

# Separations of non-monotonic randomness notions

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

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 Kasternans 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 such sequences that are highly compressible, i.e., show a large amount of redundancy, see Theorem 4.2 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, and 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 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 is implied by and in fact is quite close to Martin-Löf randomness, see Theorem 14 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  $\pi(0), \pi(1), \dots$  for a permutation or an injection  $\pi$  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.

	monotonic	permutation	injection	adaptive
total	<b>TMR</b> $\cup$	= <b>TPR</b> $\cup$	$\supseteq$ <b>TIR</b> $\cup$	$\supseteq$ <b>KLR</b> $\cup$
partial	<b>PMR</b> $\cup$	$\supseteq$ <b>PPR</b> $\cup$	$\supseteq$ <b>PIR</b> $\cup$	$\supseteq$ <b>KLR</b> $\cup$
left-computable	<b>MLR</b>	= <b>MLR</b>	= <b>MLR</b>	= <b>MLR</b>

**Fig. 1.** Known class inclusions

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 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, Kasternans 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 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  $2^{<\omega}$ ,  $\epsilon$  being the empty word. We denote the set of infinite binary sequences by  $2^\omega$ . Given two finite strings  $w, w'$ , we write  $w \sqsubseteq w'$  if  $w$  is a prefix of  $w'$ . Given an element  $x$  of  $2^\omega$  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  $X = \{x_0 < x_1 < x_2 < \dots\}$  is a subset of  $\mathbb{N}$  then  $A \upharpoonright X$  is the finite or infinite binary sequence  $A(x_0)A(x_1)\dots$ . We abbreviate  $A \upharpoonright \{0, \dots, n - 1\}$  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  $h : \mathbb{N} \rightarrow \mathbb{N}$  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  $d : 2^{<\omega} \rightarrow \mathbb{Q}$  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  $\text{Succ}(d)$  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 computable martingale succeeds on it. No matter whether we consider partial or total computable martingales, this game model can be seen 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  $\sigma : 2^{<\omega} \rightarrow \mathbb{N}$  is a function.

For a strategy  $b = (d, \sigma)$ , the term  $\sigma$  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  $n_0, n_1, \dots$  by

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

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

$$\limsup_{k \rightarrow +\infty} d(A(n_0) \dots A(n_k)) = +\infty$$

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  $\sigma$ . 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 one of the most important open problems on algorithmic randomness. 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  $\sigma$  with the (injective) function  $\pi : \mathbb{N} \rightarrow \mathbb{N}$ , where for all  $n$   $\pi(n)$  is the value of  $\sigma$  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  $\pi$  is bijective, we say that  $b$  is a **permutation strategy**. If  $\pi$  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

$$\mathbf{TMR}, \mathbf{TPR}, \mathbf{TIR}, \mathbf{PMR}, \mathbf{PPR}, \text{ and } \mathbf{PIR}, \quad (1)$$

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 Sequences in TMR and TPR may have 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  $n \in \mathbb{N}$ ,*

$$C(A \upharpoonright n \mid n) \leq h(n) + O(1).$$

*Proof (Idea).* Defeating one total computable martingale is easy and can be done computably, i.e., for every total computable martingale  $d$  there exists a sequence  $A$ , uniformly computable in  $d$ , such that  $A \notin \text{Succ}(d)$ . Indeed, given a martingale  $d$ . For any given  $w$ , one has either  $d(w0) \leq d(w)$  or  $d(w1) \leq d(w)$ . Thus, one can easily construct a computable sequence  $A$  by setting  $A \upharpoonright 0 = \epsilon$  and by induction, having defined  $A \upharpoonright n$ , we choose  $A \upharpoonright n+1 = (A \upharpoonright n)i$  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 against  $A$ .

Defeating a finite number of total computable martingales is equally easy. Indeed, given a finite number  $d_1, \dots, d_k$  of such martingales, their sum  $D = d_1 + \dots + d_k$  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  $\alpha_1 d_1 + \dots + \alpha_k d_k$ , with positive rational constants  $\alpha_i$ .

