic martingales

The above definitions assume computable martingales (partial or total) are deterministic. Our goal is to understand whether probabilistic martingales (i.e., obtained by a probabilistic algorithm) can do better. Usually, to capture the idea of probabilistic algorithm, one appeals to probabilistic models of computation, such as probabilistic Turing machines. However, from a computability-theoretic perspective, where relativization to an oracle is a bread-and-butter object of study, it is equivalent to assume that an infinite sequence of random bits is drawn in advance and given as oracle to a deterministic Turing machine which then uses it as a source of randomness. Thus, we will consider *partial computable oracle martingales*, that is, Turing functionals d where for every oracle Y, d^Y (the function computed by the functional with Y given as oracle) is a partial martingale.

▶ **Definition 7.** *A sequence $X \in 2^{\mathbb{N}}$ is called a.e. computably random if for every partial computable oracle martingale d the set of oracles Y such that d^Y is a total martingale and succeeds on X has measure zero, i.e.*

$$\mu\left(\left\{Y \in 2^{\mathbb{N}} : d^Y \textit{ is total and } \limsup_{n\to\infty} d^Y(X \upharpoonright n) = \infty\right\}\right) = 0.$$

X is said to be a.e. partial computably random if for every partial computable oracle martingale d the set of oracles Y such that d^Y succeeds on X has measure zero.

Note that we could have equivalently defined a.e. (partial) computably randomness directly from the relativization of (partial) computable randomness: a sequence X is a.e. (partial) computably random if for almost every Y, X is (partial) computably random relative to Y.

The informal question "do probabilistic gamblers perform better than deterministic ones" can now be fully formalized by the following two questions:

- Is a.e. computable randomness equal to computable randomness?
- Is a.e. partial computable randomness equal to partial computable randomness?

In [4], Buss and Minnes studied a restricted version of this problem. They considered a model of probabilistic martingales where one further requires $d^Y(\sigma)$ to be defined for all σ and almost all Y. This is a strong restriction which allows one to use an averaging technique. If d is a probabilistic martingale with this property, it is easy to prove that the average D defined by $D(\sigma) = \int_Y d^Y(\sigma)$ is a computable martingale. If X is computably random, D fails against X, that is, there is a constant c such that $D(X \upharpoonright n) < c$ for all n. Moreover, by Fatou's lemma:

$$\int_Y \liminf_n d^Y(X \upharpoonright n) \leq \liminf_n D(X \upharpoonright n) < c \qquad (\star)$$

which in turn implies that the set $\{Y : \liminf_n d^Y(X \upharpoonright n) = \infty\}$ has measure 0. In other words, the set of Y such that d^Y strongly succeeds against X has measure 0. By Lemma 5, this means that if a sequence X is computably random if and only if for every probabilistic martingale with the Buss-Minnes condition, d fails on X with probability 1.

Our main result is that, in the general case, we no longer have an equivalence of the two models: probabilistic martingales are indeed stronger than deterministic ones.

▶ **Theorem 8.** *There exist a sequence X which is partial computably random but not a.e. partial computably random and indeed not even a.e. computably random.*

We will devote the next sections to proving Theorem 8, but let us say a few words on why we believe it to be an interesting result. First of all, it is in stark contrast with Buss and Minnes' result that probabilistic martingales do not do any better than deterministic ones when they are required to be total with probability 1: in the general case, probabilistic martingales do better! Second, this is to our knowledge the first result of this kind in algorithmic randomness. If we were to define a.e. Martin-Löf randomness following the same idea (i.e., saying that X is a.e. Martin-Löf random if for almost all Y, X is Martin-Löf random relative to oracle Y), we would not get anything new, because a.e. Martin-Löf randomness coincides with Martin-Löf randomness. This is a direct consequence of the famous van Lambalgen theorem [15], which states that for every $A, B \in 2^{\mathbb{N}}$, the join $A \oplus B = A(0)B(0)A(1)B(1)\dots$ is Martin-Löf random if and only if A is Martin-Löf random and B is Martin-Löf relative to A, if and only if B is Martin-Löf random and A is Martin-Löf random relative to B. Now, let X be Martin-Löf random. For almost all Y, Y is Martin-Löf random relative to X (this is simply the fact that the set of Martin-Löf random sequences has measure 1, relativized to X), thus $X \oplus Y$ is Martin-Löf random, and thus X is Martin-Löf random relative to Y. This shows that X is a.e. Martin-Löf random. We see that van Lambalgen's theorem is key in this argument (we use it three times!). It was already known that the analogue of van Lambalgen for computable randomness fails [16], but Theorem 8 shows that it fails in a very strong sense.

