George Barmpalias, Andrew Lewis-Pye, Jason Teutsch

Lower bounds on the redundancy in computations from random oracles via betting strategies with restricted wagers

Article (Accepted version) (Refereed)

Original citation:
Barmpalias, George, Lewis-Pye, Andy and Teutsch, Jason (2016) *Lower bounds on the redundancy in computations from random oracles via betting strategies with restricted wagers.* Information and Computation, 251. pp. 287-300. ISSN 0890-5401

DOI: 10.1016/j.ic.2016.09.010

Lower bounds on the redundancy in computations from random oracles via betting strategies with restricted wagers*

George Barmpalias Andrew Lewis-Pye Jason Teutsch

September 21, 2016

Abstract. The Kučera-Gács theorem [Kuč85, Gác86] is a landmark result in algorithmic randomness asserting that every real is computable from a Martin-Löf random real. If the computation of the first n bits of a sequence requires $n + h(n)$ bits of the random oracle, then h is the *redundancy* of the computation. Kučera implicitly achieved redundancy $n \log n$ while Gács used a more elaborate coding procedure which achieves redundancy $\sqrt{n} \log n$. A similar bound is implicit in the later proof by Merkle and Mihailović [MM04]. In this paper we obtain optimal strict lower bounds on the redundancy in computations from Martin-Löf random oracles. We show that any nondecreasing computable function g such that $\sum_n 2^{-g(n)} = \infty$ is not a general upper bound on the redundancy in computations from Martin-Löf random oracles. In fact, there exists a real X such that the redundancy g of any computation of X from a Martin-Löf random oracle satisfies $\sum_n 2^{-g(n)} < \infty$. Moreover, the class of such reals is comeager and includes a Δ^0_2 real as well as all weakly 2-generic reals. On the other hand, it has been recently shown in [BLP16] that any real is computable from a Martin-Löf random oracle with redundancy g, provided that g is a computable nondecreasing function such that $\sum_n 2^{-g(n)} < \infty$. Hence our lower bound is optimal, and excludes many slow growing functions such as $\log n$ from bounding the redundancy in computations from random oracles for a large class of reals. Our results are obtained as an application of a theory of effective betting strategies with restricted wagers which we develop.

George Barmpalias
State Key Lab of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China. School of Mathematics, Statistics and Operations Research, Victoria University of Wellington, New Zealand. *E-mail:* barmpalias@gmail.com. *Web:* http://barmpalias.net

Andrew Lewis-Pye
Department of Mathematics, Columbia House, London School of Economics, Houghton St., London, WC2A 2AE, United Kingdom. *E-mail:* A.Lewis7@lse.ac.uk. *Web:* http://aemlewis.co.uk

Jason Teutsch
Department of Computer and Information Sciences, University of Alabama at Birmingham.
E-mail: teutsch@uab.edu. *Web:* http://people.cs.uchicago.edu/~teutsch

*Barmpalias was supported by the 1000 Talents Program for Young Scholars from the Chinese Government, grant no. D1101130. Additional support was received by the Chinese Academy of Sciences (CAS) and the Institute of Software of the CAS. Lewis-Pye was supported by a Royal Society University Research Fellowship. We thank the two anonymous referees and Peter Gács for feedback that improved the presentation of this article.

1 Introduction

Every sequence is computable from a sequence which is random in the sense of Martin-Löf [ML66]. This major result in algorithmic information theory is known as the Kučera-Gács theorem and was proved by Kučera [Kuč85, Kuč89] and Gács [Gác86]. Both authors showed that the use of the oracle in these reductions can be bounded above by a computable function, but Kučera did not focus on minimizing the number of bits of the oracle that are needed to compute the first n bits of the sequence. If the latter number is $n + h(n)$, we say that the computation has *redundancy* h. A close look at Kučera's argument shows that his techniques achieve redundancy $n \log n$. Gács, on the other hand, took special care to minimise the oracle use. His argument produces a slightly more elaborate computation with redundancy $3\sqrt{n}\log n$, which can easily be improved to $\sqrt{n}\log n$. Both of the arguments were formulated in terms of effective measure, i.e. according to the Martin-Löf definition of randomness.

The major difference between the results of Kučera and Gács is that the latter provides a reduction with oracle use $n + \mathbf{o}(n)$ while the former does not. Merkle and Mihailović [MM04] presented a proof in terms of effective martingales, using similar ideas to Gács' proof but expressed in terms of betting strategies. Up to now, the only known strict lower bound on the redundancy in computation from Martin-Löf random reals is the constant bound, and is due to Downey and Hirschfeldt [DH10, Theorem 9.13.2]. Turing reductions with constant redundancy are also known as *computably Lipschitz* or *cl* reductions and are well studied in computability theory, e.g. see [DH10, Chapter 9]. Downey and Hirschfeldt showed that the redundancy in the Kučera-Gács theorem cannot be $\mathbf{O}(1)$. In fact, they constructed a sequence which is not computed with constant redundancy by any real whose Kolmogorov complexity is bounded below by a computable nondecreasing unbounded function. The reals with the latter property are sometimes known as *complex reals*. A close look at this argument reveals that the set of reals which cannot be computed from any complex real with constant redundancy is comeager. Moreover, it follows from the effective nature of the argument that:

a weakly 2-generic real cannot be computed by any complex real with constant redundancy,

where a real is called *weakly 2-generic* if it has a prefix in every dense Σ^0_2 set of strings.

By [BV11] a real which is not complex has infinitely many initial segments of trivial complexity in the sense that $C(X \upharpoonright_n) = C(n) + \mathbf{O}(1)$ and $K(X \upharpoonright_n) = K(n) + \mathbf{O}(1)$, where K and C denote the prefix-free and plain Kolmogorov complexities. Sequences with the latter property are known as *infinitely often C-trivial and K-trivial* respectively. It follows that any sequence computing a weakly 2-generic sequence with constant redundancy is infinitely often C-trivial and infinitely often K-trivial.

1.1 Our results, in context

In Section 3 we show that the redundancy in computations from Martin-Löf random oracles cannot be bounded by certain slow growing functions. Recall that a real is Δ^0_2 if and only if it is computable from the halting problem.

Theorem 1.1. *There exists a real X such that* $\sum_i 2^{-g(i)} < \infty$ *for every nondecreasing computable function g for which there exists a Martin-Löf random real Y which computes X with redundancy g. In fact, the reals X with this property form a comeager class which includes every weakly 2-generic real.*

This result implies that any nondecreasing computable function g such that $\sum_i 2^{-g(i)} = \infty$ is not a general upper bound on the redundancy in computations of reals from Martin-Löf random oracles. A typical function with this property is $\lceil \log n \rceil$, so the Kučera-Gács theorem does not hold with redundancy $\lceil \log n \rceil$. On the other hand, if $g(n) = 2 \cdot \lceil \log n \rceil$ then $\sum_i 2^{-g(i)} < \infty$. It was recently shown in [BLP16] that any nondecreasing computable function g with the latter property is a general upper bound on the redundancy in computations of reals from Martin-Löf random oracles. Hence Theorem 1.1 is optimal and gives a characterization of the computable nondecreasing redundancy upper bounds in computations of reals from Martin-Löf random oracles. Note that the optimal bounds obtained in [BLP16] are exponentially smaller than the previously best known upper bound of $\sqrt{n} \log n$ from Gács [Gác86].

With slightly more effort we also obtain an effective version of Theorem 1.1, which gives many more examples of reals X which can only be computed from random oracles with large redundancy. Recall that the halting problem relative to A is denoted A'. The generalized non-low$_2$ reals are an important and extensively studied class in the context of degree theory: A is generalized low$_2$ if A'' has the same Turing degree as $(A \oplus \emptyset')'$, and a set which is not generalized low$_2$ is called generalized non-low$_2$.

Theorem 1.2 (Jump hierarchy)**.** *Every set which is generalized non-low$_2$ (including the halting problem) computes a real X with the properties of Theorem 1.1.*

The proof of Theorem 1.1 also gives a nonuniform version of the latter result, requiring a weaker condition regarding the computational power of the oracle. Recall from [DJS96] that a set A is array noncomputable if for each function f that is computable from the halting problem with a computable upper bound on the oracle use, there exists a function h which is computable from A and which is not dominated by f. A degree is array noncomputable if its members are. The class of array noncomputable degrees (again an extensively studied class) is an upwards closed superclass of the generalized non-low$_2$ degrees, and includes low degrees amongst its members.

Theorem 1.3 (Array noncomputability)**.** *Suppose that $\sum_i 2^{-g(i)} = \infty$ for some computable nondecreasing function g. Then every array noncomputable real computes a real X which is not computable by any Martin-Löf real with redundancy g.*

The proof of all of the above results relies on an analysis of effective betting strategies with restricted wagers. This is not entirely surprising as (a) Martin-Löf randomness can be expressed in terms of the success of effective martingales (see Section 2.1) and (b) there is a direct connection between Turing reductions, semi-measures and martingales, which goes back to Levin and Zvonkin [ZL70] (see the discussion before Section 3.1). A strategy (or martingale) is said to have restricted wagers when it can only bet amounts from a given set of possible values, where this set of legitimate values may be allowed to vary from stage to stage of the betting game. The subject of martingales with restricted wagers has been the focus of intense research activity recently. The simplest case is when the restriction specifies only a minimum amount that the gambler can bet at each stage. Integer-valued martingales are examples of strategies of this type, and were motivated and studied by Bienvenu, Stephan and Teutsch [BST10, BST12], Chalcraft, Dougherty, Freiling, and Teutsch [CDFT12], Teutsch [Teu14], Barmpalias, Downey, and McInerney [BDM15] and most recently Herbert [Her16]. A more general study of betting strategies with restricted wagers can be found in Peretz [Per15] and Bavly and Peretz [PB15]. Given a function g, a function on binary strings is called *g-granular* if its value on any string σ

is an integer multiple of $2^{-g(|\sigma|)}$. The notion of g-granular supermartingales is based on the above notion, and is a formalisation of the intuitive notion of betting strategies with restricted wagers. We defer the formal definition until Section 2.2, but state the following pleasing result now, which indicates their importance. Let λ denote the empty string. The definition of c.e. supermartingales and other basic terms will be reviewed in Section 2.1.