We now need to deal with the general case where we have to defeat *all* total computable martingales simultaneously. What we do is simply add total martingales one by one: we start by diagonalizing against the first total martingale  $d_1$ , then (maybe after a long time) we may introduce the second martingale  $d_2$ , with a small coefficient  $\alpha_2$  (to ensure that introducing  $d_2$  does not cost us too much) and then consider the martingale  $d_1 + \alpha_2 d_2$ . Much later we can introduce the third martingale  $d_3$  with an even smaller coefficient  $\alpha_3$ , 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, and if we add the martingales sufficiently slowly, it is easy to see than we can keep the complexity low (indeed, as long as we know what martingales we are diagonalizing against, the construction is computable; note however that the sequence as a whole will not be computable since whenever we add a martingale we need to store some information, consisting of its code, when it was added and with which coefficient).  $\square$

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  $\text{Succ}(b) \subseteq \text{Succ}(d)$ .*

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  $2^\omega$ , 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:  $\hat{b}(w) = d(w_{\pi(0)} \dots w_{\pi(N-1)})$  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. 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

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

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

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

**Proposition 7 (Buhrman et al [1], Kasternans-Lempp [3]).**

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

As Buhrman et al. [1] explained, it is not true in general that if a total computable injective strategy  $b$  succeeds on a sequence  $A$ , then  $\text{Av}_b$  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, i.e. 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”.

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



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  $\text{Succ}(d) \subseteq \text{Succ}(d')$ . Setting  $b' = (d', \pi)$ , we have  $\text{Succ}(b) \subseteq \text{Succ}(b')$ . By Proposition 7 and Lemma 8,  $d'' = \text{Av}_{b'}$  is a total computable martingale, and

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

as wanted. □

### 3.2 Understanding the strength of injective strategies: the class **TIR**

While the class of computably random sequences (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. To wit, the following theorem is true.

**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:*

$$C(A \upharpoonright n_k \mid k) \leq \log(n_k) - 3 \log(\log(n_k))$$

for infinitely many  $k$ . Then  $A \notin \mathbf{TIR}$

As an immediate corollary, we get the following.

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

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

**Theorem 11.** *For every computable order  $h$  there is a sequence  $A \in \mathbf{TIR}$  such that  $C(A \upharpoonright n \mid n) \leq \log(n) + h(n) + O(1)$  (in particular,  $C(A \upharpoonright n) \leq 2 \log(n) + h(n) + O(1)$ ).*

## 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 12 (Merkle [7]).** *If  $C(A \upharpoonright n) = O(\log n)$  then  $A \notin \mathbf{PMR}$ .*

By this theorem, together with Theorem , we immediately see that the class **PMR** is strictly contained in **TMR**. 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 13 (Muchnik et al. [10]).** *For every computable order  $h$  there is a sequence  $A \in \mathbf{PMR}$  such that, for all  $n \in \mathbb{N}$ ,*

$$C(A \upharpoonright n \mid n) \leq h(n) \log(n) + O(1).$$

#### 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. To wit, the following theorem (quite a contrast to Theorem 13!) holds.

**Theorem 14 (Muchnik [10]).** *If there is a computable order  $h$  such that for all  $n$  we have  $K(A \upharpoonright n) \leq n - h(n) - O(1)$ , then  $A \notin \mathbf{PPR}$ .*

Note that the proof used by Muchnik in [10] works if we replace  $K$  by  $C$  in the above statement. So we now know that any sequence in **PPR** must have infinitely many initial segments of high Kolmogorov complexity. The following theorem shows that some sequences in **PPR** also have infinitely many initial segments of very low complexity.

**Theorem 15.** *For every computable order  $h$  there is a sequence  $A \in \mathbf{PPR}$ , such that there are infinitely many  $n$  where  $C(A \upharpoonright n \mid n) < h(n)$ .*

*Furthermore, if we have an infinite computable set  $S \subseteq \mathbb{N}$ , we can choose the infinitely many lengths  $n$  such that they all are contained in  $S$ .*

The proof of this theorem requires the following “totalization” technique.