Let us also remark that van Lambalgen's theorem shows that Martin-Löf randomness implies a.e. (partial) computable randomness: if X is Martin-Löf random, it is also Martin-Löf random relative to Y for almost every Y, and thus also (partial) computably random relative to Y for almost every Y.

2 Turing degrees of a.e.CR sequences

Before moving to the proof of Theorem 8, we give a simple degree-theoretic proof of a weaker result, namely a separation between computable randomness and a.e. computable randomness.

Recall that every Martin-Löf random sequence is computably random but a computable random sequence is not necessarily Martin-Löf random.This separation has some interesting connections with classical computability theory, as witnessed by the following theorem (recall that a sequence Y has *high Turing degree*, or simply *is high* if it computes some function $F : \mathbb{N} \to \mathbb{N}$ such that for every total computable function f, $f(n) \leq F(n)$ for almost all n).

▶ **Theorem 9** (Nies, Stephan, Terwijn [12])**.** *Let $Y \in 2^{\mathbb{N}}$. If Y computes a sequence X such that X is computably random but not Martin-Löf random, then Y has high Turing degree. Conversely, if Y has high Turing degree, then it computes some X which is computably random but not Martin-Löf random.*

It turns out that one can get an exact analogue of this theorem for a.e. computable randomness by replacing highness with a stronger notion: almost everywhere domination. A sequence Y is said to have *almost everywhere dominating Turing degree, or a.e. dominating Turing degree* if it computes an almost everywhere dominating function F, that is, a function F such that for every Turing functional Γ and almost every Z, if Γ^Z is total, then $\Gamma^Z(n) \leq F(n)$ for almost all n. See [11] for a more complete presentation of the history of this notion, originally due to Dobrinen and Simpson [6].

▶ **Theorem 10.** *Let $Y \in 2^{\mathbb{N}}$. If Y computes a sequence X such that X is a.e. computably random but not Martin-Löf random, then Y has a.e. dominating Turing degree. Conversely, if Y has a.e. dominating Turing degree, then it computes some X which is a.e. computably random but not Martin-Löf random (in fact, it even computes some X which is a.e. computably random but not partial computably random).*

▶ Remark 11. Nies et al.'s theorem actually states a little more than what we wrote above, namely that the sequence X in the second part of the theorem can be chosen to be Turing equivalent to Y. The analogue theorem is also true for a.e. computable randomness and a.e. domination but the proof becomes substantially more technical (we would need to introduce techniques to encode information into a computably random sequence) for only a small gain.

Proof. Let us prove the first part of the theorem by its contrapositive. Let $X \in 2^{\mathbb{N}}$ whose degree is not almost everywhere dominating. Suppose also X is not Martin-Löf random, i.e., $X \in \bigcap_n \mathcal{U}_n$ for $(\mathcal{U}_n)_{n \in \mathbb{N}}$ a sequence of uniformly effectively open sets with $\mu(\mathcal{U}_n) \leq 2^{-n}$. Consider the function t^X defined by $t^X(n) := \min\{s \mid X \in \mathcal{U}_n[s]\}$. Since X does not have a.e. dominating degree, there must exist a functional Γ such that

$$\mu\{Z \mid \Gamma^Z \text{is total and } \exists^\infty n\ \Gamma^Z(n) > t^X(n)\} > 0.$$

When Γ^Z is total and $\Gamma^Z(n) > t^X(n)$ for infinitely many n, we have $X \in \mathcal{U}_n[\Gamma^Z(n)]$ for infinitely many n. Note that in that case $\mathcal{U}_n[\Gamma^Z(n)]$ is a clopen set which Z-uniformly computable in Z. It is well-known that this type of test characterizes Schnorr randomness (a notion we will no discuss here but suffices to say that Schnorr randomness is weaker than computable randomness): a sequence X is Schnorr random if and only if for every computable sequence of clopen sets $\mathcal{D}_n$ such that $\mu(\mathcal{D}_n) \leq 2^{-n}$, X belongs to only finitely $\mathcal{D}_n$ (see for example [1, Lemma 1.5.9]). Relativized to Z, this fact shows that X is not Z-Schnorr random for a positive measure of Z's, thus not Z-computably random for a positive measure of Z's.

