Separations of non-monotonic randomness notions

(Version submitted on July 19, 2010)

Laurent Bienvenu[1], Rupert Hölzl[2], Thorsten Kräling[2], and Wolfgang Merkle[2]

[1] LIAFA, CNRS & Université de Paris 7, France
[2] Institut für Informatik, Ruprecht-Karls-Universität, Heidelberg, Germany

Abstract. In the theory of algorithmic randomness, several notions of random sequence are defined via a game-theoretic approach, and the notions that received most attention are perhaps Martin-Löf randomness and computable randomness. The latter notion was introduced by Schnorr and is rather natural: an infinite binary sequence is computably random if no total computable strategy succeeds on it by betting on bits in order. However, computably random sequences can have properties that one may consider to be incompatible with being random, in particular, there are computably random sequences that are highly compressible. The concept of Martin-Löf randomness is much better behaved in this and other respects, on the other hand its definition in terms of martingales is considerably less natural.

Muchnik, elaborating on ideas of Kolmogorov and Loveland, refined Schnorr's model by also allowing non-monotonic strategies, i.e. strategies that do not bet on bits in order. The subsequent "non-monotonic" notion of randomness, now called Kolmogorov-Loveland-randomness, has been shown to be quite close to Martin-Löf randomness, but whether these two classes coincide remains a fundamental open question.

In order to get a better understanding of non-monotonic randomness notions, Miller and Nies introduced some interesting intermediate concepts, where one only allows non-adaptive strategies, i.e., strategies that can still bet non-monotonically, but such that the sequence of betting positions is known in advance (and computable). Recently, these notions were shown by Kastermans and Lempp to differ from Martin-Löf randomness. We continue the study of the non-monotonic randomness notions introduced by Miller and Nies and obtain results about the Kolmogorov complexities of initial segments that may and may not occur for such sequences, where these results then imply a complete classification of these randomness notions by order of strength.

1 Introduction

Random sequences are the central object of study in algorithmic randomness and have been investigated intensively over the last decade, which led to a wealth of interesting results clarifying the relations between the various notions of randomness and revealing interesting interactions with notions such as computational power [2, 5, 11].

Intuitively speaking, a binary sequence is random if the bits of the sequence do not have effectively detectable regularities. This idea can be formalized in terms of betting strategies, that is, a sequence will be called random in case the capital gained by successive bets on the bits of the sequence according to a fixed betting strategy must remain bounded, with fair payoff and a fixed set of admissible betting strategies understood.

The notions of random sequences that have received most attention are Martin-Löf randomness and computable randomness. Here a sequence is called computably random if no total computable betting strategy can achieve unbounded capital by betting on the bits of the sequence in the natural order, a definition that indeed is natural and suggests itself. However, computably random sequences may lack certain properties associated with the intuitive understanding of randomness, for example there are computably random sequences that are highly compressible, i.e., show a large amount of redundancy, see Theorem 4 below. Martin-Löf randomness behaves much better in this and other respects. Indeed, the Martin-Löf random sequences can be characterized as the sequences that are incompressible in the sense that all their initial segments have essentially maximal Kolmogorov complexity, and in fact this holds for several versions of Kolmogorov complexity according to celebrated results by Schnorr, by Levin and, recently, by Miller and Yu [2]. On the other hand, it has been held against the concept of Martin-Löf randomness that its definition involves effective approximations, i.e., a very powerful, hence rather unnatural model of computation. Indeed, the usual definition of Martin-Löf randomness in terms of left-computable martingales (that is, in terms of betting strategies where the gained capital can be effectively approximated from below) is not very intuitive.

It can be shown that Martin-Löf randomness (or ML-randomness for short) strictly implies computable randomness. According to the preceding discussion the latter notion is too inclusive while the former may be considered unnatural. Ideally, we would therefore like to find a more natural characterization of ML-randomness; or, if that is impossible, we are alternatively interested in a notion that is close in strength to ML-

randomness, but has a more natural definition. One promising way of achieving such a more natural characterization or definition could be to use computable betting strategies that are more powerful than those used to define computable randomness.

Muchnik [10] proposed to consider computable betting strategies that are non-monotonic in the sense that the bets on the bits need not be done in the natural order, but such that the bit to bet on next can be computed from the already scanned bits. The corresponding notion of randomness is called Kolmogorov-Loveland randomness because Kolmogorov and Loveland independently had proposed concepts of randomness defined via non-monotonic selecting of bits.

Kolmogorov-Loveland randomness has been proven to be in many respects much stronger than computable randomness (we will recall a few results supporting this claim throughout the paper). Kolmogorov-Loveland randomness is also implied by and in fact is quite close to Martin-Löf randomness, see Theorem 15 below, but whether the two notions are distinct is one of the major open problems of algorithmic randomness. In order to get a better understanding of this open problem and of non-monotonic randomness in general, Miller and Nies [9] introduced restricted variants of Kolmogorov-Loveland randomness, where the sequence of betting positions must be non-adaptive, i.e., can be computed in advance without knowing the sequence on which one bets.

The randomness notions mentioned so far are determined by two parameters that correspond to the columns and rows, respectively, of the table in Figure 1. First, the sequence of places that are scanned and on which bets may be placed, while always being given effectively, can just be monotonic, can be equal to $\pi(0), \pi(1), \ldots$ for a permutation or an injection π from $\mathbb{N}$ to $\mathbb{N}$, or can be adaptive, i.e., the next bit depends on the bits already scanned. Second, once the sequence of scanned bits is determined, betting on these bits can be according to a betting strategy where the corresponding martingale is total or partial computable, or is left-computable. The known inclusions between the corresponding classes of random sequences are shown in Figure 1, see Section 2 for technical details and for the definitions of the class acronyms that occur in the figure.

