Recursive Computational Depth *

James I. Lathrop and Jack H. Lutz
Department of Computer Science
Iowa State University
Ames, Iowa 50011
U.S.A.

Abstract

In the 1980's, Bennett introduced computational depth as a formal measure of the amount of computational history that is evident in an object's structure. In particular, Bennett identified the classes of *weakly deep* and *strongly deep* sequences, and showed that the halting problem is strongly deep. Juedes, Lathrop, and Lutz subsequently extended this result by defining the class of *weakly useful* sequences, and proving that every weakly useful sequence is strongly deep.

The present paper investigates refinements of Bennett's notions of weak and strong depth, called *recursively weak depth* (introduced by Fenner, Lutz and Mayordomo) and *recursively strong depth* (introduced here). It is argued that these refinements naturally capture Bennett's idea that deep objects are those which "contain internal evidence of a nontrivial causal history." The fundamental properties of recursive computational depth are developed, and it is shown that the recursively weakly (respectively, strongly) deep sequences form a proper subclass of the class of weakly (respectively, strongly) deep sequences. The above-mentioned theorem of Juedes, Lathrop, and Lutz is then strengthened by proving that every weakly useful sequence is recursively strongly deep. It follows from these results that not every strongly deep sequence is weakly useful, thereby answering a question posed by Juedes.

*This research was supported in part by National Science Foundation Grant CCR-9157382, with matching funds from Rockwell, Microware Systems Corporation, and Amoco Foundation.

1 Introduction

Computational depth was introduced by Bennett [3, 4] as a formal measure of the amount of computational history that is evident in the structure of a computational, physical, or biological object. Roughly speaking, if x is an object (such as a computer program, a point in a phase space, or a DNA sequence) that can be encoded in binary in a natural way — in which case we identify x with its encoding — then the computational depth of x is the amount of time required for a computation to derive x from its shortest binary description. (Precise definitions appear in sections 2 and 3 of this paper.) Like Solomonoff [27], Bennett regards a description of x as a formal analog of a scientific explanation of x. By Occam's razor, then, the shortest description of x is the most plausible explanation of x, and the computational depth of x is the amount of time required for an effective process to generate x from its most plausible explanation. Bennett thus says that a deep object is "one whose most plausible origin, via an effective process, entails a lengthy computation," and, more succinctly, that a deep object is one that contains "internal evidence of a nontrivial causal history" [4].

In order to avoid undue sensitivity to the underlying computational model, Bennett's definition of depth refers not only to an object's shortest description, but to all descriptions of the object that have nearly minimal length. This is achieved by adding a significance parameter to the definition. Specifically, for $c \in \mathbb{N}$, the computational depth of an object x at significance level c is the time required for a computation to derive x from a binary description π that is itself compressible by no more than c bits. (That is, every description of π consists of at least $|\pi| - c$ bits.)

For (infinite, binary) sequences, Bennett [3, 4] introduced two interesting depth conditions, strong depth and weak depth. A sequence S is *strongly deep* if, for every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every constant $c \in \mathbb{N}$, for all but finitely many $n \in \mathbb{N}$, the n-bit prefix $S[0..n-1]$ of S has depth greater than $t(n)$ at significance level c. If we regard a description π from which $S[0..n-1]$ can be derived in at most $t(n)$ computation steps as a $t(n)$-*compression* of $S[0..n-1]$, then this says that, for all computable time bounds t and constants c, for all but finitely many n, every $t(n)$-compression of $S[0..n-1]$ is itself compressible by more than c bits. Thus a sequence is strongly deep if no computable time bound suffices to compress infinitely many of its prefixes to within a constant number of bits of the optimal compression.

To put the matter more fancifully, no matter how (computably) much time is spent looking for inner structure (i.e., basis for compression) in a strongly deep sequence, an unbounded quantity of such inner structure remains undiscovered. A strongly deep sequence is thus analogous to a great work of literature for which no number of readings suffices to exhaust its value.

It was shown by Bennett [4] (and also in [10]) that no sequence that is either decidable or random (i.e., algorithmically random in the sense of Martin-Löf [19]) can be strongly deep. However, strongly deep sequences do exist. For example, Bennett [4] noted that K, the diagonal halting problem, is strongly deep. This is because K, unlike a decidable or random sequence, can be used (as an oracle) to decide any decidable sequence within a computable (in fact, polynomial) time bound that does not depend on the sequence.

This relationship between depth and usefulness (as an oracle) was investigated more explicitly and generally by Juedes, Lathrop, and Lutz [10], who defined strong and weak usefulness conditions for sequences. A sequence S is *strongly useful* if there is a fixed computable time bound $t : \mathbb{N} \to \mathbb{N}$ such that the set $\mathrm{DTIME}^S(t)$, consisting of all sequences that can be decided in $t(n)$ time using the oracle S, contains every decidable sequence, i.e., $\mathrm{REC} \subseteq \mathrm{DTIME}^S(t)$, where REC is the set of all decidable sequences. A sequence S is *weakly useful* if there is a fixed computable time bound $t : \mathbb{N} \to \mathbb{N}$ such that the set $\mathrm{DTIME}^S(t)$ does not have measure 0 in REC, i.e., $\mathrm{DTIME}^S(t) \cap \mathrm{REC}$ is a nonnegligible subset of REC in the sense of the recursive case of the resource-bounded measure theory developed by Lutz [18]. That is, S is weakly useful if a nonnegligible set of decidable sequences can be decided within a computable time bound that may depend on S but does not depend on the sequence being decided. By the above remark, K is strongly useful. It is evident that every strongly useful sequence is weakly useful, and Fenner, Lutz, and Mayordomo [6] have shown that the converse does not hold, so the set of strongly useful sequences is properly contained in the set of weakly useful sequences.

Juedes, Lathrop, and Lutz [10] proved that every weakly useful sequence is strongly deep. This generalized Bennett's observation that K is strongly deep and gave formal support to Bennett's informal arguments relating depth and usefulness. Strong depth is a necessary condition for weak usefulness. Juedes [9] subsequently asked whether the converse is true, i.e., whether strong depth actually characterizes weak usefulness.

In this paper, we show that weakly useful sequences have a strictly stronger depth property than strong depth, thereby answering Juedes's question negatively. In fact, this stronger depth property, a constructive refinement of strong depth called *recursively strong depth*, is the main topic of this paper.

In the terminology used above to describe strong depth, a sequence S is *recursively strongly deep* (briefly, rec-*strongly* deep) if, for every computable time bound t and constant c, there exists a computable time bound l such that, for all but finitely many n, every $t(n)$-compression of $S[0..n-1]$ is itself $l(n)$-compressible by more than c bits. It is the existence of this computable time bound l that distinguishes rec-strong depth from strong depth. Returning to the more fanciful language used earlier, no matter how (computably) much time is spent looking for inner structure in a rec-strongly deep sequence, and no matter now much additional structure (any constant number of bits) one wishes to find, there is always a greater (computable) amount of time that suffices to find that much more structure. A rec-strongly deep sequence is thus analogous to a great work of literature with the property that, no matter how many times it has been read, there is a greater number of readings from which one can derive significantly more value.

In this paper, we establish the existence of sequences that are strongly deep but not rec-strongly deep. Such a sequence S must have the following two properties.

(i) There exist a *fixed* computable time bound $t_0 : \mathbb{N} \to \mathbb{N}$ and a *fixed* constant $c_0 \in \mathbb{N}$ such that, for *every* computable time bound $l : \mathbb{N} \to \mathbb{N}$, there are infinitely many prefixes $S[0..n-1]$ of S that have $t_0(n)$-compressions that are not $l(n)$-compressible by c_0 or more bits.

(ii) For *every* constant $c \in \mathbb{N}$ (no matter now much larger than c_0), for all but finitely many prefixes $S[0..n-1]$ of S, every $t_0(n)$-compression of $S[0..n-1]$ is itself compressible by more than c bits.

By (i), none of the additional compression (beyond c_0 bits) promised in (ii) can be realized within any computable time bound. Once again comparing a sequence to a work of literature and taking a number of readings as an analogy for a computable time bound, a sequence that is strongly deep but not rec-strongly deep is analogous to a work of literature for which no number of readings exhausts its value, but some number of readings does exhaust all the value that can be exhausted by any number of readings.

Using Bennett's terminology, a rec-strongly deep sequence S shows evidence of a nontrivial causal (computational) history in the constructive, incremental sense that every explanation of S that can be realized by an effective process of computable duration is significantly less plausible than some other explanation of S that can also be realized by an effective process of some greater computable duration. In contrast, a sequence that is strongly deep but not rec-strongly deep has an explanation that (i) can be realized by an effective process of computable duration, and (ii) is as plausible as any other explanation that can be realized by an effective process of computable duration. Although such a sequence does have a more plausible explanation, there is no constructive evidence of this fact.