The strategy to prove the second part of the theorem is to take the function F computed by Y and use it as a time bound on oracle martingales in order to "totalize" them, which then allows us to use the averaging argument presented on page 4. In order for this to work, we must first prove that F can be assumed to be "simple" (in terms of Kolmogorov complexity).

▶ **Lemma 12.** *If Y has a.e. dominating Turing degree, it computes an a.e. dominating function F such that $K(F(n)) = O(\log n)$.*

Proof. Let $(\Phi_i)_{i \in \mathbb{N}}$ be an enumeration of all Turing functionals and consider the universal functional Ψ where $\Psi^{0^i 1 A} = \Phi_i^A$. It is easy to see that a function F is almost everywhere dominating if for almost all Z, either Ψ^Z is not total or $\Phi^Z(n) \leq F(n)$ for almost every n. For each Z, let $t^Z(n)$ be the minimum t, if it exists, such that $\Phi^Z(k)$ converges in time $\leq t$ for all $k \leq n$ and let $f^Z(n) = t^Z(n) + \max_{k \leq n} \Phi^Z(k)$.

Let Y be of a.e. dominating degree and $F \leq_T Y$ an almost everywhere dominating function.

For each n, let

$$\mathcal{U}_n = \{Z \mid f^Z(n)\downarrow < \infty\}$$

which is Σ^0_1 uniformly in n. We can write

$$\mathcal{U}_n = \bigcup_k \mathcal{U}_{n,k}$$

where

$$\mathcal{U}_{n,k} = \{Z \mid f^Z(n)\downarrow < k\}$$

and note that $\mathcal{U}_{n,k}$ is a clopen set, computable uniformly in n, k.

Since F is almost everywhere dominating, we have that for almost all Z and almost all n, either $f^Z(n)$ is undefined or $f^Z(n) \leq F(n)$. Said otherwise, the set

$$\mathcal{N}_0 = \limsup(\mathcal{U}_n \setminus \mathcal{U}_{n,F(n)})$$

is a nullset.

Now, for all n, let $a_n \in [0, n^2]$ be the largest integer that $\mu(\mathcal{U}_{n,F(n)}) \geq a_n/n^2$ and $F'(n)$ be the smallest k such that $\mu(\mathcal{U}_{n,k}) \geq a_n/n^2$. We see that $F'(n)$ is computable from F and furthermore,

$$K(F'(n)) \leq K(a_n) + O(1) \leq 2\log(n^2) + O(1) \leq 4\log n + O(1).$$

By definition, we have $\mu(\mathcal{U}_{n,F(n)}) \setminus \mathcal{U}_{n,F'(n)}) \leq 1/n^2$. By the Borel-Cantelli lemma,

$$\mathcal{N}_1 = \limsup(\mathcal{U}_{n,F(n)} \setminus \mathcal{U}_{n,F'(n)})$$

is a nullset. Thus, $\mathcal{N}_0 \cup \mathcal{N}_1$ is a nullset, which means that

$$\limsup(\mathcal{U}_n \setminus \mathcal{U}_{n,F'(n)})$$

is also a nullset, which in turn means that for almost all Z, for almost all n, if $f^Z(n)$ is defined, then $f^Z(n) \leq F'(n)$. By definition of f, a fortiori, for almost all Z, if Φ^Z is total, then $\Phi^Z(n) \leq F'(n)$ for almost all n. Thus the function F'

- is almost everywhere dominating
- is computable in F, hence computable in Y
- satisfies $K(F'(n)) = O(\log n)$

which finishes the proof of the lemma. ◀

As alluded to above, the function F is going to be used as a time bound. To see what we mean by this, consider a total (not necessarily computable) non-decreasing function $\psi : \mathbb{N} \to \mathbb{N}$. Let d be a (partial) exactly computable martingale. The time-bounded version of d with time bound ψ is the martingale d^ψ which mimics d but only allows it a time $\psi(n)$ to compute its bets on strings of length n. If d has not made a decision by this stage (either because it is in fact undefined, or because the time of computation is greater than $\psi(n)$)), the casino exclaims *"End of bets, nothing goes on the table!"* and the martingale is assumed to have placed an empty bet. Formally, $d^\psi(\epsilon) = d(\epsilon)$ and for any string σ and $b \in \{0, 1\}$:

$$d^\psi(\sigma b) = \begin{cases} d^\psi(\sigma) \cdot d(\sigma b)/d(\sigma) & \text{if both } d(\sigma 0)[\psi(n+1)]\downarrow \text{ and } d(\sigma 1)[\psi(n+1)]\downarrow \\ d^\psi(\sigma) \text{ otherwise.} \end{cases}$$

By definition d^ψ is always total, and when d is total, if the bound ψ dominates the convergence time of d (that is, for almost all σ, $d(\sigma)[\psi(|\sigma|)]\downarrow$), then d^ψ and d are within a multiplicative constant of one another, which in particular implies that d^ψ succeeds on the same sequences as d.