The classes in the last row of the table in Figure 1 all coincide with the class of Martin-Löf random sequences by the folklore result that left-computable martingales always yield the concept of Martin-Löf randomness, no matter whether the sequence of bits to bet on is monotonic or is determined

3

	monotonic		permutation		injection		adaptive
total	**TMR**	$=$	**TPR**	$\supseteq$	**TIR**	$\supseteq$	**KLR**
	$\cup\mathsf{I}$		$\cup\mathsf{I}$		$\cup\mathsf{I}$		$\parallel$
partial	**PMR**	$\supseteq$	**PPR**	$\supseteq$	**PIR**	$\supseteq$	**KLR**
	$\cup\mathsf{I}$		$\cup\mathsf{I}$		$\cup\mathsf{I}$		$\cup\mathsf{I}$
left-computable	**MLR**	$=$	**MLR**	$=$	**MLR**	$=$	**MLR**

Fig. 1. Known class inclusions

adaptively, because even in the latter, more powerful model one can uniformly in k enumerate an open cover of measure at most $1/k$ for all the sequences on which some universal martingale exceeds k. Furthermore, the classes in the first and second row of the last column coincide with the class of Kolmogorov-Loveland random sequences, because it can be shown that total and partial adaptive betting strategies yield the same concept of random sequence [6]. Finally, it follows easily from results of Buhrman et al. [1] that the class **TMR** of computably random sequences coincides with the class **TPR** of sequences that are random with respect to total permutation martingales, i.e., the ability to scan the bits of a sequence according to a computable permutation does not increase the power of total martingales.

Concerning non-inclusions, it is well-known that it holds that

$$\mathbf{KLR} \subsetneq \mathbf{PMR} \subsetneq \mathbf{TMR}.$$

Furthermore, Kastermans and Lempp [3] have recently shown that the Martin-Löf random sequences form a proper subclass of the class **PIR** of partial injective random sequences, i.e., $\mathbf{MLR} \subsetneq \mathbf{PIR}$.

Apart from trivial consequences of the definitions and the results just mentioned, nothing has been known about the relations of the randomness notions between computable randomness and Martin-Löf randomness in Figure 1. In what follows, we investigate the six randomness notions that are shown in Figure 1 in the range between **PIR** and **TMR**, i.e., between partial injective randomness as introduced below and computable randomness. We obtain a complete picture of the inclusion structure of these notions, more precisely we show that the notions are mutually distinct and indeed are mutually incomparable with respect to set theoretical inclusion, except for the inclusion relations that follow trivially by definition

4

and by the known relation $\mathbf{TMR} \subseteq \mathbf{TPR}$, see Figure 2 at the end of this paper. Interestingly these separation results are obtained by investigating the possible values of the Kolomogorov complexity of initial segments of random sequences for the different strategy types, and for some randomness notions we obtain essentially sharp bounds on how low these complexities can be.

Notation. We conclude the introduction by fixing some notation. The set of finite strings (or finite binary sequences, or words) is denoted by $2^{<\omega}$, ϵ being the empty word. We denote the set of infinite binary sequences by 2^ω. Given two finite strings w, w', we write $w \sqsubseteq w'$ if w is a prefix of w'. Given an element x of 2^ω or $2^{<\omega}$, $x(i)$ denotes the i-th bit of x (where by convention there is a 0-th bit and $x(i)$ is undefined if x is a word of length less than $i + 1$). If $A \in 2^\omega$ and $X = \{x_0 < x_1 < x_2 < \ldots\}$ is a subset of $\mathbb{N}$ then $A \upharpoonright X$ is the finite or infinite binary sequence $A(x_0)A(x_1)\ldots$. We abbreviate $A \upharpoonright \{0, \ldots, n - 1\}$ by $A \upharpoonright n$ (i.e., the prefix of A of length n).

C and K denote plain and prefix-free Kolmogorov complexity, respectively [2, 5]. The function log designates the logarithm of base 2. An **order** is a function $h : \mathbb{N} \to \mathbb{N}$ that is non-decreasing and tends to infinity.

2 Permutation and injection randomness

We now review the concept of martingale and betting strategy that are central for the unpredictability approach to define notions of an infinite random sequence.

Definition 1. *A **martingale** is a nonnegative, possibly partial, function $d : 2^{<\omega} \to \mathbb{Q}$ such that for all $w \in 2^{<\omega}$, $d(w0)$ is defined if and only if $d(w1)$ is, and if these are defined, then so is $d(w)$, and the relation $2d(w) = d(w0) + d(w1)$ holds. A martingale **succeeds** on a sequence $A \in 2^\omega$ if $d(A \upharpoonright n)$ is defined for all n, and $\limsup d(A \upharpoonright n) = +\infty$. We denote by $\mathrm{Succ}(d)$ the **success set** of d, i.e., the set of sequences on which d succeeds.*

Intuitively, a martingale represents the capital of a player who bets on the bits of a sequence $A \in 2^\omega$ in order, where at every round she bets some amount of money on the value of the next bit of A. If her guess is correct, she doubles her stake. If not, she loses her stake. The quantity $d(w)$, with w a string of length n, represents the capital of the player before the n-th round of the game (by convention there is a 0-th round) when the first n

bits revealed so far are those of w.

We say that a sequence A is **computably random** if no total computable martingale succeeds on it. One can extend this in a natural way to partial computable martingales: a sequence A is **partial computably random** if no partial martingale succeeds on it. No matter whether we consider partial or total computable martingales, this game model might be considered as too restrictive by the discussion in the introduction. Indeed, one could allow the player to bet on bits in any order she likes (as long as she can visit each bit at most once). This leads us to extend the notion of martingale to the notion of strategy.

Definition 2. *A **betting strategy** is a pair $b = (d, \sigma)$ where d is a martingale and $\sigma : 2^{<\omega} \to \mathbb{N}$ is a function.*

For a strategy $b = (d, \sigma)$, the function σ is called the *scan rule*. For a string w, $\sigma(w)$ represents the position of the next bit to be visited if the player has read the sequence of bits w during the previous moves. And as before, d specifies how much money is bet at each move. Formally, given an $A \in 2^{\omega}$, we define by induction a sequence of positions $n_0, n_1, \ldots$ by

$$\begin{cases} n_0 = \sigma(\epsilon), \\ n_{k+1} = \sigma\left(A(n_0)A(n_1)\ldots A(n_k)\right) \text{ for all } k \geq 0 \end{cases}$$

and we say that $b = (d, \sigma)$ **succeeds** on A if the n_i are all defined, are pairwise distinct (i.e., no bit is visited twice) and

$$\limsup_{k \to +\infty} d\left(A(n_0)\ldots A(n_k)\right) = +\infty$$

Here again, a betting strategy $b = (d, \sigma)$ can be total or partial. In fact, its partiality can be due either to the partiality of d or to the partiality of σ. We say that a sequence is **Kolmogorov-Loveland random** if no total computable betting strategy succeeds on it. As noted in [8], the concept of Kolmogorov-Loveland randomness remains the same if one replaces "total computable" by "partial computable" in the definition.