None of the above should be taken to imply that rec-strong depth is a better (or worse) notion than strong depth. Both notions merit further investigation. In the case of rec-strong depth, there are several reasons for this. First, as noted above, rec-strongly deep sequences show evidence of a "nontrivial causal history" in a natural, constructive, incremental sense. Second, as we show in this paper, rec-strong depth enjoys the same useful slow-growth property (and consequent upward closure under truth-table reductions) that Bennett [4] proved for strong depth. Third, as we show in this paper, rec-strong depth can be used to separate weak usefulness from strong depth, thereby answering Juedes's question. Fourth, as developed below, rec-strong depth is based on a *recursive* depth function (with an additional latency parameter), and therefore provide a useful model for the design and analysis of *implementable* depth measures such as the compression depth introduced by Lathrop [12]. Fifth, and perhaps most compelling, we show that the relationships among rec-strong depth, the notion of rec-weak depth introduced by Fenner, Lutz and Mayordomo [6], and the notion of rec-randomness that has been investigated by Schnorr [24, 25], van Lambalgen [28], Lutz [18], Wang [29], and others correspond closely to the relationships among strong depth, weak depth and algorithmic randomness.

This paper is largely self-contained. It can be read independently of [4, 10], but we assume that [10] is at hand for reference. In section 2 we introduce basic terminology and notation and summarize those elements of recursive measure, randomness, Kolmogorov complexity, and computational depth that are used in this paper. Section 3, the main section of this paper, presents rec-strong depth, rec-weak depth, and our results on these notions. Section 3 is divided into a preamble and four (sub-)sections. In the preamble, we develop the above-mentioned recursive depth function, $\text{depth}_c^l(w)$. In section 3.1 we use this function to define rec-strong depth; we review the notion of rec-weak depth introduced by Fenner, Lutz, and Mayordomo [6]; and we introduce the most basic properties of these notions. In section 3.2 we prove the deterministic slow growth law for recursive computational depth and establish the basic inclusion relations among the weak, strong, rec-weak, and rec-strong depth classes, namely,

$$\begin{array}{ccccc} & & \text{rec-wkDEEP} & & \\ & \subseteq & & \subseteq & \\ \text{rec-strDEEP} & & & & \text{wkDEEP.} \\ & \supseteq & & \subseteq & \\ & & \text{strDEEP} & & \end{array}$$

In section 3.3 we prove that all these inclusions are proper by proving that the classes rec-wkDEEP and strDEEP are incomparable. Both directions of the incomparability proof are nontrivial. One direction yields the stronger fact that rec-random sequences can be strongly deep, while the other direction uses the recursive version of the first Borel-Cantelli lemma [18] in a Baire category argument. In section 3.4 we prove that every weakly useful sequence is rec-strongly deep, thereby answering Juedes's question. In section 4 we briefly indicate directions for future research.

2 Preliminaries

2.1 Notation and Terminology

We use the sets $\mathbb{Z}, \mathbb{Z}^+, \mathbb{N}, \mathbb{Q}$, and $\mathbb{R}$, consisting of all integers, positive integers, non-negative integers, rational numbers, and real numbers, respectively. Given a property $\varphi(n)$, where the variable n ranges over $\mathbb{N}$, we use the abbreviations

$$(\exists^\infty n)\varphi(n) \equiv \text{ there exist infinitely many } n \in \mathbb{N} \text{ such that } \varphi(n),$$
$$(\forall^\infty n)\varphi(n) \equiv \text{ for all but finitely many } n \in \mathbb{N},\ \varphi(n).$$

The *Boolean value* of a condition ψ is $[\![\psi]\!] = $ **if** ψ **then** 1 **else** 0. All logarithms are base-2. The cardinality of a finite set X is denoted by $|X|$.

We write $\{0,1\}^*$ for the set of all (finite, binary) *strings*. We write $|w|$ for the length of a string w and λ for the empty string. The *self-delimiting version* of a string $w \in \{0,1\}^*$ is the string $\mathbf{sd}(w) = 0^{|w|}1w$. For $k \in \mathbb{N}$ and $w_0, \ldots, w_{k-1} \in \{0,1\}^*$, the *self-delimiting encoding* of the sequence $(w_0, \cdots, k_{k-1})$ is

$$\langle w_0, \cdots, w_{k-1}\rangle = \mathbf{sd}(\mathbf{sd}(w_0)\cdots\mathbf{sd}(w_{k-1})).$$

The *standard enumeration* of $\{0,1\}^*$ is the sequence $s_0 = \lambda$, $s_1 = 0$, $s_2 = 1$, $s_3 = 00$, $\ldots$, ordered first by length and then lexicographically. For $u, v \in \{0,1\}^*$, u is a *prefix* of v, and we write $u \sqsubseteq v$, if there is a string $w \in \{0,1\}^*$ such that $v = uw$. For $w \in \{0,1\}^*$ and $0 \le n < |w|$, we write $w[n]$ for the n^{th} bit of w. (The leftmost bit of w is the 0^{th} bit.) For $w \in \{0,1\}^*$ and $0 \le n \le |w|$, we write $w[0..n-1]$ for the n-bit prefix of w.

We work in the *Cantor space* $\mathbf{C}$, consisting of all (infinite, binary) *sequences*. A string $w \in \{0,1\}^*$ is a *prefix* of a sequence $S \in \mathbf{C}$, and we write $w \sqsubseteq S$ if there is a sequence $A \in \mathbf{C}$ such that $S = wA$. For $S \in \mathbf{C}$ and $n \in \mathbb{N}$, we write $S[n]$ for the n^{th} bit of S and $S[0..n-1]$ for the n-bit prefix of S. The *complement* of a set $X \subseteq \mathbf{C}$ is the set $X^{\text{c}} = \mathbf{C} - X$.

We write REC for the set of all decidable sequences in $\mathbf{C}$ and rec for the set of all computable (total) functions from $\{0,1\}^*$ to $\{0,1\}^*$. Identifying strings s_n with their indices n in the standard enumeration of $\{0,1\}^*$, we also write rec for the set of all computable functions from $\mathbb{N}$ to $\mathbb{N}$.

Given a time bound $s : \mathbb{N} \to \mathbb{N}$, we say that an oracle Turing machine M is *s-time-bounded* if, given any input $n \in \mathbb{N}$ and oracle $A \in \mathbf{C}$, M decides a bit $M^A(n) \in \{0,1\}$ in at most $s(l)$ steps, where $l = |s_n| = \lfloor \log(n+1) \rfloor$. In this case, if $B \in \mathbf{C}$ satisfies

$B[n] = M^A(n)$ for all $n \in \mathbb{N}$, then we say that B is *Turing reducible to A in time s via M*, and we write $B \leq_{\mathrm{T}}^{\mathrm{DTIME}(s)} A$ via M. We say that B is *Turing reducible to A in time s*, and we write $B \leq_{\mathrm{T}}^{\mathrm{DTIME}(s)} A$, if there exists a Turing machine M such that $B \leq_{\mathrm{T}}^{\mathrm{DTIME}(s)} A$ via M. For $A \in \mathbf{C}$ and $s : \mathbb{N} \to \mathbb{N}$, we write

$$\mathrm{DTIME}^A(s) = \left\{ B \in \mathbf{C} \,\middle|\, B \leq_{\mathrm{T}}^{\mathrm{DTIME}(s)} A \right\}.$$

(Note that the time bound is sharp; there is no "big-O.")

As in [22], we define a *truth-table condition* (briefly, a *tt-condition*) to be an ordered pair $\tau = ((n_1, \ldots n_k), g)$, where $k, n_1, \ldots n_k \in \mathbb{N}$ and $g : \{0,1\}^k \to \{0,1\}$. We write TTC for the set of all tt-conditions. The tt-*value* of a sequence $S \in \mathbf{C}$ under a tt-condition $\tau = ((n_1, \ldots, n_k), g)$ is the bit

$$\tau^S = g(S[n_1] \cdots S[n_k]).$$

A *truth-table reduction* (briefly, a *tt-reduction*) is a computable function $F : \mathbb{N} \to$ TTC. A tt-reduction F naturally *induces* a function $\widehat{F} : \mathbf{C} \to \mathbf{C}$ defined by

$$\widehat{F}(S)[n] = F(n)^S$$

for all $n \in \mathbb{N}$. In general, we identify a tt-reduction F with the induced function $\widehat{F}$, writing F for either function. For $A, B \in \mathbf{C}$, A is *truth-table reducible* (briefly, *tt-reducible*) to B, and we write $A \leq_{\mathrm{tt}} B$, if there is a tt-reduction F such that $A = F(B)$.