Now, let (d_i) be the effective enumeration of all (partial) exactly computable martingales with oracle. Without loss of generality, assume that d_i has a delay i imposed on it. Let F be the a.e dominating function as above. Let $\hat{d}$ be the oracle martingale defined by

$$\hat{d}^Z(\sigma) = \sum_i 2^{-i} d_i^{Z,F}(\sigma)$$

($d_i^{Z,F}$ is the time-bounded version of d_i^Z with time bound F).

It is a total martingale for every Z as all $d_i^{Z,F}$ are total martingales. Thus, its average D defined by

$$D(\sigma) = \int_Z \hat{d}^Z(\sigma)$$

is also a martingale.

Moreover, D is F- (exactly)computable. Indeed, because of the time bound F, the value of $d_i^{Z,F}(\sigma)$ only depends of the first $F(|\sigma|)$ bits of Z, and because of the delay on the d_i, only the martingales $(d_i)_{i\leq|\sigma|}$ matter in the computation of $D(\sigma)$. Thus the integral $\int_Z \hat{d}^Z(\sigma)$ is in fact a finite sum, can be computed from $F(|\sigma|)$, hence the F-computability of D. Even more precisely, the set of values $\{D(\sigma) \mid |\sigma| \leq n\}$ is computable from $F(n)$, and thus the Kolmogorov complexity of this set is at most $K(F(n)) + O(1) = O(\log n)$.

Let then X be the sequence which diagonalizes against D (that is, the sequence X constructed bit by bit where at each stage the chosen value of the next bit is the one that makes the martingale D lose money; all this will be detailed in the next section). Computing the first n bits of X only requires to know the set of values $\{D(\sigma) \mid |\sigma| \leq n\}$. Thus, we have established:

- $X \leq_T F$
- $K(X \upharpoonright n) \leq K(F(n)) + O(1) = O(\log n)$.

Since D does not succeed on X, by the exact same calculation as ($\star$) (see page 4), for almost all Z, $\hat{d}^Z$ does not succeed on X, and thus $d_i^{Z,F}$ does not succeed on X for any i.

But we also know, since F is a.e. dominating, for all i, for almost every Z, either d_i^Z is partial, or d_i^Z is total and its computation time is dominated by F, hence d^Z is within a multiplicative constant of $d^{Z,F}$.

Putting the two together, this entails that for almost all i and almost all Z, either d_i^Z is partial or it is total and does not succeed on X. In other words, X is a.e. computably random.

X has therefore all the desired properties:

- It is a.e. computably random,
- It is computable in F and thus computable in Y,
- $K(X \upharpoonright n) = O(\log n)$, ensuring that X is not only not Martin-Löf random, but not even partial computably random using a result of Merkle [10] (no partial computably random sequence can be of logarithmic complexity). ◀

An important result of Binns et al. [3] is that a.e. domination is strictly stronger than highness. This gives us the promised weaker version of Theorem 8.

▶ **Corollary 13.** *There exists a sequence X which is computably random but not a.e. computably random.*

Proof. Indeed, by Binn et al.'s result, take a high Turing degree $\mathbf{a}$ which is not a.e. dominating. By Theorem 9, there is an X in $\mathbf{a}$ which is computably random but not Martin-Löf random. By Theorem 10, X is not a.e. computably random either. ◀

3 The main construction

We now turn to the full proof of Theorem 8. We first recall the standard method to build a partial computably random sequence (see for example [11]). Next, we combine this construction with the so-called "fireworks" technique which can be viewed as a probabilistic forcing to see how to defeat, with probabilistic martingales, sequences that have been built using this construction.