Kolmogorov-Loveland randomness is implied by Martin-Löf randomness and whether the two notions can be separated is a long-standing open question. As we discussed above, Miller and Nies [9] proposed to look at intermediate notions of randomness, where the power of non-monotonic betting strategies is limited. In the definition of a betting strategy, the scan rule is adaptive, i.e., the position of the next visited bit depends on the bits previously seen. It is interesting to look at non-adaptive games.

Definition 3. *In the above definition of a strategy, when $\sigma(w)$ only depends on the length of w for all w (i.e., the decision of which bit should be chosen at each move is independent of the values of the bits seen in previous moves), we identify σ with the (injective) function $\pi : \mathbb{N} \to \mathbb{N}$, where for all n $\pi(n)$ is the value of σ on words of length n ($\pi(n)$ indicates the position of the bit visited during the n-th move), and we say that $b = (d, \pi)$ is an **injection strategy**. If moreover π is bijective, we say that b is a **permutation strategy**. If π is the identity, the strategy $b = (d, \pi)$ is said to be **monotonic**, and can clearly be identified with the martingale d.*

All this gives a number of possible non-adaptive, non-monotonic, randomness notions: one can consider either monotonic, permutation, or injection strategies, and either total computable or partial computable ones. This gives a total of six randomness classes, which we denote by

$$\textbf{TMR}, \ \textbf{TPR}, \ \textbf{TIR}, \ \textbf{PMR}, \ \textbf{PPR}, \ \text{and} \ \textbf{PIR}, \tag{1}$$

where the first letter indicates whether we consider total (T) or partial (P) strategies, and the second indicates whether we look at monotonic (M), permutation (P) or injection (I) strategies. For example, the class **TMR** is the class of computably random sequences, while the class **PIR** is the class of sequences A such that no partial injection strategy succeeds on A. Recall in this connection that the known inclusions between the six classes in (1) and the classes **KLR** and **MLR** of Kolmogorov-Loveland random and Martin-Löf random sequences have been shown in Figure 1 above.

3 Randomness notions based on total computable strategies

We begin our study by the randomness notions arising from the game model where strategies are total computable. As we will see, in this model, it is possible to construct sequences that are random and yet have very low Kolmogorov complexity (i.e. all their initial segments are of low Kolmogorov complexity). We will see in the next section that this is no longer the case when we allow partial computable strategies in the model.

3.1 Building a sequence in TMR of low complexity

The following theorem is a first illustration of the phenomenon we just described.

Theorem 4 (Lathrop and Lutz [4], Muchnik [10]). *For every computable order h, there is a sequence $A \in \mathbf{TMR}$ such that, for all $n \in \mathbb{N}$,*

$$\mathrm{C}\left(A \upharpoonright n \mid n\right) \leq h(n) + \mathrm{O}(1).$$

We present a sketch of the proof of this theorem (which is essentially an adaptation of Muchnik's proof of Theorem 14 below). Several other proofs in the present paper will elaborate on the techniques it utilizes.

Proof (Idea). Defeating one total computable martingale is easy and can be done computably, that is, for every total computable martingale d there exists a sequence A, uniformly computable in d, such that $A \notin \mathrm{Succ}(d)$. Indeed, given a martingale d, for any given w, one has either $d(w0) \leq d(w)$ or $d(w1) < d(w)$. Thus, one can easily construct a computable sequence A by setting $A \upharpoonright 0 = \epsilon$ and by induction, having defined $A \upharpoonright n$, setting $A \upharpoonright n + 1 = (A \upharpoonright n)i$ where $i \in \{0, 1\}$ is such that $d((A \upharpoonright n)i) \leq d(A \upharpoonright n)$. This can of course be done computably since d is total computable, and by construction of A, $d(A \upharpoonright n)$ is non-increasing, meaning in particular that d does not succeed on A.

Defeating a finite number of total computable martingales is equally easy. Indeed, given a finite number $d_1, \ldots, d_k$ of such martingales, their sum $D = d_1 + \ldots + d_k$ is itself a total computable martingale (this follows directly from the definition). Thus, we can construct as above a computable sequence A that defeats D. And since $D \geq d_i$ for all $1 \leq i \leq k$, this implies that A defeats all the d_i. Note that this argument would work just as well if we had taken D to be any weighted sum $\alpha_1 d_1 + \ldots + \alpha_k d_k$, with positive rational constants α_i.

We now need to deal with the general case where we have to defeat *all* total computable martingales simultaneously. We will proceed using a diagonalization technique. Of course, this diagonalization cannot be carried out effectively, since there are infinitely many such martingales and since we do not even know whether any one given partial computable martingale is total. The first problem can easily be overcome by introducing the martingales to diagonalize against one by one instead of all at the beginning. So at first, for a number of stages we will only take into account the first computable martingale d_1. Then (maybe after a long time) we may introduce the second martingale d_2, with a small coefficient α_2 and then consider the martingale $d_1 + \alpha_2 d_2$ (α_2 is chosen to be sufficiently

8

small to ensure that the difference in capital between d_1 and $d_1 + \alpha_2 d_2$ is small at the time when d_2 is added). Much later we can introduce the third martingale d_3 with an even smaller coefficient α_3, and diagonalize against $d_1 + \alpha_2 d_2 + \alpha_3 d_3$, and so on. So in each step of the construction we have to consider just a finite number of martingales.

The non-effectivity of the construction arises from the second problem, deciding which of our partial computable martingales are total. However, once we are supplied with this additional information, we *can* effectively carry out the construction of A. And since for each step we need to consider only finitely many potentially total martingales, the information we need to construct the first n bits of A for some fixed n is finite, too. Say, for example, that for the first n stages of the construction – i.e., to define $A \upharpoonright n$ – we decided on only considering k martingales $d_0, \ldots, d_{k-1}$. Then we need no more than k bits, carrying the information which martingales among $d_0, \ldots, d_{k-1}$ are total, to describe $A \upharpoonright n$. That way, we get $\mathrm{C}\,(A \upharpoonright n \mid n) \leq k + O(1)$.