It is easy well known that tt-reductions are equivalent to time-bounded Turing reductions in the sense that for all $A, B \in \mathbf{C}$, $A \leq_{\mathrm{tt}} B$ if and only if there exists a computable time-bound $t : \mathbb{N} \to \mathbb{N}$ such that $A \leq_{\mathrm{T}}^{\mathrm{DTIME}(t)} B$.

Definition. A *uniform reducibility* is a computable function $F : \mathbb{N} \times \mathbb{N} \to$ TTC.

If F is a uniform reducibility, then we use the notation $F_k(n) = F(k, n)$, thereby regarding F as a computable sequence $F_0, F_1, F_2, \ldots$ of tt-reductions.

Definition. If F is a uniform reducibility and $A, B \in \mathbf{C}$, then A is *F-reducible to B*, and we write $A \leq_F B$, if there exists $k \in \mathbb{N}$ such that $A = F_k(B)$.

The following well-known facts are easy to verify.

(i) For every computable function $t : \mathbb{N} \to \mathbb{N}$, there is a uniform reducibility F such that, for all $A, B \in \mathbf{C}$,

$$A \leq_F B \Longleftrightarrow A \leq_{\mathrm{T}}^{\mathrm{DTIME}(t)} B.$$

(ii) For every uniform reducibility F, there is a computable function $t : \mathbb{N} \to \mathbb{N}$ such that, for all $A, B \in \mathbf{C}$,

$$A \leq_F B \Longrightarrow A \leq_{\mathrm{T}}^{\mathrm{DTIME}(t)} B.$$

Let D be a discrete domain such as $\mathbb{N}$, $\{0,1\}^*$, or $\mathbb{N}\times\{0,1\}^*$. A function $f : D \to \mathbb{Q}$ is *exactly computable* if there exist computable functions $f_1, f_2 : D \to \mathbb{Z}$ such that, for all $x \in D$, $f(x) = f_1(x)/f_2(x)$. A function $f : D \to \mathbb{R}$ is *computable* if there is an exactly computable function $\widehat{f} : \mathbb{N} \times D \to \mathbb{Q}$ such that, for all $r \in \mathbb{N}$ and $x \in D$, $|\widehat{f}(r,x) - f(x)| \leq 2^{-r}$. A function $f : D \to \mathbb{R}$ is *lower semicomputable* if there is an exactly computable function $\widehat{f} : \mathbb{N} \times D \to \mathbb{Q}$ such that (i) for all $r \in \mathbb{N}$ and $x \in D$, $\widehat{f}(r,x) \leq \widehat{f}(r+1,x)$; and (ii) for all $x \in D$, $\lim_{r\to\infty} \widehat{f}(r,x) = f(x)$.

A series $\sum_{n=0}^{\infty} \alpha_n$ of nonnegative reals α_n is rec-*convergent* if there is a computable function $m : \mathbb{N} \to \mathbb{N}$, called a *modulus of convergence*, such that, for all $r \in \mathbb{N}$, $\sum_{n=m(r)}^{\infty} \alpha_n \leq 2^{-r}$. More generally, if $\sum_{n=0}^{\infty} \alpha_{k,n}$ is a series of nonnegative reals for each $k \in \mathbb{N}$, then the series $\sum_{n=0}^{\infty} \alpha_{k,n}$ $(k = 0, 1, \ldots)$ are *uniformly* rec-*convergent* if there is a computable function $m : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that, for all $k, r \in \mathbb{N}$, $\sum_{n=m(k,r)}^{\infty} \alpha_{k,n} \leq 2^{-r}$.

2.2 Randomness and Kolmogorov Complexity

We work with the uniform probability measure μ on the Cantor space $\mathbf{C}$. For each $w \in \{0,1\}^*$, the *cylinder*

$$\mathbf{C}_w = \left\{ A \in \mathbf{C} \,\middle|\, w \sqsubseteq A \right\}$$

is assigned the probability

$$\mu(\mathbf{C}_w) = \Pr(\mathbf{C}_w) = 2^{-|w|}.$$

For each event (measurable set) $\mathcal{E} \subseteq \mathbf{C}$, the probability $\mu(\mathcal{E}) = \Pr(\mathcal{E})$ is then defined in the standard way [23]. We write $\Pr[\varphi(A)]$ or $\Pr_A[\varphi(A)]$ for $\Pr(\{A|\varphi(A)\})$.

A *martingale* is a function $d : \{0,1\}^* \to [0,\infty)$ such that, for all $w \in \{0,1\}^*$,

$$d(w) \geq \frac{d(w0) + d(w1)}{2}.$$

The following inequality of Kolmogorov is easily verified.

Lemma 2.1. If d is a martingale and $0 \leq \alpha \in \mathbb{R}$, then

$$\Pr_A[(\exists w \sqsubseteq A) d(w) \geq \alpha \cdot d(\lambda)] \leq \frac{1}{\alpha}.$$

In particular, for all $w \in \{0,1\}^*$, $d(w) \leq 2^{|w|} d(\lambda)$.

A martingale d *succeeds* on a sequence $A \in \mathbf{C}$ if

$$\limsup_{n\to\infty} d(A[0..n-1]) = \infty.$$

The *success set* of a martingale d is

$$S^\infty[d] = \left\{ A \in \mathbf{C} \;\middle|\; d \text{ succeeds on } A \right\}.$$

It follows readily from Lemma 2.1 that a set $X \subseteq \mathbf{C}$ is a probability 0 event (i.e., $\Pr(X) = 0$) if and only if there is a martingale d such that $X \subseteq S^\infty[d]$. As in [18], we effectivize this characterization to obtain a notion of measure in REC.

A rec-*martingale* (*recursive martingale*) is a martingale that is computable in the sense defined in section 2.1.

Definition (Lutz [18]). Let $X \subseteq \mathbf{C}$.

1. X has rec-*measure 0*, and we write $\mu_{\text{rec}}(X) = 0$, if there is a rec-martingale d such that $X \subseteq S^\infty[d]$.
2. X has rec-*measure 1*, and we write $\mu_{\text{rec}}(X) = 1$, if $\mu_{\text{rec}}(X^{\text{c}}) = 0$.
3. X has *measure 0 in* REC, and we write $\mu(X|\text{REC}) = 0$, if $\mu_{\text{rec}}(X \cap \text{REC}) = 0$.
4. X has *measure 1 in* REC, and we write $\mu(X|\text{REC}) = 1$, if $\mu(X^{\text{c}}|\text{REC}) = 0$.

Results proven in [18] justify the intuition that $\mu(X|\text{REC}) = 0$ if and only if $X \cap \text{REC}$ is a *negligibly small subset* of REC. Accordingly, if $\mu(X|\text{REC}) = 1$, we say that X contains *almost every* sequence in REC.

The *unitary success set* of a martingale d is

$$S^1[d] = \bigcup_{d(w)\geq 1} \mathbf{C}_w.$$

In section 3.3 we use the following uniform, recursive version of the first Borel-Cantelli lemma.

Theorem 2.2 (Lutz [18]). Assume that

$$d : \mathbb{N} \times \mathbb{N} \times \{0,1\}^* \to [0,\infty)$$

is a computable function with the following two properties.

(i) For each $k, n \in \mathbb{N}$, the function $d_{k,n}$ defined by $d_{k,n}(w) = d(k, n, w)$ is a martingale.

(ii) The series $\sum_{n=0}^{\infty} d_{k,n}(\lambda)$ $(k = 0, 1, \ldots)$ are uniformly rec-convergent.

Then

$$\mu_{\text{rec}}\left(\bigcup_{k=0}^{\infty}\bigcap_{m=0}^{\infty}\bigcup_{n=m}^{\infty} S^1[d_{k,n}]\right) = 0.$$

Recursive randomness has been investigated by Schnorr [24, 25], van Lambalgen [28], Lutz [18], Wang [29], and others. A sequence $S \in \mathbf{C}$ is rec-*random* (*recursively random*), and we write $S \in \text{RAND(rec)}$, if there is no rec-martingale that succeeds on S. The following easy consequence of Theorem 2.2 is also used in section 3.3.

Corollary 2.3 (Lutz [18]). Assume that $S \in \text{RAND(rec)}$ and let

$$d : \mathbb{N} \times \{0, 1\}^* \to [0, \infty)$$

be a computable function with the following two properties.

(i) For each $n \in \mathbb{N}$, the function d_n defined by $d_n(w) = d(n, w)$ is a martingale.

(ii) The series $\sum_{n=0}^{\infty} d_n(\lambda)$ is rec-convergent.

Then there are only finitely many $n \in \mathbb{N}$ such that $S \in S^1[d_n]$.