3.1 Defeating finitely many martingales

Let us begin by explaining how to construct a partial computably random sequence. Let us first consider the simple case where we are trying to defeat a single martingale d, which we assume for the moment to be total computable, by making sure its capital does not go above a certain threshold. Up to multiplying d by a small rational, we may assume that that $d(\epsilon) < 1$. By induction, suppose we have already built $X \upharpoonright n$ in a way that $d(X \upharpoonright i) < 1$ for all $i \leq n$. By the fairness condition, either $d((X \upharpoonright n)^\frown 0) < 1$ or $d((X \upharpoonright n)^\frown 1) < 1$. If the former is true, we set $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 0$, otherwise we set $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 1$. Continuing in this fashion we ensure that the martingale d does not succeed against X as its never reaches 2. Observe that when the martingale d is exactly computable, the sequence X is computable (uniformly in a code for d).

Suppose now that we have a finite family of total martingales $d_1, \dots d_n$. If we want to diagonalize against all of them at the same time, one can simply find positive rationals $q_1, \dots, q_n$ such that $\sum_{i=1}^n q_i \cdot d_i(\epsilon) < 1$ and proceed as before against the martingale $\sum_{i=1}^n q_i \cdot d_i$. Again, the sequence X obtained by diagonalization against this finite family of martingales is computable uniformly in a code for the family of d_i's. But suppose now that some of the martingales in this family are partial instead of total. This does not cause much difficulty: having already built $X \upharpoonright n$, consider only the sub-family F of indices of martingales that are still defined on $(X \upharpoonright n)^\frown 0$ and $(X \upharpoonright n)^\frown 1$. The other martingales are undefined and thus will not succeed by fiat on the sequence X. Now, if $\sum_{i \in F} q_i \cdot d_i((X \upharpoonright n)^\frown 0) < 1$, set $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 0$, otherwise set $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 1$. Once again the sequence X defeats all of the d_i's, some of them because they become undefined at some stage, some of them because their capital never exceeds $1/q_i$. Moreover, X is still a computable sequence. It is not however computable uniformly in a code for the family of d_i's because one needs to specify which martingales become undefined in the construction and when (this is a finite amount of information but it cannot be uniformly computed) but this is not an obstacle for our purposes.

To summarize these preliminary considerations, we can make the following definition.

▶ **Definition 14.** *Let $(d_1, q_1), \dots (d_n, q_n)$ be a finite family where each d_i is a (code for) a partial computable martingale and q_i a positive rational. Let $\sigma \in 2^{<\mathbb{N}}$ such that, calling F the family of indices i such that $d_i(\sigma)$ converges, we have $\sum_{i \in F} q_i \cdot d_i(\sigma) < 1$. Consider the computable sequence X defined inductively by $X \upharpoonright |\sigma| = \sigma$ and if $X \upharpoonright n$ is already built, letting F_n be the family of indices such that $d_i((X \upharpoonright n)^\frown 0)$ converges, then $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 0$ if $\sum_{i \in F_n} q_i \cdot d_i(X \upharpoonright n)^\frown 0) < 1$ and $X \upharpoonright (n+1) = (X \upharpoonright n)^\frown 1$ otherwise. This sequence is called the* diagonalization against $(d_1, q_1), \dots, (d_n, q_n)$ above σ.

3.2 Defeating all partial computable martingales

When we have a countable family of martingales to diagonalize against, the standard way to proceed is to introduce them one by one during the game so that at any step we only have to diagonalize against a finite family as above. The delays between the introduction of martingales is flexible and therefore will be a parameter of the construction.

The diagonalizing sequence $\Delta((t_e)_{e\in N})$

Let $(d_i)_{i\in\mathbb{N}}$ be a standard enumeration of partial computable rational valued martingales. Let $(t_e)_{e\in\mathbb{N}}$ be a family of integers. The sequence $\Delta((t_e)_{e\in N})$ is constructed by finite extension as follows. Start with the empty string $\sigma_0 = \epsilon$ and recursively do the following. Having built σ_n, let q_{n+1} be a rational such that $\sum_{i\in F} q_i \cdot d_i(\sigma_n) < 1$ where F is the set of indices $i \in [1, n+1]$ such that $d_i(\sigma_n)$ converges. Let A be the diagonalization against $(d_1, q_1), \ldots, (d_{n+1}, q_{n+1})$ above σ_n. The sequence A is an extension of σ and is computable (see above), so let e be a code for it (say the smallest one). Define $\sigma_{n+1} = A \restriction (|\sigma_n| + t_e)$. Finally, set

$$\Delta((t_e)_{e\in N}) = \bigcup_n \sigma_n.$$

It is easy to check that $\Delta((t_e)_{e\in N})$ defeats all partial computable martingales. Moreover, the construction ensures the following important fact, which will be key for the rest of our proof:

Fact 1: For infinitely many e (namely, those codes that show up in the construction), the sequence $\Delta((t_e)_{e\in N})$ coincides with the computable sequence A of index e on a prefix of length $\geq t_e$.