As can be seen from the above example, the complexity of descriptions of prefixes of A depends on how fast we introduce the martingales. This is where our orders come into play. Fix a fast-growing computable function f with $f(0) = 0$, to be specified later. We will introduce a new martingale at every position of type $f(k)$, that is, between positions $[f(k), f(k+1))$, we will only diagonalize against $k + 1$ martingales, hence by the above discussion, for every $n \in [f(k), f(k+1))$, we have

$$\mathrm{C}\,(A \upharpoonright n \mid n) \leq k + O(1)$$

Thus, if the function f grows faster than the inverse function h^{-1} of a given order h, we get

$$\mathrm{C}\,(A \upharpoonright n \mid n) \leq h(n) + O(1)$$

for all n. $\qquad\square$

3.2 TMR = TPR: the averaging technique

It turns out that, perhaps surprisingly, the classes **TMR** and **TPR** coincide. This fact was stated explicitly in Merkle et al [8], but is easily derived from the ideas introduced in Buhrman et al [1]. We present the main ideas of their proof as we will later need them. We shall prove:

Theorem 5. *Let $b = (d, \pi)$ be a total computable permutation strategy. There exists a total computable martingale d such that $\mathrm{Succ}(b) \subseteq \mathrm{Succ}(d)$.*

This theorem states that total permutation strategies are no more powerful than total monotonic strategies, which obviously entails **TMR** = **TPR**. Before we can prove it, we first need a definition.

Definition 6. *Let $b = (d, \pi)$ be a total injective strategy. Let $w \in 2^{<\omega}$. We can run the strategy b on w as if it were an element of 2^ω, stopping the game when b asks to bet on a bit of position outside w. This game is of course finite (for a given w) since at most $|w|$ bets can be made. We define $\hat{b}(w)$ to be the capital of b at the end of this game. Formally: $\hat{b}(w) = d\left(w_{\pi(0)} \ldots w_{\pi(N-1)}\right)$ where N is the smallest integer such that $\pi(N) \geq |w|$.*

Note that if $b = (d, \pi)$ is a total computable injection martingale, $\hat{b}$ is total computable.

If $\hat{b}$ was itself a monotonic martingale, Theorem 5 would be proven. This is however not the case in general: Suppose the starting capital is $d(\epsilon) = 1$, the scan rule is $\pi(0) = 1$, $\pi(1) = 0$ and the betting strategy is described by $d(0) = 2$, $d(00) = 4$, $d(01) = 0$ and by $d(1) = 0$, $d(10) = d(11) = 0$ that is, d first visits the bit in position 1, betting everything on the value 0, then visits the bit in position 0, again betting everything on the value 0. We then have

$$\frac{\hat{b}(00) + \hat{b}(01)}{2} = \frac{4 + 0}{2} = 2 \neq 1 = \hat{b}(0),$$

which shows that $\hat{b}$ is not a martingale.

The trick is, given a betting strategy b and a word w, to look at the *expected value* of b on w, i.e., look at the mathematical expectation of $b(w')$ for large enough extensions w' of w. Specifically, given a total betting strategy $b = (d, \pi)$ and a word w of length n, we take an integer M large enough to have

$$\pi\left([0, \ldots, M-1]\right) \cap [0, \ldots, n-1] = \pi(\mathbb{N}) \cap [0, \ldots, n-1]$$

(i.e. the strategy b will never bet on a bit of position less than n after the M-th move), and define:

$$\mathrm{Av}_b(w) = \frac{1}{2^M} \sum_{\substack{w \sqsubseteq w' \\ |w'| = M}} \hat{b}(w')$$

Proposition 7 (Buhrman et al [1], Kastermans and Lempp [3]).

(i) *The quantity* $\mathrm{Av}_b(w)$ *(defined above) is well-defined i.e. does not depend on M as long as it satisfies the required condition.*

(ii) *For a total injective strategy b, Av_b is a martingale.*

(iii) *For a given injective strategy b and a given word w of length n, $\mathrm{Av}_b(w)$ can be computed if we know the set $\pi(\mathbb{N}) \cap [0, \ldots, n-1]$. In particular, if b is a total computable permutation strategy, then Av_b is total computable.*

As Buhrman et al. [1] explained, it is not true in general that if a total computable injective strategy b succeeds against a sequence A, then Av_b also succeeds on A. However, this can be dealt with using the well-known "saving trick". Suppose we are given a martingale d with initial capital, say, 1. Consider the variant d' of d that does the following: when run on a given sequence A, d' initially plays exactly as d. If at some stage of the game d' reaches a capital of 2 or more, it then puts half of its capital on a "bank account", which will never be used again. From that point on, d' bets half of what d does, that is starts behaving like $d/2$ (plus the saved capital). If later in the game the "non-saved" part of its capital reaches 2 or more, then half of it is placed on the bank account and then d' starts behaving like $d/4$, and so on.

For every martingale d' that behaves as above (i.e. saves half of its capital as soon as it exceeds twice its starting capital), we say that d' has the "saving property". It is clear from the definition that if d is computable, then so is d', and moreover d' can be uniformly computed given an index for d. Moreover, if for some sequence A one has

$$\limsup_{n \to +\infty} d(A \restriction n) = +\infty$$

then

$$\lim_{n \to +\infty} d'(A \restriction n) = +\infty$$

which in particular implies $\mathrm{Succ}(d) \subseteq \mathrm{Succ}(d')$ (it is easy to see that it is in fact an equality). Thus, whenever one considers a martingale d, one can assume without loss of generality that it has the saving property (as long as we are only interested in the success set of martingales, not in the growth rate of their capital). The key property (for our purposes) of saving martingales is the following.

Lemma 8. *Let $b = (d, \pi)$ be a total injective strategy such that d has the saving property. Let $d' = \mathrm{Av}_b$. Then $\mathrm{Succ}(b) \subseteq \mathrm{Succ}(d')$.*

Proof. Suppose that $b = (d, \pi)$ succeeds on a sequence A. Since d has the saving property, for arbitrarily large k there exists a finite prefix $A \restriction n$ of A such that a capital of at least k is saved during the finite game of b on $A \restriction n$. We then have $\hat{b}(w') \geq k$ for all extensions w' of $A \restriction n$ (as a saved capital is never used), which by definition of Av_b implies $\mathrm{Av}_b(A \restriction m) \geq k$ for all $m \geq n$. Since k can be chosen arbitrarily large, this finishes the proof. $\qquad\square$

Now the proof of Theorem 5 is as follows. Let $b = (d, \pi)$ be a total computable permutation strategy. By the above discussion, let d' be the saving version of d, so that $\mathrm{Succ}(d) \subseteq \mathrm{Succ}(d')$. Setting $b' = (d', \pi)$, we have $\mathrm{Succ}(b) \subseteq \mathrm{Succ}(b')$. By Proposition 7 and Lemma 8, $d'' = \mathrm{Av}_{b'}$ is a total computable martingale, and

$$\mathrm{Succ}(b) \subseteq \mathrm{Succ}(b') \subseteq \mathrm{Succ}(d'')$$

as wanted.