An *exact* rec-*martingale* is a martingale d with rational values (i.e., $d : \{0,1\}^* \to \mathbb{Q} \cap [0, \infty)$) that is exactly computable. The following lemma gives a convenient sufficient condition for rec-randomness. It follows immediately from the definition of rec-randomness, the recursive equivalence of martingale success and strong martingale success [28], and the Exact Computation Lemma [11, 20].

Lemma 2.4. Let $S \in \mathbf{C}$. If for every exact rec-martingale d satisfying $d(\lambda) = 1$ there exist $c_d \in \mathbb{N}$ and infinitely many prefixes $w \sqsubseteq S$ such that $d(w) \leq c_d$, then S is rec-random.

Algorithmic randomness, introduced by Martin-Löf [19], is a stronger condition than rec-randomness that can be defined in several equivalent ways. The definition in terms of martingales, introduced by Schnorr [24], states that a sequence $S \in \mathbf{C}$ is (*algorithmically*) *random* if no lower semicomputable martingale succeeds on S. We write RAND for the set of all random sequences. It is well-known [19] that $\Pr(\mathrm{RAND}) = 1$, i.e., almost every sequence is random.

We refer the reader to section 4 of [10] for a concise presentation of our terminology and notation on Kolmogorov complexity, including (self-delimiting) Turing machines, the efficient universal Turing machine U, the program sets $\mathrm{PROG}^t(x)$ and PROG^t, the Kolmogorov complexity $K(x)$, the time-bounded Kolmogorov complexity $K^t(x)$, the algorithmic probability $\mathbf{m}(x)$, and the time-bounded algorithmic probability $\mathbf{m}^t(x)$. These notions are also developed in the text by Li and Vitanyi [15]. We write $K(n)$ for $K(s_n)$, where s_n is the n^{th} string in the standard enumeration of $\{0,1\}^*$.

Lemma 2.5 (Chaitin [5]). There is a constant $c \in \mathbb{N}$ such that, for all $n, k \in \mathbb{N}$,

$$\left|\left\{x \in \{0,1\}^n \middle| K(x) \le n + K(n) - k\right\}\right| < 2^{n+c-k}.$$

We also use the following result on the noncomputability of $K(n)$.

Theorem 2.6 (Kolmogorov, reported in [30]). If $g : \mathbb{N} \to \mathbb{N}$ is partial recursive and unbounded, then there exist infinitely many $n \in \mathbb{N}$ such that $K(n) < g(n)$.

Randomness is characterized in terms of Kolmogorov complexity as follows.

Theorem 2.7 (Levin [13, 14], Schnorr [26]). A sequence $S \in \mathbf{C}$ is random if and only if there is a constant $c \in \mathbb{N}$ such that, for all $n \in \mathbb{N}$, $K(S[0..n-1]) \ge n - c$.

As in [10], for $t, g : \mathbb{N} \to \mathbb{N}$, we use the notation

$$\mathbf{K}^t_{\text{i.o.}}[< g(n)] = \left\{S \in \mathbf{C} \,\middle|\, (\exists^\infty n) K^t(S[0..n-1]) < g(n)\right\}$$

and the following result on measure in REC.

Theorem 2.8 (Lutz [18]). For every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every real number $\alpha < 1$,

$$\mu\left(\mathbf{K}^t_{\text{i.o.}}[< \alpha n] \middle| \mathrm{REC}\right) = 0.$$

2.3 Computational Depth

Following Bennett [4], we define the *computational depth* of a string $w \in \{0,1\}^*$ at a significance level $c \in \mathbb{N}$ to be

$$\text{depth}_c(w) = \min\left\{ t \in \mathbb{N} \,\middle|\, \left(\exists \pi \in \text{PROG}^t(w)\right) |\pi| < K(\pi) + c \right\}.$$

That is, $\text{depth}_c(w)$ is the minimum amount of time required to obtain w from a program π that cannot itself be obtained from a program that is c or more bits shorter than π. It is easy to see that $\text{depth}_c(w)$ is not computable from c and w. (Otherwise $K(w)$ would be computable, contradicting Theorem 2.6.) Also, for each $w \in \{0,1\}^*$, the value $\text{depth}_c(w)$ is nonincreasing in c.

Definition ([10]). For $t, g : \mathbb{N} \to \mathbb{N}$ and $n \in \mathbb{N}$, define the sets

$$\text{D}^t_g(n) = \left\{ S \in \mathbf{C} \,\middle|\, \text{depth}_{g(n)}\left(S[0..n-1]\right) > t(n) \right\}$$

and

$$\text{D}^t_g = \left\{ S \in \mathbf{C} \,\middle|\, (\forall^\infty n) S \in \text{D}^t_g(n) \right\}.$$

Note that

$$\text{D}^t_g(n) = \left\{ S \in \mathbf{C} \,\middle|\, \left(\forall \pi \in \text{PROG}^t(S[0..n-1])\right) K(\pi) \leq |\pi| - g(n) \right\}.$$

Definition (Bennett [4]). A sequence $S \in \mathbf{C}$ is *strongly deep*, and we write $S \in$ strDEEP, if for every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every constant $c \in \mathbb{N}$, $S \in \text{D}^t_c$. That is,

$$\text{strDEEP} = \bigcap_{\substack{c \in \mathbb{N} \\ t \in \text{rec}}} \text{D}^{\text{t}}_{\text{c}}.$$

The following theorem due to Bennett shows that random sequences are very shallow. A proof also appears in [10].

Theorem 2.9. RAND $\cap$ strDEEP $= \emptyset$. In fact, there exist a computable function $t(n) = O(n \log n)$ and a constant $c \in \mathbb{N}$ such that $\text{RAND} \cap \text{D}^{\text{t}}_{\text{c}} = \emptyset$.

Bennett [4] gave useful characterizations of strong depth in terms of the time-bounded Kolmogorov complexities and algorithmic probabilities of prefixes. As in [10], we state these characterizations in terms of the following classes, which turn out to be "minor variants" of the classes $\text{D}^t_g(n)$ and D^t_g.

Definition. For $t, g : \mathbb{N} \to \mathbb{N}$ and $n \in \mathbb{N}$, define the sets

$$\widehat{\mathrm{D}}_g^t(n) = \left\{ S \in \mathbf{C} \,\middle|\, K(S[0..n-1]) \leq K^t(S[0..n-1]) - g(n) \right\},$$
$$\widehat{\mathrm{D}}_g^t = \left\{ S \in \mathbf{C} \,\middle|\, (\forall^\infty n) S \in \widehat{\mathrm{D}}_g^t(n) \right\},$$
$$\widetilde{\mathrm{D}}_g^t(n) = \left\{ S \in \mathbf{C} \,\middle|\, \mathbf{m}(S[0..n-1]) \geq 2^{g(n)} \mathbf{m}^t(S[0..n-1]) \right\},$$
$$\widetilde{\mathrm{D}}_g^t = \left\{ S \in \mathbf{C} \,\middle|\, (\forall^\infty n) S \in \widetilde{\mathrm{D}}_g^t(n) \right\}.$$

Bennett's alternate characterizations of strong depth are as follows. A proof also appears in [10].

Theorem 2.10 (Bennett [4]). For $S \in \mathbf{C}$, the following three conditions are equivalent.

(1) S is strongly deep.

(2) For every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every constant $c \in \mathbb{N}$, $S \in \widehat{\mathrm{D}}_c^t$.

(3) For every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every constant $c \in \mathbb{N}$, $S \in \widetilde{\mathrm{D}}_c^t$.

Bennett defined weak depth as follows.

Definition. A sequence $S \in \mathbf{C}$ is *weakly deep*, and we write $S \in$ wkDEEP, if there is no sequence $R \in$ RAND such that $S \leq_{\mathrm{tt}} R$.

Bennett [4] proved that strDEEP $\subsetneqq$ wkDEEP. Juedes, Lathrop, and Lutz [10] subsequently proved the stronger fact that, in the sense of Baire category (defined in section 3.3 below), almost every sequence in $\mathbf{C}$ is weakly deep, but not strongly deep.

The reader is referred to [4], [10], or [15] for further discussion of computational depth.

3 Recursive Computational Depth

As noted in section 2.3, the value $\mathrm{depth}_c(w)$ – the computational depth of a string w at significance level c – is not computable from w and c. The following definition remedies this at the expense of introducing an additional variable.