3.3 Fireworks

Let $(\mathbb{P}, \leq)$ be a computable order, that is, each element $p \in \mathbb{P}$ can be encoded by an integer and for a given pair (n, m) of integers, it is decidable whether n and m are indeed codes for two elements of p and q in $\mathbb{P}$ and whether $p \leq q$. We say that a sequence $(p_i)_{i\in\mathbb{N}}$ of elements of $\mathbb{P}$ is $\mathbb{P}$-*generic* if $p_0 \geq p_1 \geq p_2 \geq \ldots$ and for every c.e. subset W of $\mathbb{P}$:

- either there exists an i such that $p_i \in W$
- or, there exists a j such that for any $q \leq p_j$, $q \notin W$

In particular, if W is dense (that is, for every $p \in \mathbb{P}$ there exists $q \leq p$ such that $q \in W$), then for every generic sequence $(p_i)_{i\in\mathbb{N}}$ there must be some i such that $p_i \in W$, in which case we say that $\mathbb{P}$ *meets* W.

For most computable orders of interest, there cannot exist a computable generic sequence. However, there is a way to probabilistically obtain one, using the so-called fireworks technique. This was first proven by Kurtz [8] who showed that one can probabilistically obtain a generic sequence when $\mathbb{P}$ is the set of strings and $\sigma \leq \tau$ when τ is a prefix of σ (Kurtz himself drew upon an argument of Martin [9] who had shown that one can probabilistically construct a hyperimmune set). The probabilistic nature of Kurtz's and Martin's arguments was somewhat hidden in their proof (they used a different framework sometimes referred to as "risking measure"). Rumyantsev and Shen [13] simplified Kurtz's presentation of this technique (although they only focused on Martin's result about hyperimmunity) by giving an explicit probabilistic algorithm. They illustrated their algorithm by a metaphor about a buyer who tries to buy fireworks in a shop, hence the name. Shen and Rumyantsev's presentation allowed Bienvenu and Patey [2, Section 1.4] to make the following generalization to any computable order.

▶ **Theorem 15** (Fireworks master theorem [2]). *For any computable order $\mathbb{P}$, there exists a Turing functional Φ with range $\mathbb{P}$ such that for a set of Z's of positive measure, we have that $\Phi^Z(i)$ is defined for all i and the sequence $(\Phi^Z(i))_{i\in\mathbb{N}}$ is generic.*

For our proof of Theorem 8, we are going to use the order $\mathbb{P}$ whose elements are finite approximations of martingales with positive rational values. Specifically, a member of $\mathbb{P}$ is a total function f whose domain is $\{0,1\}^{\leq n}$ for some n – which we call *length of* f and denote by $lh(f)$ – whose range is $\mathbb{Q}^{>0}$, such that $f(\epsilon) = 1$ and $f(\sigma) = (f(\sigma 0) + f(\sigma 1))/2$ for all σ of length $< lh(f)$. We say that $g \leq f$ if g is an extension of f (i.e., the domain of f is contained in the domain of g and the two coincide on the domain of f). It is clear that $(\mathbb{P}, \leq)$ is a computable order. It is also clear that if $f_1 \geq f_2 \geq \dots$ is a sequence of elements of $\mathbb{P}$ such that $lh(f_i)$ tends to $+\infty$, then $D = \bigcup f_i$ is a total rational valued martingale. This is in particular the case when $(f_i)_{i\in\mathbb{N}}$ is a $\mathbb{P}$-generic sequence, because for every n, the set of elements of $\mathbb{P}$ of length at least n is dense; in this case, we say that the martingale $D = \bigcup f_i$ is a *$\mathbb{P}$-generic martingale.*

▶ **Lemma 16.** *Let D be a $\mathbb{P}$-generic martingale. For every computable sequence A and integer k there exists s such that D reaches capital at least k while playing against the prefix of A of length s (that is, $D(A \restriction l) > k$ for some $l < s$).*