3.3 Understanding the strength of injective strategies: the class TIR

While the class of computably random sequence (i.e. the class **TMR**) is closed under computable permutations of the bits, we now see that this result does not extend to computable injections, as witnessed by the next theorem.

Theorem 9. *Let $A \in 2^\omega$. Let $\{n_k\}_{k \in \mathbb{N}}$ be a computable sequence of integers such that $n_{k+1} \geq 2n_k$ for all k. Suppose that A is such that:*

$$\mathrm{C}\,(A \restriction n_k \mid k) \leq \log(n_k) - 3\log(\log(n_k))$$

*for infinitely many k. Then $A \notin$ **TIR**.*

Proof. Let A be a sequence satisfying the hypothesis of the theorem. Assuming, without loss of generality, that $n_0 = 0$, we partition $\mathbb{N}$ into an increasing sequence of intervals $I_0, I_1, I_2, \ldots$ where $I_k = [n_k, n_{k+1})$. Notice that we have for all k:

$$\mathrm{C}\,(A \restriction I_k \mid k) \leq \mathrm{C}\,(A \restriction n_{k+1} \mid k+1) + O(1)$$

By the hypothesis of the theorem, the right-hand side of the above inequality is bounded by $\log(n_{k+1}) - 3\log(\log(n_{k+1}))$ for infinitely many k.

Additionally, we have $|I_k| = n_{k+1} - n_k$ which by hypothesis on the sequence n_k implies $|I_k| \geq n_{k+1}/2$, and hence $\log(|I_k|) = \log(n_{k+1}) + O(1)$ and $\log(\log(|I_k|)) = \log(\log(n_{k+1})) + O(1)$. It follows that

$$\mathrm{C}\left(A \upharpoonright I_k \mid k\right) \leq \log(|I_k|) - 3\log(\log(|I_k|)) - O(1)$$

for infinitely many k, hence

$$\mathrm{C}\left(A \upharpoonright I_k \mid k\right) \leq \log(|I_k|) - 2\log(\log(|I_k|))$$

for infinitely many k.

Let us call S_k the set of strings w of length $|I_k|$ such that $\mathrm{C}\left(w \mid |I_k|\right) \leq \log(|I_k|) - 2\log(\log(|I_k|))$ (to which $A \upharpoonright I_k$ belongs for infinitely many k). By the standard counting argument, there are at most

$$s_k = 2^{\log(|I_k|) - 2\log(\log(|I_k|))} = \frac{|I_k|}{\log^2(|I_k|)}$$

strings in S_k. For every k, we split I_k into s_k consecutive disjoint intervals of equal length:

$$I_k = J_k^0 \cup J_k^1 \cup \ldots \cup J_k^{s_k - 1}$$

We design a betting strategy as follows. We start with a capital of 2. We then reserve for each k an amount $1/(k+1)^2$ to be bet on the bits in positions in I_k (this way, the total amount we distribute is smaller than 2), and we split this evenly between the J_k^i, i.e. we reserve an amount $\frac{1}{s_k \cdot (k+1)^2}$ for every J_k^i. It is clear from the definition that the sets S_k are c.e. uniformly in k, hence we can enumerate them in parallel by dovetailing. Whenever the e-th element w_k^e of some S_k is enumerated, we see w_k^e as a possible candidate to be equal to $A \upharpoonright I_k$, and we bet the reserved amount $\frac{1}{s_k \cdot (k+1)^2}$ on the fact that $A \upharpoonright I_k$ coincides with w_k^e on the bits whose position is in J_k^e. If we are successful (this in particular happens whenever $w_k^e = A \upharpoonright I_k$), our reserved capital for this J_k^e is multiplied by $2^{|J_k^e|}$, i.e. we now have for this J_k^e, a capital of

$$\frac{1}{s_k \cdot (k+1)^2} \cdot 2^{(|I_k|/s_k)}$$

13

Replacing s_k by its value (and remembering that $|I_k| \geq 2^{k-O(1)}$), an elementary calculation shows that this quantity is greater than 1 for almost all k. Thus, our betting strategy succeeds on A. Indeed, for infinitely many k, $A \restriction I_k$ is an element of S_k, hence for some e we will be successful in the above sub-strategy, making an amount of money greater than 1 for infinitely many k, hence our capital tends to infinity throughout the game. Finally, it is easy to see that this betting strategy is total: it simply is a succession of doubling strategies on an infinite c.e. set of words, and it is injective as the J_k^e form a partition of $\mathbb{N}$, and the order of the bits we bet on is independent of A (in fact, we see our betting strategy succeeds on *all* sequences α satisfying the hypothesis of the theorem). $\qquad\square$

As an immediate corollary, we get the following.

Corollary 10. *If for a sequence A we have for all n $\mathrm{C}\,(A \restriction n \mid n) < \log n - 4 \log \log n + \mathrm{O}(1)$, then $A \notin \mathbf{TIR}$.*

Another interesting corollary of our construction is that the class of all computable sequences can be covered by a single total computable injective strategy.

Corollary 11. *There exists a single total computable injective strategy which succeeds against all computable elements of 2^ω.*

Proof. This is because, as just remarked, the strategy we construct in the proof of Theorem 9 succeeds against *every* sequence A such that $\mathrm{C}\,(A \restriction n_k \mid k) \leq \log(n_k) - 3 \log(\log(n_k))$ for infinitely many k. This in particular includes all computable sequences A, for which $\mathrm{C}\,(A \restriction n_k \mid k) = O(1)$. $\qquad\square$

The lower bound on Kolmogorov complexity given in Theorem 9 is quite tight, as witnessed by the following theorem.