Theorem 1.4 (Granular supermartingales)**.** *Suppose that g is a nondecreasing and computable function.*

(a) If $\sum_i 2^{-g(i)} < \infty$, for every c.e. supermartingale N there exists a g-granular c.e. supermartingale M such that for each X we have $\limsup_s M(X \upharpoonright_n) = \infty$ if and only if $\limsup_s N(X \upharpoonright_n) = \infty$.

(b) If $\sum_i 2^{-g(i)} = \infty$, there exists a c.e. supermartingale N such that for all g-granular c.e. supermartingales M there exists some X such that $\limsup_s N(X \upharpoonright_n) = \infty$ and $\limsup_s M(X \upharpoonright_n) < \infty$.

Informally, the first clause of Theorem 1.4 expresses the fact that if $\sum_i 2^{-g(i)} < \infty$ for a computable nondecreasing function g, then g-granular supermartingales suffice for the definition of Martin-Löf randomness. The second clause of Theorem 1.4 says that, in fact, $\sum_i 2^{-g(i)} < \infty$ is also a necessary condition for the sufficiency of g-granular supermartingales for the purpose of defining Martin-Löf randomness. The proofs of Theorem 1.1, Theorem 1.2 and Theorem 1.3 rely on clause (b) of Theorem 1.4, and more specifically on the following more detailed version of this statement, which is of independent interest.

Lemma 1.5. *Suppose that nondecreasing g is computable and $\sum_i 2^{-g(i)} = \infty$. Given any g-granular c.e. supermartingale M there exists a $(g+1)$-granular c.e. supermartingale N and a real X which is computable from M, such that $\limsup_n M(X \upharpoonright_n) \leq M(\lambda)$ and $\limsup_n N(X \upharpoonright_n) = \infty$.*

Lemma 1.5 clearly implies clause (b) of Theorem 1.4, since it implies that the universal c.e. supermartingale will satisfy Theorem 1.4 (b). However it is stronger than the latter, because the supermartingale N is said to be $(g+1)$-granular, i.e. just a single step more granular than the given supermartingale M.

1.2 Further related work in the literature

The present work is a step towards characterizing the optimal redundancy that can be achieved through a general process for coding reals into Martin-Löf random reals, which was completed in [BLP16]. Doty [Dot06] revisited the Kučera-Gács theorem from the viewpoint of constructive dimension. He characterized the optimal asymptotic ratio between n and the use on argument n when a random oracle computes X, in terms of the constructive dimension of X. Recall that the effective packing dimension of a real can be defined as

$$\mathrm{Dim}(X) = \limsup_n \frac{K(X \upharpoonright_n)}{n}.$$

Doty [Dot06] showed that the number of bits of a random oracle needed to compute $X \upharpoonright_n$ is at most $\mathrm{Dim}(X) \cdot n + \mathbf{o}(n)$. So for any real X with $\mathrm{Dim}(X) < 1$, its redundancy is negative on almost all of its prefixes. On the other hand, any Martin-Löf random real has redundancy 0 since it reduces to itself. Thus, Theorem 1.1 refers to reals that are non-random, but that have effective packing dimension 1. One difference between Doty's work and our project is that we are looking to characterize the redundancy that is possible *for every sequence* regardless its effective dimension. A second difference with the work in [Dot06] (as well as [MM04]) is that we are interested in precise bounds on the redundancy of computations from Martin-Löf random reals, rather than just the asymptotic ratio between n and the use on argument n.

Asymptotic conditions on the redundancy g in computations from random oracles such as the ones in Theorem 1.4, have been used with respect to Chaitin's Ω in Tadaki [Tad09] and Barmpalias, Fang and Lewis-Pye [BFLP16]. However the latter work only refers to computations of computably enumerable sets and reals and does not have essential connections with the present work, except perhaps for some apparent analogy of the statements proved.

2 Betting strategies with restricted wagers

The proof of our main result, Theorem 1.1, relies substantially on a lemma concerning effective betting strategies, as formalised by martingales. This section is devoted to proving that lemma, but is also a contribution to the study of strategies with restricted wagers. We are interested in strategies where the wager at step s of the game must be an integer multiple of a rational number which is a function of s. In the next subsection we summarise some required background material.

2.1 Algorithmic randomness and effective martingales

The three main approaches to the definition of algorithmically random sequences are based on (a) incompressibility (Kolmogorov complexity), (b) unpredictability (effective betting strategies) and (c) measure theory (effective statistical tests). There are direct translations between any pair of (a), (b) or (c), and most notions of algorithmic randomness (of various strengths) are naturally defined via any of these approaches. The first two approaches are most relevant to the present work.

Informally, the Kolmogorov complexity of a string is the length of its shortest description. The concept of *description* is formalised via the use of a Turing machine V. Given V, we say that σ is a description of τ if $V(\sigma)$ is defined and equal to τ. There are different versions of Kolmogorov complexity that may be considered, depending on the type of machine that is used in order to formalise the concept of a description. Prefix-free complexity, based on prefix-free machines, is just one way to approach algorithmic randomness, and is the notion of complexity that we shall use in order to obtain our results here. Note, however, that our main results concern only the robust concept of Martin-Löf randomness, which can be defined equivalently with respect to a number of different machine models (or more generally via a number of diverse approaches, as we discuss in the following). A set of binary strings is prefix-free if it does not contain any pair of distinct strings such that one is an extension of the other. A prefix-free Turing machine is a Turing machine with domain which is a prefix-free subset of the finite binary strings. The prefix-free Kolmogorov complexity of a string σ with respect to a prefix-free Turing machine N, denoted K_N, is the length of the shortest string τ such that $N(\tau)\downarrow=\sigma$. Let (N_e) be an effective list of all prefix-free machines. Prefix-free Kolmogorov complexity is based on the existence of an optimal universal prefix-free machine U i.e. such that K_U is minimal, modulo a constant, amongst all K_{N_e}. For the duration of this paper, we adopt a standard choice for U, which is defined by

$U(0^e * 1 * \sigma) \simeq N_e(\sigma)$ (where '$\simeq$' means that one side is defined iff the other is, and that if defined the two sides are equal). From this definition it follows immediately that $K_U(\sigma) \leq K_{N_e}(\sigma) + e + 1$ for all σ and all e. Clearly U is a universal prefix-free machine which can simulate any other prefix-free machine with only a constant overhead, the size of its index. For simplicity we let $K(\sigma)$ denote $K_U(\sigma)$, i.e. when the underlying prefix-free machine is the default U, we suppress the subscript in the notation of Kolmogorov complexity. We identify subsets of $\mathbb{N}$ with their characteristic functions, viewed as an infinite binary sequences, and often refer to them as *reals*. Given a real A, we let $A \upharpoonright_n$ denote the first n bits of A. The algorithmic randomness of infinite binary sequences is often defined in terms of prefix-free Kolmogorov complexity. We say that an infinite binary sequence A is 1-random if there exists a constant c such that $K(A \upharpoonright_n) \geq n - c$ for all n. Informally, these are the infinite sequences for which all initial segments are incompressible.

An equivalent definition of algorithmic randomness for reals can be given in terms of effective statistical tests [ML66]. A Martin-Löf test is an effective sequence of Σ^0_1 classes (V_e) (which we may view as a uniformly c.e. sequence of sets of strings) such that $\mu(V_e) < 2^{-e}$ for each e. A real X is Martin-Löf random if $X \notin \cap_e V_e$ for any Martin-Löf test (V_e). A third way to define algorithmic randomness, due to Schnorr [Sch71b, Sch71a], can be given in terms of betting strategies, normally formalised as martingales or supermartingales. We are interested in supermartingales as functions $h : 2^{<\omega} \to \mathbb{R}^{\geq 0}$ with the property $h(\sigma 0) + h(\sigma 1) \leq 2h(\sigma)$. A supermartingale such that $h(\sigma 0) + h(\sigma 1) = 2h(\sigma)$ for all σ is called a martingale. We say that:

$$\text{the supermartingale } h \textit{ succeeds on a real } X \text{ if } \limsup_s h(X \upharpoonright_n) = \infty.$$

Note that a stronger notion of success is the condition $\lim_s h(X \upharpoonright_n) = \infty$. In many situations, such as in the characterization of Martin-Löf random sequences in terms of martingales (see below), it is not important which notion of success is used. In the present work, however, it seems more appropriate to use the weaker notion as a default, and to mention the stronger notion explicitly when it plays a role in an argument. We say that a function $f : 2^{<\omega} \to \mathbb{R}^{\geq 0}$ is left-c.e. if there is a computable function $f_0 : 2^{<\omega} \times \mathbb{N} \to \mathbb{Q}^{\geq 0}$ which is nondecreasing in the second argument and such that $f(\sigma) = \lim_s f_0(\sigma, s)$ for each σ. In this case the function f_0 is called the left-c.e. approximation to f. A (super)martingale is called c.e. if it is left-c.e. as a function. It is a well known fact, due to Schnorr [Sch71b, Sch71a] (see for example [DH10, Theorems 6.2.3, 6.3.4]), that the following are equivalent for each real X:

(i) X is Martin-Löf random;

(ii) no c.e. supermartingale succeeds on X;

(iii) $K(X \upharpoonright_n) \geq n - c$ for some constant c and all n.