Definition. For $w \in \{0,1\}^*$ and $c, l \in \mathbb{N}$, the *recursive computational depth of* w *at significance level* c *with latency* l is

$$\mathrm{depth}_c^l(w) = \min\left\{ t \in \mathbb{N} \,\middle|\, \left(\exists \pi \in \mathrm{PROG}^t(w)\right) |\pi| < K^l(\pi) + c \right\}.$$

That is, $\mathrm{depth}_c^l(w)$ is the minimum amount of time required to obtain w from a program π that cannot itself be obtained in time l from a program that is c or more bits shorter than π. It is clear that $\mathrm{depth}_c^l(w)$ is computable from w, c, and l; this is why it is called the *recursive* computational depth. Two other properties of $\mathrm{depth}_c^l(w)$ are immediately evident. For each $w \in \{0,1\}^*$ and $c \in \mathbb{N}$, $\mathrm{depth}_c^l(w)$ is nondecreasing in l, and $\lim_{l\to\infty} \mathrm{depth}_c^l(w) = \mathrm{depth}_c(w)$. For each $w \in \{0,1\}^*$ and $l \in \mathbb{N}$, the value $\mathrm{depth}_c^l(w)$ is, like $\mathrm{depth}_c(w)$, nonincreasing in c.

In this section, we use the quantity $\mathrm{depth}_c^l(w)$ to define recursively strong depth; we review the notion of recursively weak depth introduced by Fenner, Lutz, and Mayordomo [6]; and we investigate the relationships of these notions to each other, to the strong and weak depth notions of Bennett [4] (defined in section 2.3), and to the notion of weak usefulness introduced by Juedes, Lathrop, and Lutz [10].

3.1 Recursive Depth Classes

We begin by defining the recursive analogs of the depth classes $\mathrm{D}_g^t(n)$ and D_g^t discussed in section 2.3.

Definition. For $t, g, l : \mathbb{N} \to \mathbb{N}$ and $n \in \mathbb{N}$, define the sets

$$\mathrm{D}_g^{t,\,l}(n) = \left\{ S \in \mathbf{C} \,\middle|\, \mathrm{depth}_{g(n)}^{l(n)}(S[0..n-1]) > t(n) \right\}$$

and

$$\mathrm{D}_g^{t,\,l} = \bigcup_{m=0}^{\infty} \bigcap_{n=m}^{\infty} \mathrm{D}_g^{t,\,l}(n) = \left\{ S \in \mathbf{C} \,\middle|\, (\forall^\infty n)\, S \in \mathrm{D}_g^{t,\,l}(n) \right\}.$$

Note that

$$\mathrm{D}_g^{t,\,l}(n) = \left\{ S \in \mathbf{C} \,\middle|\, \left(\forall \pi \in \mathrm{PROG}^t(S[0..n-1])\right) K^{l(n)}(\pi) \le |\pi| - g(n) \right\}.$$

(It is crucial here that the left-hand side of the inequality is $K^{l(n)}(\pi)$, not $K^l(\pi)$, i.e., that the time bound is $l(n)$, not $l(|\pi|)$.)

Definition. Let $t, g : \mathbb{N} \to \mathbb{N}$. A sequence $S \in \mathbf{C}$ is *recursively t-deep at significance level g*, and we write $S \in \mathrm{D}_g^{t,\,\mathrm{rec}}$, if there is a computable function $l : \mathbb{N} \to \mathbb{N}$ such that $S \in \mathrm{D}_g^{t,\,l}$. That is,

$$\mathrm{D}_g^{t,\,\mathrm{rec}} = \bigcup_{l \in \mathrm{rec}} \mathrm{D}_g^{t,\,l}.$$

It is clear that, for all $t, g, l : \mathbb{N} \to \mathbb{N}$ with l computable, $\mathrm{D}_g^{t,\,l} \subseteq \mathrm{D}_g^{t,\,\mathrm{rec}} \subseteq \mathrm{D}_g^t$. To define recursively strong depth, we substitute $\mathrm{D}_g^{t,\,\mathrm{rec}}$ for D_g^t in the definition of strong depth.

Definition. A sequence $S \in \mathbf{C}$ is *recursively strongly deep* (or, briefly, rec-*strongly deep*), and we write $S \in$ rec-strDEEP, if for every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every constant $c \in \mathbb{N}$, $S \in \mathrm{D}_c^{t,\,\mathrm{rec}}$. That is,

$$\text{rec-strDEEP} = \bigcap_{\substack{c \in \mathbb{N} \\ t \in \mathrm{rec}}} \mathrm{D}_c^{t,\,\mathrm{rec}}.$$

We first note that every rec-strongly deep sequence is strongly deep.

Observation 3.1. rec-strDEEP $\subseteq$ strDEEP.

Proof. This follows immediately from the fact that each $\mathrm{D}_c^{t,\,\mathrm{rec}} \subseteq \mathrm{D}_c^t$. □

Since REC $\cap$ strDEEP $= \emptyset$ [4] (see also [10]), it follows immediately from Observation 3.1 that no recursive sequence can be rec-strongly deep.

Recall that a sequence S is strongly deep if, for every computable time bound t and constant c, all but finitely many prefixes of S can be described at least c bits more succinctly without a time bound than with the time bound t. In contrast, a sequence S is rec-strongly deep if, for every computable time bound t and constant c, there exists a *computable* time bound l such that all but finitely many prefixes of S can be described at least c bits more succinctly *with the time bound l* than with the time bound t. Very informally, a sequence is strongly deep if it has more regularity than can be explained by a causal (computational) history of any computable duration. For a sequence to be rec-strongly deep, it must also be the case that, for every computable duration t there is a larger computable duration l such that more of the sequence's regularity can be explained by a causal history of duration l than can be explained by a causal history of duration t.

Our next objective is to prove a recursive analog of Theorem 2.9, stating that rec-strongly deep sequences cannot be rec-random.

Lemma 3.2. Let $t, g, l : \mathbb{N} \to \mathbb{N}$ be computable. If $\Pr(\mathrm{D}_g^{t,\,l}) = 0$, then $\mu_{\mathrm{rec}}(\mathrm{D}_g^{t,\,l}) = 0$.

Proof. Assume the hypothesis. Then $\Pr\left(\bigcup_{m=0}^{\infty}\bigcap_{n=m}^{\infty} \mathrm{D}_g^{t,\,l}(n)\right) = 0$, so for each $m \in \mathbb{N}$, $\Pr\left(\bigcap_{n=m}^{\infty} \mathrm{D}_g^{t,\,l}(n)\right) = 0$. Thus, for each $m, k \in \mathbb{N}$, there exists $r \in \mathbb{N}$ such that $\Pr\left(\bigcap_{n=m}^{r} \mathrm{D}_g^{t,\,l}(n)\right) \leq 2^{-k}$. Since $\Pr\left(\bigcap_{n=m}^{r} \mathrm{D}_g^{t,\,l}(n)\right)$ is computable from m and r, it follows that the function $r : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined by

$$r(m,k) = \text{ the least } r \in \mathbb{N} \text{ such that } \Pr\left(\bigcap_{n=m}^{r} \mathrm{D}_g^{t,\,l}(n)\right) \leq 2^{-k}$$

is computable. For each $m, k \in \mathbb{N}$, define $d_{m,k} : \{0,1\}^* \to [0,1]$ by

$$d_{m,k}(w) = \Pr\left(\bigcap_{n=m}^{r(m,k)} \mathrm{D}_g^{t,\,l}(n) \;\middle|\; \mathbf{C}_w\right),$$

and define $d : \{0,1\}^* \to [0,\infty)$ by

$$d(w) = \sum_{m=0}^{\infty}\sum_{k=0}^{\infty} 2^{-m} d_{m,k}(w).$$

(Note that each $d_{m,k}(\lambda) \leq 2^{-k}$, so $d(\lambda) \leq 4$.) It is routine to check that each $d_{m,k}$ is an exact rec-martingale, whence d is a rec-martingale.

Let $S \in \mathrm{D}_g^{t,\,l}$, and let $a \in \mathbb{N}$ be arbitrary. Fix $m \in \mathbb{N}$ such that $S \in \bigcap_{n=m}^{\infty} \mathrm{D}_g^{t,\,l}(n)$, and let $r = r(m, 2^m \cdot a)$. Let $w = S[0..r-1]$. Then

$$\mathbf{C}_w \subseteq \bigcap_{n=m}^{r} \mathrm{D}_g^{t,\,l}(n),$$

so for all $0 \leq k \leq 2^m \cdot a$,

$$\mathbf{C}_w \subseteq \bigcap_{n=m}^{r(m,k)} \mathrm{D}_g^{t,\,l}(n),$$

whence $d_{m,k}(w) = 1$. It follows that

$$d(w) \geq 2^{-m} \sum_{k=0}^{r} d_{m,k}(w) = 2^{-m}(1 + 2^m \cdot a) > a.$$

Since $a \in \mathbb{N}$ is arbitrary here, this shows that $S \in S^{\infty}[d]$.