Theorem 12. *For every computable order h there is a sequence $A \in \mathbf{TIR}$ such that $\mathrm{C}(A \restriction n \mid n) \leq \log(n) + h(n) + \mathrm{O}(1)$. In particular, we have $\mathrm{C}(A \restriction n) \leq 2\log(n) + h(n) + \mathrm{O}(1)$.*

Proof. The proof is a modification of the proof of Theorem 4. This time, we want to diagonalize against all *non-monotonic* total computable injective betting strategies. Like in the proof of Theorem 4, we add them one by one, discarding the partial strategies. However, to achieve the construction of A

by diagonalization, we will diagonalize against the average martingales of the strategies we consider. As explained on page 11, we can assume that all total computable injective strategies have the saving property, hence defeating Av_b is enough to defeat b (by Lemma 8). The proof thus goes as follows:

Fix a fast growing computable function f, to be specified later. We start with a martingale $D_0 = 1$ (the constant martingale equal to 1) and $w_0 = \epsilon$. For all k we do the following. Assume we have constructed a prefix w_k of A of length $f(k)$, and that we are currently diagonalizing against a martingale D_k, so that $D_k(w_k) < 2$. We then enumerate a new partial computable injective betting strategy b. If it is not total, we memorize this fact using one extra bit of information, and we set $D_{k+1} = D_k$. Otherwise, we set $d_{k+1} = \mathrm{Av}_b$ and compute a positive rational α_{k+1} such that $(D_k + \alpha_{k+1} d_{k+1})(w_k) < 2$, and finally set $D_{k+1} = D_k + \alpha_{k+1} d_{k+1}$.

Then, we define w_{k+1} to be the extension of w_k of length $f(k + 1)$ by the usual diagonalization against D_{k+1}, maintaining the inequality $D_{k+1}(u) < 2$ for all prefixes u of w_{k+1}. The infinite sequence A obtained this way defeats all the average martingales of all total computable injective strategies, hence by Lemma 8, $A \in \mathbf{TIR}$.

It remains to show that A has low Kolmogorov complexity. Suppose we want to describe $A \upharpoonright n$ for some $n \in [f(k), f(k + 1))$. This can be done by giving n, the subset of $\{0, \dots, k\}$ (of complexity $k + O(1)$) corresponding to the indices of the total computable injective strategies among the first k partial computable ones, and by giving the restriction of D_{k+1} to words of length at most n. From all this, $A \upharpoonright n$ can be reconstructed following the above construction. It remains to evaluate the complexity of the restriction of D_{k+1} to words of length at most n. We already know the total computable injective strategies $b_0, \dots, b_k$ that are being considered in the definition of D_{k+1}. For all i, let π_i be the injection associated to b_i. We need to compute, for all $0 \le i \le k$, the martingale $d_i = \mathrm{Av}_{b_i}$ on words of length at most n. By Proposition 7, this can be done knowing $\pi_i(\mathbb{N}) \cap [0, \dots, n-1]$ for all $0 \le i \le k$. But if the π_i are known, this set is uniformly c.e. in i, n. Hence, we can enumerate all the sets $\pi_i(\mathbb{N}) \cap [0, \dots, n-1]$ (for $0 \le i \le k$) in parallel, and simply give the last couple (i, l) such that l is enumerated in $\pi_i(\mathbb{N}) \cap [0, \dots, n-1]$. Since $0 \le i \le k$ and $0 \le l < n$, this costs an amount of information $O(\log k) + \log n$. To sum up, we get

$$\mathrm{C}\left(A \upharpoonright n \mid n\right) \le k + \mathrm{O}(\log k) + \log n$$

Thus, it suffices to take f growing fast enough to ensure that the term $k + \mathrm{O}(\log k)$ is smaller than $h(n) + \mathrm{O}(1)$. $\qquad\qquad\square$

4 Randomness notions based on partial computable strategies

We now turn our attention to the second line of Figure 1, i.e., to those randomness notions that are based on partial computable betting strategies.

4.1 The class PMR: partial computable martingales are stronger than total ones

We have seen in the previous section that some sequences in **TIR** (and a fortiori **TPR** and **TMR**) may be of very low complexity, namely logarithmic. This is not the case anymore when one allows partial computable strategies, even monotonic ones.

Theorem 13 (Merkle [7]). *If* $\mathrm{C}(A \restriction n) = \mathrm{O}(\log n)$ *then* $A \notin \mathbf{PMR}$.

However, the next theorem, proven by An. A. Muchnik, shows that allowing slightly super-logarithmic growth of the Kolmogorov complexity is enough to construct a sequence in **PMR**.

Theorem 14 (Muchnik et al. [10]). *For every computable order* h *there is a sequence* $A \in \mathbf{PMR}$ *such that, for all* $n \in \mathbb{N}$,

$$\mathrm{C}\left(A \restriction n \mid n\right) \leq h(n) \log(n) + \mathrm{O}(1).$$

Proof (Idea). The proof is almost identical to the proof of Theorem 4. The only difference is that we insert *all* partial computable martingales one by one, and diagonalize against their weighted sum as before. It may happen however, that at some stage of the construction, one of the martingales becomes undefined. All we need to do then is to memorize this, and ignore this particular martingale from that point on. Call A the sequence we obtain by this construction. We want to describe $A \restriction n$. To do so, we need to specify n, and, out of the k partial computable martingales that are inserted before stage n, which ones have diverged, and at what stage, hence an information of $\mathrm{O}(k \log n)$ (giving the position where a particular martingale diverges costs $\mathrm{O}(\log n)$ bits, and there are k martingales). Since we can insert martingales as slowly as we like (following

some computable order), the complexity of $A \restriction n$ given n can be taken to be smaller than $h(n) \log n + O(1)$ (where h is a computable order, fixed before the construction of A). $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

4.2 The class PPR

In the case of total strategies, allowing permutation gives no real additional power, as **TMR** = **TPR**. Very surprisingly, Muchnik showed that in the case of partial computable strategies, permutation strategies are a real improvement over monotonic ones, as the following theorem shows (quite a contrast to Theorem 14!).

Theorem 15 (Muchnik [10]). *If there is a computable order h such that for all n we have $\mathrm{K}(A \restriction n) \leq n - h(n) - \mathrm{O}(1)$, then $A \notin$ **PPR**.*

Note that the proof used by Muchnik in [10] works if we replace K by C in the above statement.