**Proposition 16.** *Let  $b = (d, \pi)$  be a partial computable permutation strategy (resp. injective strategy). Let  $\mathcal{C}$  be an effectively closed subset of  $2^\omega$ . 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  $\text{Succ}(b) \cap \mathcal{C} = \text{Succ}(b') \cap \mathcal{C}$ .*

*Proof (of Theorem 15, sketch).* Like for Theorem , we proceed by diagonalization against all partial permutation martingales, “transforming” them into total monotonic martingales in order to carry out the construction. Suppose that during the construction we have constructed an initial segment  $w$  of our infinite sequence by diagonalization against a weighted sum of total monotonic martingales  $\alpha_1 d_1 + \dots + \alpha_k d_k$  (say that the value of this sum at  $w$  is less than 2). Suppose that we now want to add a new partial permutation martingale  $D$ . There are two cases:

1. Either there exists an extension  $v$  of  $w$  such that  $D$  diverges on  $v$ , and such that the value of  $\alpha_1 d_1 + \dots + \alpha_k d_k$  still does not exceed 2 at  $v$ . In that case, we choose  $v$  as an initial segment of our sequence, as it both defeats  $D$  (no matter how we further extend  $v$  later) and keeps  $\alpha_1 d_1 + \dots + \alpha_k d_k$  low.
2. Or, if we are not in the first case, then  $D$  is total on the set  $T$  of sequences  $v$  such that  $\alpha_1 d_1 + \dots + \alpha_k d_k$  has value less than 2 at  $v$  and at all its prefixes. Notice that  $T$  is a computable tree, and thus defines an effectively closed set  $\mathcal{C}$ . By Proposition 16, we can therefore replace  $D$  by a total permutation strategy. Then, by Theorem 5, we can additionally make  $D$  monotonic. The martingale  $d_{k+1}$  we obtain can now be added to the other ones, with a coefficient  $\alpha_{k+1}$  sufficiently small to ensure that  $\alpha_1 d_1 + \dots + \alpha_k d_k + \alpha_{k+1} d_{k+1}$  has value less than 2 at  $w$ , and go on with the diagonalization.

Here again, if we wait for a long time before introducing a new strategy, we can keep the Kolmogorov complexity low. However, we cannot keep it low *all the time* as in the above case 1, the string  $v$  cannot be found effectively (we cannot check that a strategy diverges on a string), and in particular it may have high Kolmogorov complexity.  $\square$

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

**Theorem 17.** *The following statements hold.*

1. **PPR**  $\not\subseteq$  **TIR**
2. **TIR**  $\not\subseteq$  **PMR**
3. **PMR**  $\not\subseteq$  **PPR**

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

*Proof.* 1. Choose a computable sequence  $\{n_k\}_k$  fulfilling the requirements of Theorem 9 such that  $C(k) \leq \log \log n_k$  for all  $k$ . The members of this set then form a computable set  $S$ . Use Theorem 15 to construct a sequence  $A \in$  **PPR** such that  $C(A \upharpoonright n \mid n) < \log \log n$  at infinitely many places in  $S$ . We then have for infinitely many  $k$

$$C(A \upharpoonright n_k \mid k) \leq C(A \upharpoonright n_k) \leq C(A \upharpoonright n_k \mid n_k) + 2 \log \log n_k \leq 3 \log \log n_k,$$

so  $A$  cannot be in **TIR** according to Theorem 9.

	monotonic		permutation		injection
total	<b>TMR</b>	=	<b>TPR</b>	⊇	<b>TIR</b>
	∪†		∪†		∪†
partial	<b>PMR</b>	⊇	<b>PPR</b>	⊇	<b>PIR</b>

**Fig. 2.** Assembled class inclusion results

2. Follows immediately from Theorems 11 and 12.
3. Follows immediately from Theorems 13 and 14. □

**Note:** An extended version of this paper (with full proofs of theorems) can be found at <http://arxiv.org/pdf/0907.2324>.

## 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*. Springer, to appear.
3. Bart Kastermans and Steffen Lemp. Comparing notions of randomness. Manuscript, 2008.
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 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.