The preceding paragraph establishes that $\mathrm{D}_g^{t,\,l} \subseteq S^{\infty}[d]$, whence $\mu_{\mathrm{rec}}(\mathrm{D}_g^{t,\,l}) = 0$ □

Lemma 3.3. There exist a computable function $t(n) = O(n \log n)$ and a constant $c \in \mathbb{N}$ such that, for every computable function $l : \mathbb{N} \to \mathbb{N}$, $\mu_{\mathrm{rec}}(\mathrm{D}_c^{t,\,l}) = 0$.

Proof. Let t and c be as in Theorem 2.9, and let $l : \mathbb{N} \to \mathbb{N}$ be computable. Then $\mathrm{D}_c^{t,\,l} \subseteq \mathrm{D}_c^{t}$. so $\mathrm{RAND} \cap \mathrm{D}_{\mathrm{c}}^{\mathrm{t},\,\mathrm{l}} \subseteq \mathrm{RAND} \cap \mathrm{D}_{\mathrm{c}}^{\mathrm{t}} = \emptyset$, so $\Pr(\mathrm{D}_c^{t,\,l}) = 0$. It follows by Lemma 3.2 that $\mu_{\mathrm{rec}}(\mathrm{D}_c^{t,\,l}) = 0$. □

Theorem 3.4. RAND(rec) ∩ rec-strDEEP $= \emptyset$. In fact, there exist a computable function $t(n) = O(n \log n)$ and a constant $c \in \mathbb{N}$ such that $\mathrm{RAND(rec)} \cap \mathrm{D}_{\mathrm{c}}^{\mathrm{t},\,\mathrm{rec}} = \emptyset$.

Proof. Let t and c be as in Lemma 3.3. To see that $\mathrm{RAND(rec)} \cap \mathrm{D}_{\mathrm{c}}^{\mathrm{t},\,\mathrm{rec}} = \emptyset$, let $S \in \mathrm{D}_c^{t,\,\mathrm{rec}}$. Fix a computable function $l : \mathbb{N} \to \mathbb{N}$ such that $S \in \mathrm{D}_c^{t,\,l}$. Then, by Lemma 3.3, $\mu_{\mathrm{rec}}(\mathrm{D}_c^{t,\,l}) = 0$, so $S \notin \mathrm{RAND(rec)}$. □

As with strong depth, it is useful to have characterizations of rec-strong depth in terms of the time-bounded Kolmogorov complexities and algorithmic probabilities of prefixes. To this end, we define recursive analogs of the classes $\widehat{\mathrm{D}}_g^t$ and $\widetilde{\mathrm{D}}_g^t$ of [10].

Definition. For $t, g, l : \mathbb{N} \to \mathbb{N}$ and $n \in \mathbb{N}$, we define the sets

$$\widehat{\mathrm{D}}_g^{t,\,l}(n) = \left\{ S \in \mathbf{C} \,\middle|\, K^l(S[0..n-1]) \leq K^t(S[0..n-1]) - g(n) \right\},$$

$$\widehat{\mathrm{D}}_g^{t,\,l} = \left\{ S \in \mathbf{C} \,\middle|\, (\forall^{\infty} n)\, S \in \widehat{\mathrm{D}}_g^{t,\,l}(n) \right\},$$

$$\widehat{\mathrm{D}}_g^{t,\,\mathrm{rec}} = \left\{ S \in \mathbf{C} \,\middle|\, (\exists l \in \mathrm{rec})\, S \in \widehat{\mathrm{D}}_g^{t,\,l} \right\},$$

$$\widetilde{\mathrm{D}}_g^{t,\,l}(n) = \left\{ S \in \mathbf{C} \,\middle|\, \mathbf{m}^l(S[0..n-1]) \geq 2^{g(n)} \mathbf{m}^t(S[0..n-1]) \right\},$$

$$\widetilde{\mathrm{D}}_g^{t,\,l} = \left\{ S \in \mathbf{C} \,\middle|\, (\forall^{\infty} n)\, S \in \widetilde{\mathrm{D}}_g^{t,\,l}(n) \right\},$$

$$\widetilde{\mathrm{D}}_g^{t,\,\mathrm{rec}} = \left\{ S \in \mathbf{C} \,\middle|\, (\exists l \in \mathrm{rec})\, S \in \widetilde{\mathrm{D}}_g^{t,\,l} \right\}.$$

The following lemma is exactly analogous to Lemma 5.3 of [10], which is due to Bennett [4]. The proof is a straightforward adaptation of the proof in [10].

Lemma 3.5. If $t, l : \mathbb{N} \to \mathbb{N}$ are computable, then there exist constants $c_0, c_1, c_2 \in \mathbb{N}$ and computable functions $t_1, l_1, l_2 : \mathbb{N} \to \mathbb{N}$ such that the following nine conditions hold for all $g : \mathbb{N} \to \mathbb{N}$ and $n \in \mathbb{N}$.

1. $\mathrm{D}^{t,\,l}_{g+c_0}(n) \subseteq \widehat{\mathrm{D}}^{t,\,l_1}_{g}(n)$	4. $\mathrm{D}^{t,\,l}_{g+c_0} \subseteq \widehat{\mathrm{D}}^{t,\,l_1}_{g}$	7. $\mathrm{D}^{t,\,\mathrm{rec}}_{g+c_0} \subseteq \widehat{\mathrm{D}}^{t,\,\mathrm{rec}}_{g}$
2. $\widehat{\mathrm{D}}^{t_1,\,l}_{g+c_1}(n) \subseteq \widetilde{\mathrm{D}}^{t,\,l}_{g}(n)$	5. $\widehat{\mathrm{D}}^{t_1,\,l}_{g+c_1} \subseteq \widetilde{\mathrm{D}}^{t,\,l}_{g}$	8. $\widehat{\mathrm{D}}^{t_1,\,\mathrm{rec}}_{g+c_1} \subseteq \widetilde{\mathrm{D}}^{t,\,\mathrm{rec}}_{g}$
3. $\widetilde{\mathrm{D}}^{t,\,l}_{g+c_2}(n) \subseteq \mathrm{D}^{t,\,l_2}_{g}(n)$	6. $\widetilde{\mathrm{D}}^{t,\,l}_{g+c_2} \subseteq \mathrm{D}^{t,\,l_2}_{g}$	9. $\widetilde{\mathrm{D}}^{t,\,\mathrm{rec}}_{g+c_2} \subseteq \mathrm{D}^{t,\,\mathrm{rec}}_{g}$

This lemma immediately yields the following alternative characterizations of recursively strong depth.

Theorem 3.6. For $S \in \mathbf{C}$, the following conditions are equivalent.

1. S is rec-strongly deep.
2. For every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every constant $c \in \mathbb{N}$, $S \in \widehat{\mathrm{D}}^{t,\,\mathrm{rec}}_{c}$.
3. For every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every constant $c \in \mathbb{N}$, $S \in \widetilde{\mathrm{D}}^{t,\,\mathrm{rec}}_{c}$.

We now turn to recursively weak depth, which was introduced by Fenner, Lutz, and Mayordomo [6]. Recall from section 2.1 the definitions of tt-reductions and the set TTC of all tt-conditions.

Definition. A *uniform reducibility* is a computable function $F : \mathbb{N} \times \mathbb{N} \to \mathrm{TTC}$.

If F is a uniform reducibility, then we use the notation $F_k(n) = F(k, n)$, thereby regarding F as a computable sequence $F_0, F_1, F_2, \ldots$ of tt-reductions.

Definition. If F is a uniform reducibility and $A, B \in \mathbf{C}$, then A is *F-reducible to B*, and we write $A \leq_F B$, if there exists $k \in \mathbb{N}$ such that $A = F_k(B)$.

The following fact is well-known and easy to verify.

Observation 3.7.

1. For every computable function $t : \mathbb{N} \to \mathbb{N}$, there is a uniform reducibility F such that, for all $A, B \in \mathbf{C}$,

$$A \leq_F B \Longleftrightarrow A \leq_{\mathrm{T}}^{\mathrm{DTIME}(t)} B.$$

2. For every uniform reducibility F, there is a computable function $t : \mathbb{N} \to \mathbb{N}$ such that, for all $A, B \in \mathbf{C}$,

$$A \leq_F B \Longrightarrow A \leq_{\mathrm{T}}^{\mathrm{DTIME}(t)} B$$

Definition. If F is a uniform reducibility and $A \in \mathbf{C}$, then the *upper F-span* of A is the set

$$F^{-1}(A) = \left\{ B \in \mathbf{C} \,\middle|\, A \leq_F B \right\}.$$

Definition (Fenner, Lutz, and Mayordomo [6]). Let F be a uniform reducibility. A sequence $S \in \mathbf{C}$ is *recursively F-deep* (briefly, rec-*F-deep*), and we write $S \in$ rec-F-DEEP, if $\mu_{\mathrm{rec}}(F^{-1}(S)) = 0$.