While Muchnik's theorem shows that all sequences in **PPR** have to have infinitely many prefixes of high complexity, it turns out that some of them also have infinitely many prefixes of very low complexity.

Theorem 16. *For every computable order h there is a sequence $A \in$ **PPR**, such that there are infinitely many n where $\mathrm{C}\,(A \restriction n \mid n) < h(n)$.*

Furthermore, given any infinite computable set $S \subseteq \mathbb{N}$, the above remains true if we replace "infinitely many n" by "infinitely many n in S".

Lemma 17. *Let d be a partial computable martingale. Let $\mathcal{C}$ be an effectively closed subset of 2^ω (where "effectively closed" is another expression for being a Π^0_1 class). Suppose that d is total on every element of $\mathcal{C}$. Then there exists a total computable martingale d' such that $\mathrm{Succ}(d) \cap \mathcal{C} = \mathrm{Succ}(d') \cap \mathcal{C}$.*

Proof. The idea of the proof is simple: the martingale d' will try to mimic d while enumerating the complement $\mathcal{U}$ of $\mathcal{C}$. If at some stage a cylinder $[w]$ is covered by $\mathcal{U}$, then d' will be passive (i.e. defined but constant) on the sequences extending w. As we do not care about the behavior of d' on $\mathcal{U}$ (as long as it is defined), this will be enough to get the conclusion.

The details will now be given. Let $d, \mathcal{C}$ be as above. We build the martingale d' on words by induction. Define $d'(\epsilon) = d(\epsilon)$ (here we assume

17

without loss of generality that $d(\epsilon)$ is defined, otherwise there is nothing to prove). During the construction, some words will be marked as inactive, on which the martingale will be passive; initially, there is no inactive word. On active words w, we will have $d(w) = d'(w)$.

Suppose for the sake of the induction that $d'(w)$ is already defined. If w is marked as inactive, we mark $w0$ and $w1$ as inactive, and set $d'(w0) = d'(w1) = d'(w)$. Otherwise, by the induction hypothesis, we have $d(w) = d'(w)$. We then run in parallel the computation of $d(w0)$ and $d(w1)$, and enumerate the complement $\mathcal{U}$ of $\mathcal{C}$ until one of the two following events happens:

(a) $d(w0)$ and $d(w1)$ become defined. Then set $d'(w0) = d(w0)$ and $d'(w1) = d(w1)$
(b) The cylinder $[w]$ gets covered by $\mathcal{U}$. In that case, mark $w0$ and $w1$ as inactive and set $d'(w0) = d'(w1) = d'(w)$

Note that one of these two events *must* happen: indeed, if $d(w0)$ and $d(w1)$ are undefined (remember that by the definition of a martingale, Definition 1, they are either both defined or both undefined), then this means that d diverges on *any* element of $[w0] \cup [w1] = [w]$. Hence, by assumption, $[w] \cap \mathcal{C} = \emptyset$, i.e. $[w] \subseteq \mathcal{U}$. It remains to verify that $\mathrm{Succ}(d) \cap \mathcal{C} = \mathrm{Succ}(d') \cap \mathcal{C}$. Let $A \in \mathcal{C}$. Since d is total on A by assumption, during the construction of d' on A, we will always be in case (a), hence we will have for all n, $d(A \restriction n) = d'(A \restriction n)$. The result follows immediately. $\qquad\square$

Corollary 18. *Let $b = (d, \pi)$ be a partial computable permutation strategy (resp. injective strategy). Let $\mathcal{C}$ be an effectively closed subset of 2^ω. Suppose that b is total on every element of $\mathcal{C}$. Then there exists a total computable permutation strategy (resp. injective strategy) b' such that $\mathrm{Succ}(b) \cap \mathcal{C} = \mathrm{Succ}(b') \cap \mathcal{C}$.*

Proof. This follows from the fact that the image or pre-image of an effectively closed set under a computable permutation of the bits is itself a closed set: take $b = (d, \pi)$ and $\mathcal{C}$ as above. Let $\bar{\pi}$ be the map induced on 2^ω by π, i.e. the map defined for all $A \in 2^\omega$ by

$$\bar{\pi}(A) = A(\pi(0))A(\pi(1))A(\pi(2))\dots$$

For any given sequence $A \in \mathcal{C}$, b succeeds on A if and only if d succeeds on $\bar{\pi}(A)$. As $\bar{\pi}(A) \in \bar{\pi}(\mathcal{C})$, and $\bar{\pi}(\mathcal{C})$ is an effectively closed set, by

Lemma 17, there exists a total martingale d' such that $\operatorname{Succ}(d) \cap \bar{\pi}(\mathcal{C}) = \operatorname{Succ}(d') \cap \bar{\pi}(\mathcal{C})$. Thus, d' succeeds on $\bar{\pi}(A)$, or equivalently, $b' = (d', \pi)$ succeeds on A. Thus b' is as desired. $\qquad\square$

Proof (of Theorem 16). Again, this proof is a variant of the proof of Theorem 4: we add strategies one by one, diagonalizing, at each stage, against a finite weighted sum of total monotonic strategies (i.e. martingales). Of course, not all the strategies we diagonalize against are total and monotonic, but we can reduce to this case using the techniques we presented above. Suppose that in the construction of our sequence A, we have already constructed an initial segment w_k, and that up to this stage we played against a weighted sum of k total martingales

$$D_k = \sum_{i=1}^{k} \alpha_i \, d_i$$

where the d_i are total computable martingales, ensuring that $D_k(u) < 2$ for all prefixes u of w. Suppose we want to introduce a new strategy $b = (d, \pi)$. There are three cases:

Case 0: the new strategy is not valid, i.e. π is not a permutation. In this case, we just add one bit of extra information to record this, and ignore b from now on, that is we set $w_{k+1} = w_k$, $d_{k+1} = 0$ (the zero martingale), and $D_{k+1} = D_k + d_{k+1} = D_k$.

Case 1: the strategy b is indeed a partial computable permutation strategy, and there exists an extension w' of w such that $D_k(u) < 2$ for all prefixes u of w', and b diverges on w' (meaning that for some $n < |w'|$, $\pi(0), \dots, \pi(n) < |w'|$, but $d(w_{\pi(0)} \dots w_{\pi(n)})$ diverges) . In this case, we simply take w' as our new prefix of A, as it both diagonalizes against D, and defeats b (since b diverges on w', it will not win against *any* possible extension of w'). We can thus ignore b from that point on, so we set $w_{k+1} = w'$, $d_{k+1} = 0$ and $D_{k+1} = D_k + d_{k+1} = D_k$.