In fact this equivalence is effective, in the sense of Lemma 2.1. Recall that λ denotes the empty string. The weight of a prefix-free set of strings S is $\sum_{\sigma \in S} 2^{-|\sigma|}$, and is equal to the measure of the Σ^0_1 class of reals represented by S, i.e. the reals that have a prefix in S.

Lemma 2.1 (Schnorr, implicit in [Sch71b, Sch71a])**.** *Given the index for a c.e. supermartingale M, $m \geq M(\lambda)$ and $c \in \mathbb{N}$, one can effectively find k for which the following holds: for any finite string σ, if $M(\sigma) \geq k$, then $K(\sigma) \leq |\sigma| - c$.*

Proof. Given a supermartingale M with $M(\lambda) \leq m$, by Kolmogorov's inequality the measure of reals X for which there exists n such that $M(X \upharpoonright_n) \geq k$ is bounded above by m/k. On the other hand, given an integer c and a prefix-free and c.e. set of finite strings V such that the weight of V is bounded above by 2^{-c}, we can effectively define a prefix-free machine N such that $K_N(\sigma) \leq |\sigma| - c$ for all $\sigma \in V$ (this is a typical application of the so-called Kraft-Chaitin online algorithm for the construction of a prefix-free machine). The crucial point is that given a c.e. supermartingale M and k, the set of reals X such that $M(X \upharpoonright_n) \geq k$ for some n, is a Σ^0_1 class.

For each k let $V(k)$ be the set of all of those strings σ for which $M(\sigma) > k$. Note that the measure of $V(k)$ is bounded above by $m \cdot k^{-1}$. We can effectively find k as required by the lemma, via the recursion theorem (and its uniformity) as follows. We construct a prefix-free machine N, and by the recursion theorem we may use its index b in its definition. We let k be $2^{m+b+c+1}$ and define N via the Kraft-Chaitin online algorithm such that $K_N(\sigma) \leq |\sigma| - c - b - 1$ for all $\sigma \in V(k)$. Since the measure of $V(k)$ is bounded above by $m \cdot 2^{-c-m-b-1} < 2^{-c-b-1}$, the definition of N is valid, and the application of the Kraft-Chaitin online algorithm along with the definition of the sets $V(k)$ ensures that $K_N(\sigma) \leq |\sigma| - c - b - 1$ for all strings σ such that $M(\sigma) \geq k$. But according to our choice of optimal universal machine U this implies that $K(\sigma) \leq |\sigma| - c$ for all strings σ such that $M(\sigma) \geq k$, which concludes the proof. □

Martingales are expressions of betting games on sequences of binary outcomes. More specifically, if h is a martingale, then $h(\sigma)$ can be thought of as expressing the capital of the player betting according to the strategy h, after the sequence σ of outcomes. If at state σ of the game we bet α on 0, then our capital at the next stage will be $h(\sigma 0) = h(\sigma) + \alpha$ or $h(\sigma 1) = h(\sigma) - \alpha$ according to whether the outcome was 0 or 1, respectively. So $h(\sigma 0) + h(\sigma 1) = 2h(\sigma)$. Martingales can therefore be seen as modeling the capital in a betting game along every possible sequence of outcomes. Given a martingale h, the amount that is bet at state σ is $|h(\sigma 0) - h(\sigma)| = |h(\sigma 1) - h(\sigma)|$ and is bet on 0 or 1 according to whether $h(\sigma 0) > h(\sigma 1)$ or not. Hence every martingale determines a betting strategy, which we may regard as a function from strings to the non-negative reals, which determines what amount is bet and on which outcome. Conversely, a betting strategy corresponds to a martingale, which models the remaining capital at the end of each bet. Our definition of granular betting strategies in Section 2.2 relies on the condition that the bets made at each stage (and not necessarily the remaining capital) are granular, in the sense that they correspond to numbers from a specific set. For more detailed background on the notions discussed in this section, we refer the reader to [DH10, Chapter 6]. For a general introduction to algorithmic randomness we refer to [LV97].

2.2 Restricted martingales

Restricting the set of possible betting strategies may give rise to weaker forms of randomness. There are many ways to impose such restrictions, but the method which is relevant to our work involves dictating a minimum wager at each step of the betting process, and requiring that the gambler bets an integer multiple of that minimum wager. We formalise this notion in the following definitions.

Definition 2.2 (Granularity of functions). *Given functions $g : \mathbb{N} \to \mathbb{N}$ and $M : 2^{<\omega} \to \mathbb{R}$, we say that M is g-granular (or has granularity g) if for every string σ the value of $M(\sigma)$ is an integer multiple of $2^{-g(|\sigma|)}$.*

We could now restrict our attention to supermartingales that are g-granular as functions, for some computable non-decreasing function g. Indeed, this approach suffices for most of the results in this paper. However we formalise betting strategies with restricted wagers in a slightly more general way, which is both intuitively justifiable and also allows to prove the rather elegant characterization of Theorem 1.4.

Definition 2.3 (Granular c.e. supermartingales)**.** *Given a nondecreasing computable function $g : \mathbb{N} \to \mathbb{N}$, we say that a c.e. supermartingale M is g-granular if there exists a computable sequence of rationals (q_i) and a g-granular left-c.e. function $N : 2^{<\omega} \to \mathbb{R}$ such that $M(\sigma) = N(\sigma) + \sum_{i \geq |\sigma|} q_i$. In the special case where $f(n) = \sum_{i \geq n} q_i$ is constantly zero we say that M is a strongly g-granular c.e. supermartingale.*

Intuitively speaking, the function f in the above definition represents a part of the capital which is not used for betting, and is transferred from each round to the next round, perhaps reduced due to inflation, in accordance with the standard interpretation of supermartingales as betting strategies. More precisely, the value of f does not depend on the particular bets that we have placed up to a certain stage, but rather on the number of these bets, i.e. the stage of the game. The particular case where f is the zero function is of special importance, as it is the notion that will be used in the proofs of most of the results in this paper. We emphasize the fact that in Definition 2.3 we require N to be a left-c.e. function, and so a c.e. index of a granular c.e. supermartingale M is not merely a program which gives a left-c.e. approximation to M but a program that enumerates the values (q_i) and also gives a left-c.e. approximation N – thereby specifying a left-c.e. approximation to M.

We are ready to present and prove the main result of this section, which is a more elaborate version of Lemma 1.5. Clearly Lemma 2.4 implies Lemma 1.5. However Lemma 2.4 also gives the rate of growth of the supermartingale N as a function of g, which is absent in the statement of Lemma 1.5. Summing up, Lemma 2.4 implies Lemma 1.5, which in turn implies clause (b) of Theorem 1.4. The reason we preceded the following elaborate statement with the two weaker ones, is that the additional technical information may only be of interest to some readers, and may distract others from the main result, namely Theorem 1.4.

Lemma 2.4 (Granular c.e. supermartingales)**.** *Given a string v_0, a nondecreasing computable function $g : \mathbb{N} \to \mathbb{N}$ and a g-granular c.e. supermartingale M, there exists a real $X \supset v_0$ and a $(g+1)$-granular c.e. supermartingale N such that $N(X \upharpoonright_n) \geq \sum_{0 \leq i \leq n} 2^{-g(i)-1}$ and $M(X \upharpoonright_n) \leq M(v_0)$ for all $n \geq |v_0|$. Moreover if M is strongly g-granular, then N can also be chosen to be strongly $(g+1)$-granular.*

Proof. For the sake of ease of notation, we may assume that v_0 is the empty string. The proof of the more general case is a direct adaptation of the proof of this special case. A first naive attempt would be to let N bet in the opposite way to M, which means to define

$$N(\sigma * i) = \begin{cases} N(\sigma) + 2^{-g(|\sigma|+1)} & \text{if } M(\sigma * i) < M(\sigma * (1-i)) \\ N(\sigma) - 2^{-g(|\sigma|+1)} & \text{otherwise} \end{cases}$$

and let X carve a path on the binary tree where N wins (so M loses) at every stage (ignoring for now the possibility that $M(\sigma * 0) = M(\sigma * 1)$). This martingale, however, is not necessarily c.e., because M is merely c.e. and not computable, so the condition $M(\sigma * i) < M(\sigma * (1-i))$ is not decidable. Following the same basic idea (letting N bet on the outcomes where M does not increase its capital) we produce a more sophisticated definition, which defines N as a $(g+1)$-granular c.e. supermartingale.

The idea for this argument is to effectivize the above definition of N so that the resulting function is a c.e. supermartingale. In order to do this, we need to avoid using the condition $M(\sigma * i) < M(\sigma * (1-i))$ in the above definition of N, since it is not decidable. The solution is to incorporate the effective approximations to $M(\sigma * i), M(\sigma * (1-i))$ into the definition of $N(\sigma * i), N(\sigma * (1-i))$ in such a way that we can still gain additional capital by choosing the right value of i. It turns out that we can do this by using an additive term of $2^{-g(|\sigma|+1)-1}$, thus making N a $(g+1)$-granular supermartingale, as indicated in (2.2.3). In the following we formalize this idea, and prove that it works.