Definition (Fenner, Lutz, and Mayordomo [6]). A sequence $S \in \mathbf{C}$ is *recursively weakly deep* (briefly, rec-*weakly deep*), and we write $S \in$ rec-wkDEEP, if, for every uniform reducibility F, S is rec-F-deep.

If S is a recursive sequence, then it is easy to see that there is a uniform reducibility F such that $F^{-1}(S) = \mathbf{C}$. (Intuitively, the reduction decides S without using the oracle.) It is thus immediate from the definition that no recursive sequence is rec-weakly deep.

The notion of rec-weak depth is analogous to the notion of weak depth, in the sense that (as is easily seen) a sequence $S \in \mathbf{C}$ is weakly deep if and only if, for every uniform reducibility F, the upper span $F^{-1}(S)$ has constructive measure 0. The following is also true.

Observation 3.8. No rec-weakly deep sequence is tt-reducible to a rec-random sequence.

Proof. Assume that $S \leq_{\mathrm{tt}} R \in \mathrm{RAND}(\mathrm{rec})$. Then there is a uniform reducibility F such that $R \in F^{-1}(S)$. Since R is rec-random, this implies that $\mu_{\mathrm{rec}}(F^{-1}(S)) \neq 0$, whence S is not rec-weakly deep. □

We do not know whether the converse of this observation holds, i.e., whether a sequence that is not tt-reducible to any rec-random sequence must be rec-weakly deep. As it is, however, Observation 3.8, together with the fact that $\mathrm{RAND} \subseteq \mathrm{RAND}(\mathrm{rec})$, tells us that every rec-weakly deep sequence is weakly deep.

Observation 3.9 (Fenner, Lutz, and Mayordomo [6]). rec-wkDEEP $\subseteq$ wkDEEP.

If F is any uniform reducibility such that the relation $\leq_F$ is reflexive, then by the proof of Observation 3.8, the set rec-F-DEEP must be disjoint from RAND(rec), and hence must have measure 0 in $\mathbf{C}$. However, the measure of rec-F-DEEP in REC is a different matter.

Theorem 3.10 (Fenner, Lutz, and Mayordomo [6]). If F is a uniform reducibility, then

$$\mu\left(\text{rec-}F\text{-DEEP} \,\middle|\, \mathrm{REC}\right) = 1,$$

i.e., almost every sequence in REC is recursively F-deep.

3.2 Class Inclusions

In this section, we establish the basic inclusion relations that hold among the weak and strong depth classes defined in sections 2.3 and 3.1. For this and later purposes, we need a technical lemma. This result, called the *deterministic slow-growth law for recursive computational depth*, places a quantitative upper bound on the ability of a time-bounded oracle Turing machine to amplify the depth of its oracle. The first slow-growth laws were proven by Bennett [4]. The slow-growth law here is a refinement of Lemma 5.5 of [10]. As in [10], we need two special notations. First, for any function $s : \mathbb{N} \to \mathbb{N}$, we define the function $s^* : \mathbb{N} \to \mathbb{N}$ by

$$s^*(n) = 2^{s(\lceil \log n \rceil)+1}.$$

Second, for any unbounded, nondecreasing function $f : \mathbb{N} \to \mathbb{N}$, we define the special-purpose "inverse" function $f^{-1} : \mathbb{N} \to \mathbb{N}$ by

$$f^{-1}(n) = \max\left\{m \,\middle|\, f(m) < n\right\}.$$

Also as in [10], say that a function $s : \mathbb{N} \to \mathbb{N}$ is *time-constructible* if there exist a constant $c_s \in \mathbb{N}$ and a Turing machine that, given the standard binary representation w of a natural number n, computes the standard binary representation of $s(n)$ in at most $c_s \cdot s(|w|)$ steps. Using standard techniques [2, 8], it is easy to show that, for every computable function $r : \mathbb{N} \to \mathbb{N}$, there is a strictly increasing, time-constructible function $s : \mathbb{N} \to \mathbb{N}$ such that, for all $n \in \mathbb{N}$, $r(n) \le s(n)$.

Lemma 3.11 (Slow Growth Lemma, version I). Let $s : \mathbb{N} \to \mathbb{N}$ be strictly increasing and time-constructible with the constant $c_s \in \mathbb{N}$ as witness. For each s-time-bounded oracle Turing machine M, there is a constant $c_M \in \mathbb{N}$ with the following property. Given nondecreasing functions $t, g, l : \mathbb{N} \to \mathbb{N}$, define the functions $\tau, \widehat{t}, \widehat{g}, \widehat{l} : \mathbb{N} \to \mathbb{N}$ by

$$\begin{aligned}
\tau(n) &= t(s^*(n+1)) + 4s^*(n+1) + 2(n+1)c_s \cdot s(\lfloor \log(n+1) \rfloor) + \\
&\qquad 2ns^*(n+1)s(\lfloor \log(n+1) \rfloor), \\
\widehat{t}(n) &= c_M(1 + \tau(n) \lceil \log \tau(n) \rceil), \\
\widehat{g}(n) &= g(s^*(n+1)) + c_M, \\
\widehat{l}(n) &= c_M(1 + l(\widehat{t}(n)) \log l(\widehat{t}(n))).
\end{aligned}$$

For all $A, B \in \mathbf{C}$, if $B \le_{\mathrm{T}}^{\mathrm{DTIME}(s)} A$ via M and $B \in \mathrm{D}_{\widehat{g}}^{\widehat{t}, l}$, then $A \in \mathrm{D}_{g}^{t, \widehat{l}}$.

Proof. To save space, we refer to the proof of Lemma 5.5 in [10].

Let s and M be as in the statement of the lemma. Let $M', \pi_{M'}, c_{M'}, M''$, and $\pi_{M''}$ be as in [10]. Since our universal Turing machine U is efficient, there is a constant $c_{M''} \in \mathbb{N}$ such that, for all $\pi^* \in \{0,1\}^*$,

$$time_U(\pi_{M''}\pi^*) \le c_{M''} \cdot (1 + time_{M''}(\pi^*) \log time_{M''}(\pi^*)).$$

Let

$$c_M = \max\{c_{M'}, c_{M''}, |\pi_{M'}| + |\pi_{M''}|\}.$$

Let $t, g, l : \mathbb{N} \to \mathbb{N}$ be nondecreasing, and define $\tau, \widehat{t}, \widehat{g}$, and $\widehat{l}$ as in the statement of the lemma. Assume that $A, B \in \mathbf{C}$ are such that $B \le_{\mathrm{T}}^{\mathrm{DTIME}(s)} A$ via M and $B \in \mathrm{D}_{\widehat{g}}^{\widehat{t}, l}$. Fix $n_0 \in \mathbb{Z}^+$ such that $B \in \mathrm{D}_{\widehat{g}}^{\widehat{t}, l}(n)$ for all $n \ge n_0$, and let $m_1 = s^*(n_0) + 1$.

Claim 1. For all $m > s^*(1)$ and $\pi \in \{0,1\}^*$, if $\pi \in \mathrm{PROG}^t(A[0..m-1])$, then $\pi_{M'}\pi \in \mathrm{PROG}^{\widehat{t}}(B[0..n-1])$, where $n = (s^*)^{-1}(m)$.

Claim 2. For all $m \ge m_1$ and all $\pi \in \mathrm{PROG}^t(A[0..m-1])$,

$$K^{\widehat{l}}(\pi) \le |\pi| - \widehat{g}(n) + c_M$$

where $n = (s^*)^{-1}(m)$.

Claim 1 was proven in [10]. To prove Claim 2, we again follow the proof in [10]. Let m, π, n, and π^* be as in [10]. Then

$$time_U(\pi^*) \leq l(|\pi_{M'}\pi|),$$

so

$$\begin{aligned} time_U(\pi_{M''}\pi^*) &\leq c_{M''} \cdot (1 + time_{M''}(\pi^*) \log time_{M''}(\pi^*)) \\ &= c_{M''} \cdot (1 + time_U(\pi^*) \log time_U(\pi^*)) \\ &\leq c_{M''} \cdot (1 + l(|\pi_{M'}\pi|) \log l(|\pi_{M'}\pi|)) \\ &\leq c_{M''} \cdot (1 + l(\widehat{t}(n)) \log l(\widehat{t}(n))) \\ &= \widehat{l}(n). \end{aligned}$$

Thus,

$$\begin{aligned} K^{\widehat{l}}(\pi) &\leq |\pi_{M''}\pi^*| = K^l(\pi_{M'}\pi) + |\pi_{M''}| \\ &\leq |\pi| - \widehat{g}(n) + c_M, \end{aligned}$$

verifying Claim 2.