Case 2: if we are not in one of the two previous cases, this means that our strategy $b = (d, \pi)$ is a partial computable permutation strategy, and that b is total on the whole Π_1^0 class

$$\mathcal{C}_k = [w_k] \cap \{X \in 2^\omega \mid \forall n \, D_k(X \restriction n) < 2\}$$

Thus, by Lemma 18, there exists a total computable permutation strategy b' such that $\operatorname{Succ}(b) \cap \mathcal{C}_k = \operatorname{Succ}(b') \cap \mathcal{C}_k$. And by Theorem 5, there

exists a total computable martingale d'' such that $\mathrm{Succ}(b') \subseteq \mathrm{Succ}(d'')$. Thus, we can replace b by d'', and defeating d'' will be enough to defeat b as long as the sequence we construct is in $\mathcal{C}_k$. We thus set $d_{k+1} = d''$, $w_{k+1} = w_k$ and

$$D_{k+1} = \sum_{i=1}^{k+1} \alpha_i \, d_i$$

where α_{k+1} is sufficiently small to have $D_{k+1}(w_{k+1}) < 2$.

Once we have added a new monotonic martingale, we (as usual) computably find an extension w'' of w_{k+1}, ensuring that $D_{k+1}(u) < 2$ for all prefixes u of w'', taking w'' long enough to have $\mathrm{C}\,(w'' \mid |w''|) \leq h(|w''|)$. We then set $w_{k+1} = w''$, then add a $k+2$-th strategy and so on.

Note that since w'' can be chosen arbitrarily large, if we have fixed a computable subset S of $\mathbb{N}$, we can also ensure that $|w''|$ belongs to S if we like.

It is clear that the infinite sequence A constructed via this process satisfies

$$\mathrm{C}\,(A \upharpoonright n \mid n) \leq h(n)$$

for infinitely many n (and, since Case 2 happens infinitely often, if we fix a given computable set S, we can ensure that infinitely many of such n belong to S). To see that it belongs to **PPR**, we notice that for all k, $D_{k+1} \geq D_k$ and $w_k \sqsubseteq w_{k+1}$, which by definition implies $\mathcal{C}_{k+1} \subseteq \mathcal{C}_k$ and thus $A \in \bigcap_k \mathcal{C}_k$. Now, given a partial computable permutation strategy $b = (d, \pi)$ which is defined on all prefixes of A, let k be the stage where b was introduced. Since b is total along A, Case 1 did not hold for b then and it was replaced by a martingale d'' as in Case 2. Since by construction of A, d_{k+1} does not succeed on A and by definition of d'', $\mathrm{Succ}(b) \cap \mathcal{C}_k \subseteq \mathrm{Succ}(d'') \cap \mathcal{C}_k$, it follows that $A \notin \mathrm{Succ}(b)$. $\qquad\square$

Now that we have assembled all our tools, we can easily prove the desired results.

Theorem 19. *The following statements hold.*

1. **PPR** $\nsubseteq$ **TIR**
2. **TIR** $\nsubseteq$ **PMR**
3. **PMR** $\nsubseteq$ **PPR**

From these results it easily follows that in Figure 2 no inclusion holds except those indicated and those implied by transitivity.

	monotonic		permutation		injection
total	**TMR**	$=$	**TPR**	$\supsetneq$	**TIR**
	$\cup\!\!\!\!\backslash$		$\cup\!\!\!\!\backslash$		$\cup\!\!\!\!\backslash$
partial	**PMR**	$\supsetneq$	**PPR**	$\supsetneq$	**PIR**

Fig. 2. Assembled class inclusion results

Proof. 1. Choose a computable sequence $\{n_k\}_k$ fulfilling the requirements of Theorem 9 such that $\mathrm{C}(k) \leq \log\log n_k$ for all k. The members of this set then form a computable set S. Use Theorem 16 to construct a sequence $A \in \mathbf{PPR}$ such that $\mathrm{C}(A \restriction n \mid n) < \log\log n$ at infinitely many places in S. We then have for infinitely many k

$$\mathrm{C}(A \restriction n_k \mid k) \leq \mathrm{C}(A \restriction n_k) \leq \mathrm{C}(A \restriction n_k \mid n_k) + 2\log\log n_k \leq 3\log\log n_k,$$

so A cannot be in **TIR** according to Theorem 9.
2. Follows immediately from Theorems 12 and 13.
3. Follows immediately from Theorems 14 and 15. □

References

1. Harry Buhrman, Dieter van Melkebeek, Kenneth Regan, D. Sivakumar, and Martin Strauss. A generalization of resource-bounded measure, with application to the BPP vs. EXP problem. *SIAM Journal of Computing*, 30(2):576–601, 2000.
2. Rod Downey and Denis Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, 2010.
3. Bart Kastermans and Steffen Lempp. Comparing notions of randomness. *Theor. Comput. Sci.*, 411:602–616, 2010.
4. James Lathrop and Jack Lutz. Recursive computational depth. *Information and Computation*, 153(1):139–172, 1999.
5. Ming Li and Paul Vitányi. *Kolmogorov Complexity and Its Applications*. Springer, 2008.
6. Wolfgang Merkle. The Kolmogorov-Loveland stochastic sequences are not closed under selecting subsequences. *Journal of Symbolic Logic*, 68:1362–1376, 2003.
7. Wolfgang Merkle. The complexity of stochastic sequences. *Journal of Computer and System Sciences*, 74(3):350–357, 2008.

8. Wolfgang Merkle, Joseph S. Miller, André Nies, Jan Reimann, and Frank Stephan. Kolmogorov-Loveland randomness and stochasticity. *Annals of Pure and Applied Logic*, 138(1-3):183–210, 2006.

9. Joseph S. Miller and André Nies. Randomness and computability: open questions. *Bulletin of Symbolic Logic*, 12(3):390–410, 2006.

10. Andrei A. Muchnik, Alexei Semenov, and Vladimir Uspensky. Mathematical metaphysics of randomness. *Theoretical Computer Science*, 207(2):263–317, 1998.

11. André Nies. *Computability and Randomness*. Oxford University Press, 2009.