Let $(M_s) = (\hat{M}_s + f_s)$ be a left-c.e. approximation to M, such that each M_s is a g-granular supermartingale, $f_s(n) = \sum_{i:n\le i\le s} q_i$ for the computable sequence of rationals q_i, $\hat{M}_s(\sigma)$ is a g-granular function for each s and such that $\hat{M}_s(\sigma)$ is nondecreasing as a function of s. The reader may find the proof more tractable if they assume f to be constantly zero. This corresponds to the case of the lemma regarding strongly granular supermartingales and contains all the important ideas of the general proof. For completeness, however, we present the full argument here. There exist left-c.e. integer-valued functions $t : 2^{<\omega} \to \mathbb{N}$, $q : 2^{<\omega} \to \mathbb{N}$ with left-c.e. approximations t_s, q_s such that:

$$\hat{M}_s(\sigma * 0) = t_s(\sigma) \cdot 2^{-g(|\sigma|+1)} \quad \text{and} \quad \hat{M}_s(\sigma * 1) = q_s(\sigma) \cdot 2^{-g(|\sigma|+1)} \tag{2.2.1}$$

for all σ, s. Recall that λ denotes the empty string. We will define a computable sequence of supermartingales (N_s), which is also a left-c.e. approximation to their limit N, a c.e. supermartingale. In fact, we will define a computable sequence of $(g+1)$-granular functions $(\hat{N}_s)$ such that the functions $N_s(\sigma) := \hat{N}_s(\sigma) + f_s(|\sigma|)$ are computable supermartingales. Then clearly the limit N of N_s will be a supermartingale and by Definition 2.3, the function N will also be a $(g+1)$-granular supermartingale. Let $\hat{N}_s(\lambda) = \hat{M}_s(\lambda) + 2^{-g(0)-1}$ for all stages s and let $\hat{N}_0(\sigma) = 0$ for all nonempty strings σ. The values of $\hat{N}_s(\sigma)$ for $s > 0$ and nonempty strings σ are defined inductively as follows. We order the strings first by length and then lexicographically. The notion of accessibility is defined dynamically during the construction. At stage 0, no string has been accessed.

Construction of $\hat{N}_s$. At each stage $s+1$, if $\hat{M}_{s+1}(\lambda) \neq \hat{M}_s(\lambda)$ then do nothing other than define $\hat{N}_{s+1}(\lambda) = \hat{M}_{s+1}(\lambda) + 2^{-g(0)-1}$ and $\hat{N}_{s+1}(\tau) = \hat{N}_s(\tau)$ for all $\tau \neq \lambda$ (g-granularity means this can only occur at finitely many stages). Otherwise, find the least string σ of length at most s, such that for all $\eta \subseteq \sigma$,

$$\hat{N}_s(\eta) \ge \hat{M}_s(\eta) + \sum_{i\le|\eta|} 2^{-g(i)-1} \tag{2.2.2}$$

and one of the following clauses holds:

(a) σ has not been accessed at any stage $\le s$;

(b) σ was last accessed at stage $m < s+1$ and either $t_{s+1}(\sigma) \neq t_m(\sigma)$ or $q_{s+1}(\sigma) \neq q_m(\sigma)$.

If such a string does not exist, let $\hat{N}_{s+1}(\eta) = \hat{N}_{s+1}(\eta)$ for all strings η. Otherwise define:

$$\begin{cases} \hat{N}_{s+1}(\sigma * 0) = & \sum_{i\le|\sigma|} 2^{-g(i)-1} + q_{s+1}(\sigma) \cdot 2^{-g(|\sigma|+1)} + 2^{-g(|\sigma|+1)-1} \\ \hat{N}_{s+1}(\sigma * 1) = & \sum_{i\le|\sigma|} 2^{-g(i)-1} + t_{s+1}(\sigma) \cdot 2^{-g(|\sigma|+1)} - 2^{-g(|\sigma|+1)-1} \end{cases} \tag{2.2.3}$$

and declare that σ has been accessed at stage $s+1$. Note that in this case we have $\hat{M}_s(\sigma) = \hat{M}_{s+1}(\sigma)$, because if this was not true and η is the immediate predecessor of σ, then $t_{s+1}(\eta) \neq t_s(\eta)$ or $q_{s+1}(\eta) \neq q_s(\eta)$, which contradicts the minimality of σ.

For τ other than $\sigma * 0$ and $\sigma * 1$ define $\hat{N}_{s+1}(\tau) = \hat{N}_s(\tau)$. Note also that (a) the roles of q and t are reversed in the above definition in the sense that q_{s+1} is used in the definition of $\hat{N}_{s+1}(\sigma * 0)$ rather than $\hat{N}_{s+1}(\sigma * 1)$, and (b) the definitions of $\hat{N}_{s+1}(\sigma * 0)$ and $\hat{N}_{s+1}(\sigma * 1)$ are *not* symmetrical, since we add $2^{-g(|\sigma|+1)-1}$ in defining the former value, while we subtract it in defining the latter.

This concludes the construction of the functions $\hat{N}_s$ and we let $N_s(\sigma) = \hat{N}_s(\sigma) + f_s(|\sigma|)$ for all σ. We also let $N(\sigma) = \lim_s N_s(\sigma)$ for all σ.

Intuition for the construction. The driving force behind the construction is (2.2.2), which is guaranteed to hold for the empty string, but not for all strings. However, as we are going to verify in the following, inductively we can argue that there is real X such that all of its initial segments η satisfy (2.2.2). The updates defined in (2.2.3) ensure that (2.2.2) continues to hold for at least one immediate extension of σ. The updates are made gradually, following the approximations to M, in order to ensure that N is a c.e. supermartingale. Moreover the equations in the update mechanism (2.2.3) will ensure that N is a $(g + 1)$-granular supermartingale, as required.

Verification. The fact that g is nondecreasing means that $\hat{N}_t$ can never take negative values (in particular the term $-2^{-g(|\sigma|+1)-1}$ in the definition of $\hat{N}_{t+1}(\sigma * 1)$ cannot cause negative values). By (2.2.3) and the fact that $t_s(\sigma), q_s(\sigma)$ are nondecreasing we have that

$$\hat{N}_t(\sigma) \leq \hat{N}_{t+1}(\sigma) \quad \text{ for all } t, \sigma. \tag{2.2.4}$$

Since $N_s(\sigma) = \hat{N}_s(\sigma) + f_s(|\sigma|)$, and (f_s) is a left-c.e. approximation to the function f, it follows that (N_s) is a left-c.e. approximation to the limit N of (N_s). Hence N is a left-c.e. function.

Next, we verify that each N_t is a supermartingale. We must show that for all t:

$$N_t(\sigma * 0) + N_t(\sigma * 1) \leq 2 \cdot N_t(\sigma). \tag{2.2.5}$$

For $t = 0$ this property clearly holds. Given any $t > 0$, consider the largest $s + 1 \leq t$ at which σ was accessed during the construction. If such stage does not exist, then $\hat{N}_t(\sigma * 0) = \hat{N}_t(\sigma * 1) = 0$ and (2.2.5) holds by the monotonicity of f and its approximations f_s. Otherwise, according to the construction, and in particular (2.2.3), we must have

$$\hat{N}_s(\sigma) \geq \hat{M}_s(\sigma) + \sum_{i \leq |\sigma|} 2^{-g(i)-1} \quad \text{and} \quad \hat{N}_{s+1}(\sigma) \geq \hat{M}_{s+1}(\sigma) + \sum_{i \leq |\sigma|} 2^{-g(i)-1} \tag{2.2.6}$$

where the second inequality holds because $\hat{N}_{s+1}(\sigma) = \hat{N}_s(\sigma)$ (since σ was accessed at $s + 1$ and not any of its predecessors) and $\hat{M}_{s+1}(\sigma) = \hat{M}_s(\sigma)$ (because otherwise a predecessor of σ would have been accessed at stage $s + 1$, or else $\sigma = \lambda$ and σ would not have been accessed at stage $s + 1$). Moreover by the choice of s we have $\hat{N}_{s+1}(\sigma * 0) = \hat{N}_t(\sigma * 0), \hat{N}_{s+1}(\sigma * 1) = \hat{N}_t(\sigma * 1)$. Hence

$$N_t(\sigma * 0) + N_t(\sigma * 1) = \hat{N}_{s+1}(\sigma * 0) + \hat{N}_{s+1}(\sigma * 1) + 2 \cdot f_t(|\sigma| + 1). \tag{2.2.7}$$

According to (2.2.3) we have

$$N_{s+1}(\sigma * 0) + N_{s+1}(\sigma * 1) \leq 2 \cdot \Big(\sum_{i \leq |\sigma|} 2^{-g(i)-1} \Big) + 2^{-g(|\sigma|+1)} \cdot \Big(t_{s+1}(\sigma) + q_{s+1}(\sigma) \Big) + 2 \cdot f_{s+1}(|\sigma| + 1). \tag{2.2.8}$$

By (2.2.1) and the fact that M_{s+1} is a supermartingale we have:

$$2^{-g(|\sigma|+1)}\cdot\big(t_{s+1}(\sigma)+q_{s+1}(\sigma)\big)+2\cdot f_{s+1}(|\sigma|+1) = \hat{M}_{s+1}(\sigma*0)+\hat{M}_{s+1}(\sigma*1)+2\cdot f_{s+1}(|\sigma|+1) \leq 2\cdot\hat{M}_{s+1}(\sigma)+2\cdot f_{s+1}(|\sigma|)$$

so plugging this back to (2.2.8) we get

$$N_{s+1}(\sigma * 0) + N_{s+1}(\sigma * 1) \leq 2 \cdot \Big(\sum_{i \leq |\sigma|} 2^{-g(i)-1}\Big) + 2 \cdot \hat{M}_{s+1}(\sigma) + 2 \cdot f_{s+1}(|\sigma|).$$

Then applying the second inequality of (2.2.6) to the preceding inequality, we get

$$N_{s+1}(\sigma * 0) + N_{s+1}(\sigma * 1) \leq 2 \cdot (\hat{N}_{s+1}(\sigma) + f_{s+1}(|\sigma|)) = 2N_{s+1}(\sigma).$$

Since $\hat{N}_{s+1}(\sigma*0) = \hat{N}_t(\sigma*0), \hat{N}_{s+1}(\sigma*1) = \hat{N}_t(\sigma*1)$ and since $f_t(|\sigma|)-f_{s+1}(|\sigma|) \geq f_t(|\sigma|+1)-f_{s+1}(|\sigma|+1)$ this gives:

$$N_t(\sigma * 0) + N_t(\sigma * 1) \leq 2 \cdot N_t(\sigma).$$

Hence for each t the function N_t is a computable supermartingale. By (2.2.4) and the fact that (f_s) is a left-c.e. approximation to f, it follows that (N_s) is a left-c.e. approximation to N. Hence N is a left-c.e. supermartingale. In order to establish that N is a $(g + 1)$-granular supermartingale, recall that M is a g-granular supermartingale, and for each σ the integer parameters $t_s(\sigma), q_s(\sigma)$ are nondecreasing and reach a limit after finitely many stages. Note that the only redefinition of $\hat{N}_{s+1}$ in the construction occurs through (2.2.3). This, and the fact that $t_s(\sigma), q_s(\sigma)$ are integers, shows that each $\hat{N}_t$ is a $(g + 1)$-granular function. Hence the limit $\hat{N}$ of $(\hat{N}_s)$ is also $(g + 1)$-granular. Then by Definition 2.3 it follows that N is a $(g + 1)$-granular c.e. supermartingale.