Proof. Fix a computable A and consider the set

$$W = \{g \in \mathbb{P} \mid (\exists l)\ g(A \restriction l) > k\}.$$

We claim that W is a dense c.e. subset of $\mathbb{P}$. That it is c.e. is clear. Now, take any $f \in \mathbb{P}$. Let $n = lh(f)$. By definition of $\mathbb{P}$, $f(A \restriction n)$ is positive, so we can pick an $m > n$ such that $2^{m-n} \cdot f(A \restriction n) > k$. Let g be the martingale of length m which behaves like f up to length n and after that stage plays the doubling strategy on A (and stops betting outside of A). Formally:

$$g(\tau) = \begin{cases} f(\tau) & \text{if } |\tau| \leq n \\ f(\tau \restriction n) & \text{if } |\tau| \geq n \text{ and } \tau \restriction n \neq A \restriction n \\ 0 & \text{if } \tau \restriction n = A \restriction n \text{ but } \tau \text{ is not a prefix of } A \\ f(A \restriction n) \cdot 2^{|\tau|-n} & \text{if } \tau \text{ is a prefix of } A. \end{cases}$$

It is easy to check that g is a finite approximation of martingale which extends f and by construction $g(A \restriction m) = 2^{m-n} \cdot f(A \restriction n) > k$. Thus W is indeed dense. ◀

We can now finish the proof of our main result.

Proof of Theorem 8. By Theorem 15 applied to our partial order $(\mathbb{P}, \leq)$, there is a Turing functional Φ and a set $\mathcal{G}$ of positive measure such that for every $Z \in \mathcal{G}$, $\Phi^Z(n)$ is a $\mathbb{P}$-generic sequence. Thus for $Z \in \mathcal{G}$, $D^Z = \bigcup_n \Phi^Z(n)$ is a $\mathbb{P}$-generic martingale.

Let A be a computable sequence and e be a code for A. By Lemma 16, for every $Z \in \mathcal{G}$, there exists some l_e^Z such that D^Z – being a $\mathbb{P}$-generic martingale – reaches capital at least e at some point while playing against the prefix $A \restriction l_e^Z$.

Now, for each e which is the code of a computable sequence choose some s_e large enough to have

$$\mu\{Z \in \mathcal{G} \mid l_e^Z \leq s_e\} \geq (1 - 2^{-e-1})\mu(\mathcal{G})$$

(and for e which is not a code for a computable sequence, choose s_e arbitrarily).

This guarantees that

$$\mu\{Z \in \mathcal{G} \mid (\forall e \text{ code for a computable seq.})\ l_e^Z \leq s_e\} \geq \mu(\mathcal{G})/2 > 0.$$

Let $\mathcal{H}$ be the set of the left-hand side of this inequality.

Let us consider the sequence $\Delta((s_e)_{e\in\mathbb{N}})$, which by construction is partial computably random. For every $Z \in \mathcal{H}$, for every computable sequence A of code e, the martingale D^Z reaches capital at least e on $A \upharpoonright s_e$. On the other hand, by Fact 1, we know that for infinitely many e, the sequence $\Delta((s_e)_{e\in N})$ coincides with the computable sequence A of index e on a prefix of length $\geq s_e$. Thus this guarantees that for $Z \in \mathcal{H}$, D^Z reaches capital at least e while playing on $\Delta((s_e)_{e\in\mathbb{N}})$. Thus $\Delta((s_e)_{e\in\mathbb{N}})$ is partial computably random but not almost everywhere computably random since $\mathcal{H}$ has positive measure. ◀

4 Conclusion and open questions

In this paper, we have compared the power of deterministic and probabilistic prediction. To this end, we have introduced two notions – a.e. partial computable randomness and a.e. computable randomness. In contrast with Buss and Minnes' results [4], where (due to the stronger limitations on the class of martingales considered) the authors obtained equivalent characterizations of partial computable and computable randomness in terms of probabilistic martingales, our notions do not correspond to their deterministic counterparts, but are, indeed, strictly stronger. The following diagram summarizes the mutual relationships between these notions.

$$\begin{tikzcd}
 & \text{PCR} \arrow[dr, phantom, sloped, "\subsetneq"] \arrow[dd, phantom, "\not\subseteq\quad\not\supseteq"] & \\
\text{a.e.PCR} \arrow[ur, phantom, sloped, "\subsetneq"] \arrow[dr, phantom, sloped, "\subsetneq"] & & \text{CR} \\
 & \text{a.e.CR} \arrow[ur, phantom, sloped, "\subsetneq"] &
\end{tikzcd}$$

The main results of this paper, in fact, concern the incomparability of the notions of a.e. computable randomness and partial computable randomness: on the one hand, by Theorem 8, partial computable randomness does not imply a.e. computable randomness; on the other hand, Theorem 10 states that every a.e. dominating degree computes (actually, contains) a sequence which is a.e. computably random but not partial computably random.