It remains to show that there exists a real X such that $N(X \restriction_n) \geq \sum_{i\leq n} 2^{-g(i)-1}$ and $M(X \restriction_n) \leq M(\lambda)$ for all $n \geq 0$. By the definition of M, N and the fact that f is non-negative and nonincreasing, it suffices to show that there exists a real X such that $\hat{N}(X \restriction_n) \geq \sum_{i\leq n} 2^{-g(i)-1}$ and $\hat{M}(X \restriction_n) \leq \hat{M}(\lambda)$ for all $n \geq 0$. The idea is as we described it at the beginning of the proof, i.e. to let X follow the path where M does not increase its capital. Define X inductively as follows. Given $X \restriction_n$ define:

$$X(n) = \begin{cases} 0 & \text{if } \hat{M}(X \restriction_n *0) \leq \hat{M}(X \restriction_n *1) \\ 1 & \text{if } \hat{M}(X \restriction_n *0) > \hat{M}(X \restriction_n *1). \end{cases}$$

We shall establish the stronger condition that:

$$\hat{N}(X \restriction_n) \geq \sum_{i\leq n} 2^{-g(i)-1} + \hat{M}(X \restriction_n) \quad \text{for all } n. \tag{2.2.9}$$

We prove this by induction on n. It is clear that the claim holds for $n = 0$. Let $T_m = \lim_s t_s(X \restriction_m)$ and $Q_m = \lim_s q_s(X \restriction_m)$ for each m, and suppose that (2.2.9) holds for n. Suppose first that $T_n < Q_n$, so that $\hat{M}$ may be thought of as betting that $X(n) = 1$, while $\hat{N}$ guesses correctly that $X(n) = 0$. In this case it follows from the fact that $\hat{M}$ is a g-granular function that $Q_n \cdot 2^{-g(n+1)} \geq \hat{M}(X \restriction_{n+1}) + 2^{-g(n+1)}$. From (2.2.3) we then have

$$\hat{N}(X \restriction_{n+1}) > \hat{M}(X \restriction_{n+1}) + \sum_{i\leq n+1} 2^{-g(i)-1}.$$

Suppose next that $T_n = Q_n$. In this case we still have $X(n) = 0$, but now $Q_n \cdot 2^{-g(n+1)} = \hat{M}(X \restriction_{n+1})$. The final term $2^{-g(n+1)-1}$ in (2.2.3), however, means that (2.2.9) still holds. Suppose finally that $T_n > Q_n$, so that $X(n) = 1$. Then $T_n \cdot 2^{-g(n+1)} \geq \hat{M}(X \upharpoonright_{n+1}) + 2^{-g(n+1)}$, so then even though we subtract $2^{-g(n+1)-1}$ in (2.2.3), we may again conclude that (2.2.9) holds. This completes the inductive step and the proof of (2.2.9). Finally, it is clear from the above argument that if f is constantly zero then N is a strongly g-granular c.e. supermartingale. This shows the latter clause of the lemma. □

We make three observations regarding Lemma 2.4, which follow from its proof. First, not only does N succeed on X, but it does so in an essentially monotonic fashion, in the sense of (2.2.9). Second, N is obtained uniformly from M, in the sense that there is a computable function which, given a c.e. index for M (i.e. a program which produces left-c.e. approximations to $\hat{M}$ and f), produces a c.e. index for N with the prescribed properties. Finally, the real X is computable from M, which is a left-c.e. function. Therefore X is computable from the halting problem. Note that Lemma 1.5 is a special case of Lemma 2.4 when $\sum_i 2^{-g(i)} = \infty$.

2.3 Granular supermartingales and effective randomness

In this section we give a proof of Theorem 1.4. For clause (b) of Theorem 1.4, suppose that we are given g with the assumed properties. Consider the universal c.e. supermartingale N. By Lemma 1.5, given any g-granular supermartingale M we can find X such that $\limsup_n M(X \upharpoonright_n)$ is finite while $\limsup_n N_*(X \upharpoonright_n)$ is infinite for some c.e. supermartingale N_*. By the universality of N, the latter condition implies that $\limsup_n N(X \upharpoonright_n)$ is also infinite, which concludes the proof of clause (b). For clause (a), let N be a c.e. supermartingale. Given positive rational numbers q, p let $\mathcal{S}(q, p)$ be the largest multiple of p which is less than q. For each string σ we define

$$M(\sigma) = \sum_{i>|\sigma|} 2^{-g(i)} + \mathcal{S}(N(\sigma), 2^{-g(|\sigma|)})$$

and note that M is c.e. as a function, because N is a c.e. function. Moreover, M is clearly g-granular, and since N is a supermartingale we have

$$M(\sigma * 0) + M(\sigma * 1) \leq N(\sigma * 0) + N(\sigma * 1) + 2 \cdot \sum_{i>|\sigma|+1} 2^{-g(i)} \leq 2 \cdot \left(N(\sigma) + \sum_{i>|\sigma|+1} 2^{-g(i)} \right)$$

But by the definition of S we have $N(\sigma) \leq \mathcal{S}\left(N(\sigma), 2^{-g(|\sigma|)}\right) + 2^{-g(|\sigma|)}$ so

$$N(\sigma) + \sum_{i>|\sigma|+1} 2^{-g(i)} \leq \mathcal{S}\left(N(\sigma), 2^{-g(|\sigma|)}\right) + \sum_{i>|\sigma|} 2^{-g(i)} = M(\sigma).$$

Hence we may conclude that $M(\sigma * 0) + M(\sigma * 1) \leq M(\sigma)$ for all σ, which means that M is a c.e. g-granular supermartingale. Also note that

$$\sum_{i>|\sigma|} 2^{-g(i)} + N(\sigma) \leq M(\sigma) + 2^{-g(|\sigma|)} \quad \text{and} \quad M(\sigma) \leq \sum_{i>|\sigma|} 2^{-g(i)} + N(\sigma).$$

The first inequality shows that if $\limsup_s N(X \upharpoonright_n) = \infty$ for some X, then $\limsup_s N(X \upharpoonright_n) = \infty$. The second inequality above shows that if $\limsup_s M(X \upharpoonright_n) = \infty$ for some X then $\limsup_s N(X \upharpoonright_n) = \infty$, which concludes the proof of Theorem 1.4.

3 Lower bounds on the redundancy in computation from random reals

In this section we give proofs of Theorems 1.1, 1.2 and 1.3, using the result we now have for restricted betting strategies. We start with the definition of redundancy, following Gács [Gác86].

Definition 3.1 (Oblivious use-function and redundancy)**.** *We say that f is a use-function of the Turing functional Φ if for every X and n, during the computation $\Phi^X(n)$ (whether it halts or not) all bits of X that are queried are smaller than $f(n)$. In this case we say that* $\max\{f(n)-n, 0\}$ *is a redundancy of* Φ.

Note that this definition is oblivious to the oracle X, a choice which reflects the fact that we are interested in general upper bounds for the Kučera-Gács theorem. Clearly, given a Turing functional, there are many choices for its use function and its redundancy. However we are generally interested in minimising the use-function and the redundancy of computations. Moreover, we only consider use-functions and redundancy functions which are computable and nondecreasing. Given a Turing functional Φ with nondecreasing computable use-function f, we may view Φ as a partial computable function which maps strings of length $f(n)$ to strings of length n (for each n). The following fact links Turing reductions with supermartingales.

Lemma 3.2 (Supermartingales from Turing functionals)**.** *Let Φ be a Turing functional with computable nondecreasing redundancy g, and for each string ν let $h(\nu)$ be the number of strings τ of length $|\nu|+g(|\nu|)$ such that $\Phi^\tau = \nu$. Then the function $h^*(\nu) := 2^{-g(|\nu|)} \cdot h(\nu)$ is a strongly g-granular c.e. supermartingale.*

Proof. Since Φ is a Turing functional, we have $h(\nu 0) + h(\nu 1) \le 2^{|\nu|+1+g(|\nu|+1)-|\nu|-g(|\nu|)} \cdot h(\nu)$ and so

$$h(\nu 0) + h(\nu 1) \le 2^{1+g(|\nu|+1)-g(|\nu|)} \cdot h(\nu) \qquad \text{for all strings } \nu.$$

Since h is an integer-valued function, h^* is a g-granular function. Moreover:

$$h^*(\nu 0) + h^*(\nu 1) = 2^{-g(|\nu|+1)} \cdot (h(\nu 0) + h(\nu 1)) \le 2^{-g(|\nu|+1)} \cdot 2^{1+g(|\nu|+1)-g(|\nu|)} \cdot h(\nu) = 2 \cdot h_e^*(\nu).$$

So h^* is a strogly g-granular supermartingale. Finally h is a left-c.e. function, because Φ is a Turing functional. So h^* is a left-c.e. function, which concludes the proof. □

Lemma 3.2 establishes a method for constructing supermartingales from Turing reductions. Restricted wagers in the supermartingales constructed correspond to upper bounds on the oracle-use of the Turing reductions they are built from. Similar arguments have been used in [BL07] and [DH10, Theorem 9.13.2], for the special case of integer-valued martingales and Turing reductions with constant redundancy. The underlying general topic here is the connection between Turing functionals and semimeasures, which was explored in [ZL70]. For a recent account of this topic the reader is referred to [BHPS16], while [DH10] also contains related material in various sections of Chapters 3,6 and 7.

We are now ready to apply Lemma 2.4 in order to prove a density lemma, which will be the basis of an inductive construction specifying the reals required by Theorem 1.1. The proof of Theorem 1.2 also establishes Theorem 1.1, but is slightly more involved than a direct proof of the latter. We therefore choose to give a simple proof of Theorem 1.1 in Section 3.1, before expanding that proof in order to obtain a proof of Theorem 1.2.

3.1 Proof of Theorem 1.1

We use an effective forcing or finite extension argument, based on the following fact.

Lemma 3.3 (Density lemma)**.** *Let Φ be a Turing functional with redundancy a computable nondecreasing function g such that $\sum_i 2^{-g(i)} = \infty$. Given any $c \in \mathbb{N}$ and any finite string ν_0, there exists an extension $\nu \supset \nu_0$ such that $K(\mu) < |\mu| - c$ for every string μ of length $|\nu| + g(|\nu|)$ for which $\Phi^\mu = \nu$.*

Proof. Given the functional Φ, recall the associated functions h, h^* from Lemma 3.2. Since Φ has redundancy g, it follows that h^* is a g-granular c.e. supermartingale. Then given ν_0, by Lemma 2.4 it follows that there exists a constant d, a c.e. supermartingale N and a real $Z \supset \nu_0$, such that $h^*(Z \restriction_n) < 2^d$ for all n and N succeeds on Z . By the characterization of Martin-Löf randomness in terms of c.e. supermartingales, it follows that the real Z is not Martin-Löf random. Moreover since $h^*(Z \restriction_n) < 2^d$ for all n, given the definition of h^* we have that $h(Z \restriction_n) < 2^{d+g(n)}$ for all n. We claim that there exists a prefix-free machine M such that:

$$\forall n \in \mathbb{N}, \mu \in 2^{n+g(n)} \quad (\Phi^\mu = Z \restriction_n \quad \Rightarrow \quad K_M(\mu) \leq K(Z \restriction_n) + g(n) + d)\,. \tag{3.1.1}$$

The machine M is defined in the following self-delimiting way. Given a program σ, M first looks for an initial segment σ_0 of σ which is in the domain of the universal prefix-free machine U. If and when it finds σ_0, M calculates $\tau = U(\sigma_0)$ – one can think of the machine as interpreting this string τ as $Z \restriction_n$. It then calculates $g(n)$ (where n is the length of τ) and reads σ_1, which is the following $g(n) + d$ bits of σ (starting from bit $|\sigma_0| + 1$). If σ does not have sufficiently many bits that σ_1 is defined then M loops indefinitely. Otherwise, M interprets the string σ_1 as a number $t \leq 2^{d+g(n)}$. It then interprets the number t as the priority index of a string μ in the universal enumeration of strings ρ such that $\Phi^\rho = \tau$. In other words, M runs this universal enumeration and starts producing the computably enumerable sequence of strings ρ with $\Phi^\rho = \tau$, stopping at the tth such string μ. If there are less than t many strings ρ such that $\Phi^\rho = \tau$, then M loops indefinitely. Finally M assigns $\sigma_0 * \sigma_1$ as a description of μ (i.e. we define $M(\sigma_0 * \sigma_1) = \mu$). Since U is prefix-free and the length of σ_1 is determined by σ_0, the machine M is prefix-free. Moreover, given a real Z such that $h(Z \restriction_s) < 2^{d+g(s)}$ for all s, M will describe every string μ such that $\Phi^\mu = Z \restriction_n$ with a string of length $K(Z \restriction_n)+g(n)+d$. Indeed, by the property $h(Z \restriction_n) < 2^{d+g(n)}$, if M is given as an input the concatenation of a description of Z_n and a string of length $d + g(n)$ which codes the priority index of string μ in the enumeration of all strings ρ with $\Phi^\rho = Z \restriction_n$, it will follow the steps above, and will eventually output the string μ. This completes the proof of (3.1.1).

We can now use our assumption that Z is not Martin-Löf random in order to complete the proof of the lemma. Since M is a prefix-free machine there exists some constant c_0 such that $K(\rho) < K_M(\rho) + c_0$ for all strings ρ. Since Z is not Martin-Löf random we can choose some $n > |\nu_0|$ such that $K(Z \restriction_n) < n - c_0 - d - c$. Then given any string μ of length $n + g(n)$ such that $\Phi_e^\mu = Z \restriction_n$, according to (3.1.1) we have:

$$K(\mu) < K(Z \restriction_n) + g(n) + d + c_0 < n - d - c_0 - c + g(n) + d + c_0 = n + g(n) - c = |\mu| - c.$$

This concludes the proof of the lemma. □

Let (Φ_e, g_e) an effective enumeration of all pairs of Turing functionals Φ and partial computable nondecreasing functions g which are a redundancy function for Φ. This means that for each e, n and each oracle X, if $\Phi_e^X(n)$ is defined then $g_e(n)$ is defined and the oracle-use in the computation $\Phi_e^X(n)$ is bounded above by $n+g_e(n)$. Let I contain the indices e such that g_e is a total function with $\sum_i 2^{-g(i)} = \infty$. For each e, let $S(e,c)$ be the set of strings ν with the property that for all μ of length $|\nu| + g(|\nu|)$ such that $\Phi_e^\mu = \nu$ we have $K(\mu) < |\mu| - c$. Then the sets $S(e,c)$ are Σ_2^0, uniformly in e, c. By Lemma 3.3, for each $e \in I$ and all c the set $S(e,c)$ is dense, i.e. every string has an extension in $S(e,c)$. Therefore every weakly 2-generic real has a prefix in $S(e,c)$, for each $e \in I$ and each c. Theorem 1.1 follows directly from this fact.

3.2 Essential part of the proof of Theorem 1.2

We denote Turing reducibility by $\leq_T$. In this section we show how to effectivize the argument of Section 3.1 in order to obtain a set $X \leq_T \emptyset'$ with the properties of Theorem 1.1. Then in Section 3.3 we use standard computability-theoretic apparatus in order to show that for any given set A which is generalized non-low$_2$ there exists such an X with $X \leq_T A$, thus completing the proof of Theorem 1.2.

In the proof of Lemma 3.6 we will need to restrict the enumeration of strings μ such that $\Phi^\mu = \nu$. The following definition introduces some notation for imposing such restrictions.

Definition 3.4 (Restricted enumeration of Φ)**.** *Let Φ be a Turing functional with redundancy g. Given any any string ν, let $Q_0(\nu)$ be the set of all strings μ of length $|\nu| + g(|\nu|)$ such that $\Phi^\mu = \nu$, and let $Q_0(\nu)[s]$ be a computable enumeration of this set. For each $d \in \mathbb{N}$ define*

$$Q(d,\nu) = \begin{cases} Q_0(\nu), & \text{if } |Q_0(\nu)| < 2^{d+g(|\nu|)}; \\ Q_0(\nu)[s(d,\nu)], & \text{otherwise.} \end{cases}$$

where $s(d,\nu)$ is the largest stage such that $|Q_0(\nu)|[s] < 2^{d+g(|\nu|)}$, in the case where $|Q_0(\nu)|[s]| \geq 2^{d+g(|\nu|)}$, and $s(d,\nu)$ is undefined otherwise.

Note that $Q(d,\nu)$ is uniformly c.e. in Φ, g, ν, d. Of course the definition of $Q(d,\nu)$ also depends upon Φ and the redundancy g, but these inputs will always be clear from context and so we suppress them for the sake of tidy notation. The following lemma will also be used in the proof of Lemma 3.6.

Lemma 3.5 (Turing functionals and prefix-free complexity)**.** *Let Φ be a Turing functional with redundancy g. There exists a prefix-free machine M such that*

$$\forall d, \nu\ \ \forall \mu \in Q(d,\nu)\ \ \Big(\Phi^\mu = \nu \quad \Rightarrow \quad K_M(\mu) \leq K(\nu) + g(|\nu|) + d\Big).$$

Moreover an index for M is uniformly computable from indices for Φ, g.

Proof. Such a machine M can be constructed as in the proof of Lemma 3.3. □

Lemma 3.6 (Effective density lemma)**.** *Let Φ be a Turing functional with computable nondecreasing redundancy g. There exists a computable function f such that for every c, ν_0:*

$$\sum_i 2^{-g(i)} > f(c,\nu_0) \Rightarrow \Big[\exists \nu \supset \nu_0\ \forall \mu \in 2^{|\nu|+g(|\nu|)} \Big(\Phi^\mu = \nu \Rightarrow K(\mu) < |\mu| - c\Big)\Big]. \tag{3.2.1}$$

Moreover an index of f can be obtained effectively from indices for Φ, g.