We conclude this paper by pointing out interesting further directions to be investigated on this topic.

The main goal we have achieved is the construction of a partial computable random sequence X which is not a.e. computably random: from the perspective of algorithmic randomness, this amounts to say that any sufficiently random sequence Z derandomizes X, in the sense that X is not computably random relative to Z. But how much randomness is actually needed to derandomize such a sequence? In particular, is Martin-Löf randomness enough? In this regard, we ask the following question.

▶ **Question 1.** *Given a partial computably random sequence X which is not a.e. computably random, can there be a Martin-Löf random sequence Z such that X is still computably random relative to Z? If so, is there always such a Z?*

The second open question is more general, and strongly related with one of the main theoretical motivations leading to this work, namely the failure of the analogue of van Lambalgen's theorem for computable randomness. Theorem 8, in fact, can be regarded as a strong failure of this result for computable randomness, because of the existence of computably random sequences that, nevertheless, can be derandomized by almost every oracle. It is known that the analogue of van Lambalgen's theorem fails for other randomness notions studied in the literature, such as Schnorr randomness, Kurtz randomness and Demuth randomness (see [7]). However, we do not know if it fails in the strong sense mentioned above.

▶ **Question 2.** *Are there other randomness notions for which an analogue of Theorem 8 holds (namely, for which there is a random sequence which is not a.e. random)?*

In particular, it seems that our constructions may be easily modified to get results about a.e. Schnorr randomness.

References

1 Laurent Bienvenu. *Game-theoretic characterizations of randomness: unpredictability and stochasticity.* PhD thesis, Université de Provence, 2008.

2 Laurent Bienvenu and Ludovic Patey. Diagonally non-computable functions and fireworks. *Information and Computation*, 253:64–77, 2017.

3 Stephen Binns, Bjørn Kjos-Hanssen, Manuel Lerman, and Reed Solomon. On a conjecture of Dobrinen and Simpson concerning almost everywhere domination. *Journal of Symbolic Logic*, 71(1):119–136, 2006.

4 Sam Buss and Mia Minnes. Probabilistic algorithmic randomness. *Journal of Symbolic Logic*, 78(2):579–601, 2013.

5 Karel de Leeuw, Edward F. Moore, Claude Shannon, and Norman Shapiro. Computability by probabilistic machines. In *Automata Studies.* Princeton University Press, 1956.

6 Natasha Dobrinen and Stephen G. Simpson. Almost everywhere domination. *Journal of Symbolic Logic*, 69(3):914–922, 2004.

7 Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity.* Theory and Applications of Computability. Springer New York, New York, NY, 2010.

8 Stuart Alan Kurtz. *Randomness and Genericity in the Degrees of Unsolvability.* PhD thesis, University of Illinois at Urbana–Champaign, 1982.

9 Donald Martin. Measure, category, and degrees of unsolvability. Unpublished manuscript, 1967.

10 Wolfgang Merkle. The complexity of stochastic sequences. *Journal of Computer and System Sciences*, 74(3):350–357, 2008.

11 André Nies. *Computability and randomness.* Oxford Logic Guides. Oxford University Press, 2009.

12 André Nies, Frank Stephan, and Sebastiaan Terwijn. Randomness, relativization and Turing degrees. *Journal of Symbolic Logic*, 70:515–535, 2005.

13 Andrei Rumyantsev and Alexander Shen. Probabilistic constructions of computable objects and a computable version of Lovász local lemma. *Fundamenta Informaticae*, 132(1):1–14, 2014.

14 Claus Schnorr. *Zufälligkeit und Wahrscheinlichkeit*, volume 218 of *Lecture Notes in Mathematics.* Springer-Verlag, Berlin-Heidelberg-New York, 1971.

15 Michiel van Lambalgen. *Random sequences.* PhD dissertation, University of Amsterdam, Amsterdam, 1987.

16 Liang Yu. When van Lambalgen's theorem fails. *Proceedings of the American Mathematical Society*, 135(3):861–864, 2007.