Proof. Given Φ, g as in the hypothesis, consider the functions h, h^* of Lemma 3.2. By Lemma 2.4, for each string ν_0 there exists a constant $d = d(\nu_0)$ (an example of which we can find effectively since all that is required is an upper bound for $h^*(\nu_0)$) and there exist a left-c.e. supermartingale N and a real Z such that:

$$\big(Z \supset \nu_0\big) \ \bigwedge\ \big(h(Z\upharpoonright_n) < 2^{d(\nu_0)+g(n)} \quad \text{for all } n\big) \ \bigwedge\ \Big(N(Z\upharpoonright_n) \geq \sum_{i=0}^{n} 2^{-g(i)} \quad \text{for all } n > |\nu_0|\Big).$$

Note that the second clause of the conjunction above follows since $h^*(Z\upharpoonright_n) < 2^d$ implies $h(Z\upharpoonright_n) < 2^{d+g(n)}$. Moreover, as observed in Section 2.2, an index for N (together with an upper bound for $N(\lambda)$) can be obtained effectively from ν_0 and indices for Φ and g. So by Lemma 2.1, there exists a computable function f_0 (whose index is computable from the indices of Φ, g) such that for all $t \in \mathbb{N}$ we have:

$$\sum_i 2^{-g(i)} > f_0(c, \nu_0) \Rightarrow \exists \nu \supset \nu_0 \left(h(\nu) < 2^{d(\nu_0)+g(|\nu|)} \ \wedge\ K(\nu) < |\nu| - c\right). \tag{3.2.2}$$

Now consider the machine M of Lemma 3.5, and let m be its index, which is a computable function of indices for Φ, g. We define $f(c, \nu_0) = f_0(c + m + 1 + d(\nu_0), \nu_0)$ for each c, ν_0, and show that f meets condition (3.2.1). Given our choice for the underlying universal prefix-free machine U, we have $K(\rho) < K_M(\rho)+m+1$ for all strings ρ. Fix c, ν_0 and assume that the left-hand-side of (3.2.1) holds. Then by (3.2.2) and the definition of f, there exists an extension ν of ν_0 such that $K(\nu) < |\nu| - c - d(\nu_0) - m - 1$ and $h(\nu) < 2^{d(\nu_0)+g(|\nu|)}$. By Definition 3.4 the latter inequality implies that

$$\{\mu \in 2^{|\nu|+g(|\nu|)} \mid \Phi^\mu = \nu\} = Q(d(\nu_0), \nu).$$

From Lemma 3.5 it follows that for all strings μ of length $|\nu| + g(|\nu|)$ with $\Phi^\mu = \nu$:

$$K_M(\mu) \leq K(\nu) + g(|\nu|) + d(\nu_0) \leq (|\nu| - c - m - 1 - d(\nu_0)) + g(|\nu|) + d(\nu_0).$$

This establishes that $K(\mu) \leq |\nu| + g(|\nu|) - c = |\mu| - c$. Finally observe that f is obtained effectively from indices for Φ and g, which concludes the proof of the lemma. □

From the above proof and Lemma 2.1 we can see that Lemma 3.6 also holds for partial computable functions g in the following sense. Let (Φ_e, g_e) an effective enumeration of all pairs of Turing functionals Φ and partial computable nondecreasing functions g.

> There exists a computable function f_* such that for each e, c, ν_0, k, if $g_e(i)\downarrow$ for all $i \leq k$ and $\sum_{i=0}^{k} 2^{-g_e(i)} > f_*(e, c, \nu_0)$ then there exists an extension ν of ν_0 of length k such that $K(\mu) < |\mu| - c$ for all strings μ of length $|\nu| + g_e(|\nu|)$ such that $\Phi_e^\mu = \nu$. (3.2.3)

Note that f of Lemma 3.6 had two arguments, while f_* has three arguments, as it deals with every potential redundancy function g_e. We are now ready to prove Theorem 1.2. Given Lemma 3.6, this is a standard argument in computability theory. We first describe the construction of a real $X \leq_T \emptyset'$ which meets the requirements of the theorem, which are:

$$\mathcal{R}_{e,c}: \ \text{If } g_e \text{ is total and } \sum_i 2^{-g_e(i)} = \infty \quad \text{then} \quad \forall Y \left(\Phi_e^Y = X \Rightarrow \exists n \ K(Y\upharpoonright_n) \leq n - c\right).$$

The proofs of the full claims of Theorems 1.2 and 1.3, regarding generalized non-low$_2$ and array non-computable sets, will be modifications of this simpler case. Let $\langle .,. \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a computable bijection. We describe a finite extension construction for the set X, which is computable from the halting problem. The main issue here is that the left-hand-side of the outer implication in $\mathcal{R}_{e,c}$ is not computable from the halting problem. This is the reason why we need Lemma 3.6 and (3.2.3), and not just Lemma 3.3.

We define a monotone sequence of strings (σ_i), beginning with the empty string σ_0 and eventually defining $X = \cup_i \sigma_i$. At stage $\langle e, c\rangle + 1$ we meet $\mathcal{R}_{e,c}$. Let f_* be the function from (3.2.3). Inductively assume that σ_i, $i \le \langle e, c\rangle$ have been defined. At stage $\langle e, c\rangle + 1$ we ask if there exists k such that

$$g_e(i) \downarrow \text{ for all } i \le k \qquad \text{and} \qquad \sum_{i=0}^{k} 2^{-g_e(i)} > f_*(e, c, \nu_0).$$

If not, we simply let $\sigma_{\langle e,c\rangle+1}$ be $\sigma_{\langle e,c\rangle} * 0$. Otherwise we search for a proper extension ν of $\sigma_{\langle e,c\rangle}$ of length at most k such that

$$\forall \mu \in 2^{|\nu|+g_e(|\nu|)} \ \left(\Phi_e^{\mu} = \nu \Rightarrow K(\mu) < |\mu| - c\right). \tag{3.2.4}$$

By (3.2.3) such a string ν exists. In this case we define $\sigma_{\langle e,c\rangle+1} = \nu$. This completes the inductive definition of (σ_i) and X. An inspection of the construction suffices to verify that only Σ_1^0 questions are asked, so $X \le_T \emptyset'$. Moreover for each e, c, condition $\mathcal{R}_{e,c}$ is met by all reals extending $\sigma_0 * \cdots * \sigma_{\langle e,c\rangle+1}$. So the real $X = \cup_i \sigma_i$ meets $\mathcal{R}_{e,c}$ for all e, c.

3.3 Proof of Theorem 1.2

Suppose that A is a generalized non-low$_2$ set. We modify the construction of the previous section so as to build $X \le_T A$. The requirements to be satisfied are $\mathcal{R}_{e,c}$ as before. Recall that since A is a generalized non-low$_2$, for every Δ_2^0 function $n \mapsto p(n)$ there exists a function $n \mapsto q(n)$ which is computable from A and is not dominated by p, i.e. such that there exist infinitely many n for which $q(n) > p(n)$. The rough idea is the same as always when modifying constructions with oracle the halting set, in order to work below A which is generalized non-low$_2$. One defines a function p which is computable in the halting set, and which gives an upper bound for the length of computable search required at each stage of the construction in order to proceed 'correctly'. Then one shows that, in fact, it suffices to use q which is not dominated by p in order to bound the search at each stage.

3.3.1 Dominating function and dynamics of strategies

We first define a function $p \le_T \emptyset'$ which is sufficiently fast growing so that it provides good approximations to the conditions involved in $\mathcal{R}_{e,c}$. Recall the definition of f_* from (3.2.3). We may assume that $f_*(e, c, \nu_0) > |\nu_0|$.

We let $p_0(e, c, \nu_0)$ be the least $k > |\nu_0|$ such that

$$g_e(i)\downarrow \text{ for all } i \le k \quad \text{and} \quad \sum_{i=0}^{k} 2^{-g_e(i)} > f_*(e, c, \nu_0) \tag{3.3.1}$$

if there exists such, and we define $p_0(e, c, \nu_0) = 0$ otherwise. Let $p_0(e, c, \nu_0) = k$. Then we define $p_1(e, c, \nu_0)$ to be the least $s > k$ such that:

(a) $g_e(i)[s]\downarrow$ for all $i \le k$ if $k > 0$.

(b) For all ν of length at most k and μ of length $|\nu| + g_e(|\nu|)$ such that $\Phi_e^\mu \downarrow= \nu$, we have $\Phi_e^\mu[s] \downarrow= \nu$.

(c) For all μ of length at most $k + g(k)$, $K(\mu)$ has settled by stage s, i.e. $K_s(\mu) = K(\mu)$.

Finally, we define $p(s)$ to be the least number greater than $p_1(e, c, \nu_0)$ for all e, c such that $\langle e, c\rangle \le s$ and ν_0 of length at most s.

Clearly $p \le_T \emptyset'$. Now fix a function $q \le_T A$ which is not dominated by p.

We are going to use q in order to construct X which meets all requirements $\mathcal{R}_{e,c}$. This will also be a finite extension construction, but it is important to ensure that the length of X that has been determined at stage e is of length e. This ensures that when we encounter some e such that $q(e) \ge p(e)$, it will not be too late to make the right decision in terms of satisfying some requirement of high priority that had remained unsatisfied in the previous stages. At the start of each stage $s + 1$ the initial segment $X \upharpoonright_s$ has been defined in the previous stages and we are called to define $X(s)$, therefore specifying $X \upharpoonright_{s+1}$.

At stage $s + 1$ we say that $\mathcal{R}_{e,c}$ *requires attention* if it has not already been declared satisfied, and:

(i) there exists a least $k > s$ such that: $k < q(s)$, for all $i \le k$ we have $g_e(i)[q(s)]\downarrow$, and $\sum_{i=0}^{k} 2^{-g_e(i)} > f_*(e, c, X \upharpoonright_s)$;

(ii) for this least k, there exists $\nu \supset X \upharpoonright_s$ of length k which satisfies the following condition: for all μ of length $k + g_e(k)$ such that $\Phi_e^\mu[q(s)] \downarrow= \nu$, $K_{q(s)}[\mu] < |\mu| - c$.

In this case we also say that $\mathcal{R}_{e,c}$ requires attention via ν for the lexicographically least ν satisfying the conditions of (ii) above.

3.3.2 Construction of the real

At stage 0: Define $X \upharpoonright_0 = \lambda$.

At stage $s + 1$: If there does not exist $\langle e, c\rangle \le s$ such that $\mathcal{R}_{e,c}$ requires attention, then define $X(s) = 0$. Otherwise, let $\langle e, c\rangle$ be that of highest priority, and let ν be such that $\mathcal{R}_{e,c}$ requires attention via ν. Define $X(s) = \nu(s)$. If $|\nu| = s + 1$ then declare $\mathcal{R}_{e,c}$ to be satisfied.

3.3.3 Verification of the construction

Suppose that no requirement of higher priority than $\mathcal{R}_{e,c}$ requires attention at any stage $> s_0$. We show that $\mathcal{R}_{e,c}$ is satisfied, and that there exists a stage after which this requirement does not require attention.

Let $s_1 > s_0$ be such that $q(s_1) > p(s_1)$. From the definition of $p(s_1)$ and the fact that $q(s_1) > p(s_1)$ it follows that if $\mathcal{R}_{e,c}$ does not require attention at stage $s_1 + 1$ then either g_e is not total, or else g_e is total and $\sum_i 2^{-g_e(i)}$ is finite. If $g_e(i) \uparrow$ for some i, then $\mathcal{R}_{e,c}$ cannot require attention subsequent to stage i. If $\sum_i 2^{-g_e(i)}$ is finite, then it follows directly from our assumption that $f_*(e, c, v_0) > |v_0|$ that $\mathcal{R}_{e,c}$ can only require attention at finitely many stages. So suppose, on the other hand, that $\mathcal{R}_{e,c}$ requires attention at stage $s_1 + 1$ via v. In this case the requirement will be declared satisfied by the end of stage $|v|$.

3.4 Proof of Theorem 1.3

Recall that given an array noncomputable set A and any function p which is weak truth-table computable in $\emptyset'$, there exists a function $q \leq_T A$ which is not dominated by p. So if the function p of Section 3.3.1 was computable from the halting problem with *computable bound on the oracle use*, then we would have proved Theorem 1.2 under the weaker assumption of array noncomputability. Unfortunately, this is not the case. However by the same argument we can obtain a nonuniform version of Theorem 1.2, under the weaker hypothesis of array noncomputability on the oracle A. Let A, g be as in the statement of Theorem 1.3, and let e be an index of g. We wish to construct $X \leq_T A$ which satisfies all requirements $\mathcal{R}_{e,c}$, $c \in \mathbb{N}$ of Section 3.2, for the fixed index e of g. The crucial point is that if we fix e such that g_e is total and follow the definition of p of Section 3.3.1 restricting to this fixed e, then the corresponding function p_e is computable from the halting problem with *computable bound on the oracle use*. Hence, if A is array noncomputable, we may choose an increasing function $q_e \leq A$ which is not dominated by p_e. Then the construction of Section 3.3.2, restricted to a fixed e such that $g_e = g$, gives a real $X \leq_T q_e$ which meets all requirements $\mathcal{R}_{e,c}, c \in \mathbb{N}$. Again, the proof of this fact is the argument of Section 3.3.3, restricted to our fixed e. Hence $X \leq_T q_e \leq_T A$ and X has the properties claimed in Theorem 1.3.

4 Conclusions

Kučera [Kuč85, Kuč89] and Gács [Gác86] showed that every real is computable from a random real. The best known general upper bound for the redundancy of such computations is $\sqrt{n} \cdot \log n$, and is due to Gács [Gác86] (Merkle and Mihailović [MM04] have provided a different proof of this fact). In the present paper we asked for the optimal redundancy that can be achieved in the Kučera-Gács theorem. We showed that no computable nondecreasing function g such that $\sum_i 2^{-g(i)} = \infty$ can be such an upper bound and demonstrated that a large class of oracles require larger redundancy when they are computed by random reals. This result improves the constant bound obtained by Downey and Hirschfeldt [DH10, Theorem 9.13.2]. Our result shows that, in general, the redundancy cannot be as slow growing as $\log n$, but a large exponential gap with the currently known bound of $\sqrt{n} \cdot \log n$ remained. Recently it was shown in [BLP16] that the strict lower bounds that we obtain in the present paper are optimal. In other words, any computable nondecreasing function g such that $\sum_i 2^{-g(i)} < \infty$ is a general upper bound on the redundancy in the computation of any real from some Martin-Löf random oracle. This provides a complete characterization of the redundancy bounds in the Kučera-Gács theorem.

References

[BDM15] George Barmpalias, Rod Downey, and Michael McInerney. Integer-valued betting strategies and Turing degrees. *Journal of Computer and System Sciences*, 81:1387–1412, 2015.

[BFLP16] George Barmpalias, Nan Fang, and Andy Lewis-Pye. Optimal asymptotic bounds on the oracle use in computations from chaitin's omega. *Journal of Computer and System Sciences*, 82:1283–1299, 2016.

[BHPS16] Laurent Bienvenu, Rupert Hölzl, Cris Porter, and Paul Shafer. Randomness and semimeasures. *Notre Dame Journal of Formal Logic*, 2016. In press.

[BL07] George Barmpalias and Andrew E. M. Lewis. Randomness and the linear degrees of computability. *Ann. Pure Appl. Logic*, 145(3):252–257, 2007.

[BLP16] George Barmpalias and Andy Lewis-Pye. Optimal redundancy in computations from random oracles. Preprint, ArXiv:1606.07910, 2016.

[BST10] Laurent Bienvenu, Frank Stephan, and Jason Teutsch. How powerful are integer-valued martingales? In *Programs, Proofs, Processes, 6th Conference on Computability in Europe, CiE 2010, Ponta Delgada, Azores, Portugal, June 30 - July 4, 2010. Proceedings*, pages 59–68, 2010.

[BST12] Laurent Bienvenu, Frank Stephan, and Jason Teutsch. How powerful are integer-valued martingales? *Theory of Computing Systems*, 51(3):330–351, 2012.

[BV11] George Barmpalias and Charlotte S. Vlek. Kolmogorov complexity of initial segments of sequences and arithmetical definability. *Theoretical Computer Science*, 412(41):5656–5667, 2011.

[CDFT12] Adam Chalcraft, Randall Dougherty, Chris Freiling, and Jason Teutsch. How to build a probability-free casino. *Information and Computation*, 211:160–164, 2012.

[DH10] Rod G. Downey and Denis Hirshfeldt. *Algorithmic Randomness and Complexity*. Springer, 2010.

[DJS96] Rod G. Downey, Carl G. Jockusch, Jr., and Michael Stob. Array nonrecursive sets and genericity. In *Computability, Enumerability, Unsolvability: Directions in Recursion Theory*, volume 224 of *London Mathematical Society Lecture Notes Series*, pages 93–104. Cambridge University Press, 1996.

[Dot06] David Doty. Every sequence is decompressible from a random one. In *Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006, Swansea, UK, June 30-July 5, 2006, Proceedings*, pages 153–162, 2006.

[Gác86] Péter Gács. Every sequence is reducible to a random one. *Inform. and Control*, 70(2-3):186–192, 1986.

[Her16] Ian Herbert. Lowness for integer-valued randomness. *Computability*, 5:103–109, 2016.

[Kuč85] Antonín Kučera. Measure, Π^0_1-classes and complete extensions of PA. In *Recursion theory week (Oberwolfach, 1984)*, volume 1141 of *Lecture Notes in Math.*, pages 245–259. Springer, Berlin, 1985.

[Kuč89] Antonin Kučera. On the use of diagonally nonrecursive functions. In *Logic Colloquium '87 (Granada, 1987)*, volume 129 of *Stud. Logic Found. Math.*, pages 219–239. North-Holland, Amsterdam, 1989.

[LV97] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Graduate Texts in Computer Science. Springer-Verlag, New York, second edition, 1997.

[ML66] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.

[MM04] Wolfgang Merkle and Nenad Mihailović. On the construction of effectively random sets. *J. Symb. Log.*, 69(3):862–878, 2004.

[PB15] Ron Peretz and Gilad Bavly. How to gamble against all odds. *Games and Economic behavior*, 94:157–168, 2015. Arxiv:1311.2109.

[Per15] Ron Peretz. Effective martingales with restricted wagers. *Information and Computation*, 245:152–164, 2015. Arxiv 1301.7465.

[Sch71a] Claus-Peter Schnorr. A unified approach to the definition of random sequences. *Math. Systems Theory*, 5:246–258, 1971.

[Sch71b] Claus-Peter Schnorr. *Zufälligkeit und Wahrscheinlichkeit. Eine algorithmische Begründung der Wahrscheinlichkeitstheorie*. Springer-Verlag, Berlin, 1971. Lecture Notes in Mathematics, Vol. 218.

[Tad09] Kohtaro Tadaki. Chaitin ω numbers and halting problems. In *Proceedings of the 5th Conference on Computability in Europe: Mathematical Theory and Computational Practice*, CiE '09, pages 447–456, Berlin, Heidelberg, 2009. Springer-Verlag.

[Teu14] Jason Teutsch. A savings paradox for integer-valued gambling strategies. *International Journal of Game Theory*, 43(1):145–151, 2014.

[ZL70] Alexander K. Zvonkin and Leonid A. Levin. The complexity of finite objects and the basing of the concepts of information and randomness on the theory of algorithms. *Uspehi Mat. Nauk*, 25(6(156)):85–127, 1970.