To complete the proof of the lemma, let $m \geq m_1$, and let $\pi \in \mathrm{PROG}^t(A[0..m-1])$. Then, by Claim 2 and the monotonicity of g,

$$\begin{aligned} K^{\widehat{l}}(\pi) &\leq |\pi| - \widehat{g}((s^*)^{-1}(m)) + c_M \\ &= |\pi| - g(s^*((s^*)^{-1}(m) + 1)) \\ &\leq |\pi| - g(m). \end{aligned}$$

Thus $A \in \mathrm{D}_g^{t,\widehat{l}}(m)$. Since this holds for all $m \geq m_1$, it follows that $A \in \mathrm{D}_g^{t,\widehat{l}}$. □

The Slow Growth Lemma will often be used in the following slightly weaker form, which is an immediate consequence of Lemma 3.11

Lemma 3.12 (Slow Growth Lemma, version II). Let $s : \mathbb{N} \to \mathbb{N}$ be strictly increasing and time-constructible, with the constant $c_s \in \mathbb{N}$ as witness. For each s-time-bounded oracle Turing machine M, there is a constant $c_M \in \mathbb{N}$ with the following property. Given nondecreasing functions $t, g : \mathbb{N} \to \mathbb{N}$, define the functions $\tau, \widehat{t}, \widehat{g} : \mathbb{N} \to \mathbb{N}$ by

$$\begin{aligned} \tau(n) &= t(s^*(n+1)) + 4s^*(n+1) + 2(n+1)c_s \cdot s(\lfloor \log(n+1) \rfloor) + \\ &\qquad 2ns^*(n+1)s(\lfloor \log(n+1) \rfloor), \\ \widehat{t}(n) &= c_M(1 + \tau(n)\lceil \log \tau(n) \rceil), \\ \widehat{g}(n) &= g(s^*(n+1)) + c_M. \end{aligned}$$

For all $A, B \in \mathbf{C}$, if $B \leq_{\mathrm{T}}^{\mathrm{DTIME}(s)} A$ via M and $B \in \mathrm{D}_{\widehat{g}}^{\widehat{t},\,\mathrm{rec}}$, then $A \in \mathrm{D}_{g}^{t,\,\mathrm{rec}}$.

An easy consequence of the Slow Growth Lemma is the fact that the class of rec-strongly deep sequences is (like the class of strongly deep sequences [10]) closed upwards under tt-reductions.

Theorem 3.13. Let $A, B \in \mathbf{C}$. If $B \leq_{\mathrm{tt}} A$ and B is rec-strongly deep, then A is rec-strongly deep.

The proof of Theorem 3.13 is identical to the proof of Theorem 5.6 in [10], except that Lemma 3.12 above is used in place of Lemma 5.5 in [10].

We now come to the main result of section 3.2. The following theorem gives the inclusion relations that hold among the weak, strong, rec-weak, and rec-strong depth classes defined in sections 2.3 and 3.1.

Theorem 3.14. The following diagram of inclusions holds.

$$\begin{array}{ccccc} & & \text{rec-wkDEEP} & & \\ & \subseteq & & \supseteq & \\ \text{rec-strDEEP} & & & & \text{wkDEEP} \\ & \subseteq & & \supseteq & \\ & & \text{strDEEP} & & \end{array}$$

Proof. It was shown by Bennett [4] (see also [10]) that strDEEP $\subseteq$ wkDEEP, and Observations 3.1 and 3.9 tell us that rec-strDEEP $\subseteq$ strDEEP and rec-wkDEEP $\subseteq$ wkDEEP. All that remains, then, is to show that rec-strDEEP $\subseteq$ rec-wkDEEP.

Let $S \in$ rec-strDEEP, and let F be a uniform reducibility. Fix a strictly increasing, time-constructible function $s : \mathbb{N} \to \mathbb{N}$ such that, for all $A, B \in \mathbf{C}$,

$$A \leq_F B \Longrightarrow A \leq_{\mathrm{T}}^{\mathrm{DTIME}(s)} B.$$

Choose t, c as in Lemma 3.3. Let $g(n) = c$ and define $\widehat{t}$ and $\widehat{g}$ as in Lemma 3.11. Then $\widehat{g}(n)$ is constant; say $\widehat{g}(n) = \widehat{c}$. Now $S \in \mathrm{D}_{\widehat{c}}^{\widehat{t},\,\mathrm{rec}}$, so there is a computable function $l : \mathbb{N} \to \mathbb{N}$ such that $S \in \mathrm{D}_{\widehat{c}}^{\widehat{t},\,l}$. Define $\widehat{l}$ as in Lemma 3.11. Then Lemma 3.11 tells us that $F^{-1}(S) \subseteq \mathrm{D}_{c}^{t,\,\widehat{l}}$. By Lemma 3.3, $\mu_{\mathrm{rec}}(\mathrm{D}_{c}^{t,\,\widehat{l}}) = 0$, so $\mu_{\mathrm{rec}}(F^{-1}(S)) = 0$, i.e., S is rec-F-deep. Since F is arbitrary here, this shows that $S \in$ rec-wkDEEP. □

3.3 Class Separations

We now show that all four inclusions in Theorem 3.14 are proper. It is most efficient (and most informative) to prove this by proving the two non-inclusions

$$\text{strDEEP} \not\subseteq \text{rec-wkDEEP}$$

and

$$\text{rec-wkDEEP} \not\subseteq \text{strDEEP}.$$

We prove these in succession.

We prove that strDEEP $\subsetneqq$ rec-wkDEEP by proving the much stronger fact that, in contrast with Theorems 2.9 and 3.4, strongly deep sequences can be recursively random. We do this by examining the Kolmogorov and the time-bounded Kolmogorov complexities of recursively random sequences.

We first prove that rec-random sequences have very high time-bounded Kolmogorov complexities.

Theorem 3.15. Assume that S is rec-random and that $t, g : \mathbb{N} \to \mathbb{N}$ are computable functions with g nondecreasing and unbounded. Then, for all but finitely many $n \in \mathbb{N}$,

$$K^t(S[0..n-1]) > n - g(n).$$

Proof. Assume the hypothesis. For each $n \in \mathbb{N}$ and $w \in \{0,1\}^*$, let

$$\mathcal{E}_n = \left\{ A \,\middle|\, K^t(A[0..n-1]) \leq n - g(n) \right\}$$

and

$$d_n(w) = \Pr(\mathcal{E}_n \,|\, \mathbf{C}_w).$$

It suffices to show that the set

$$J = \left\{ n \in \mathbb{N} \,\middle|\, S \in \mathcal{E}_n \right\}$$

is finite.

It is easy to see that the function $(n, w) \mapsto d_n(w)$ is computable, and that each d_n is a martingale. Choose a constant $c \in \mathbb{N}$ as in Lemma 2.5, and define $m : \mathbb{N} \to \mathbb{N}$ by

$$m(r) = \text{ the least } m \in \mathbb{N} \text{ such that } g(m) \geq r + c.$$

Then m is computable, and for all $r \in \mathbb{N}$,

$$\begin{aligned}
\sum_{n=m(r)}^{\infty} d_n(\lambda) &= \sum_{n=m(r)}^{\infty} \Pr(\mathcal{E}_n) \\
&\leq \sum_{n=m(r)}^{\infty} \Pr\left[K(A[0..n-1]) \leq n - g(n)\right] \\
&< \sum_{n=m(r)}^{\infty} 2^{c-K(n)-g(n)} \\
&\leq \sum_{n=m(r)}^{\infty} 2^{c-K(n)-g(m(r))} \\
&\leq 2^{-r} \sum_{n=m(r)}^{\infty} 2^{-K(n)} \\
&\leq 2^{-r}.
\end{aligned}$$

Thus the series $\sum_{n=0}^{\infty} d_n(\lambda)$ is computably convergent. It follows by Corollary 2.3 that there are only finitely many $n \in \mathbb{N}$ such that $S \in S^1[d_n]$. Since, for all $n \in N$,

$$n \in J \implies d_n(S[0..n-1]) = 1 \implies S \in S^1[d_n],$$

it follows that J is finite. □

The function g above may be very slowly growing, e.g., an inverse Ackermann function. Theorem 3.15 thus says that, for every rec-random sequence S and computable time bound t, all but finitely many of the prefixes of S have K^t-complexities that are nearly as large as their lengths.

We next show that the situation is very different in the absence of the time bound t.

Definition. A sequence $S \in \mathbf{C}$ is *ultracompressible* if, for every computable, nondecreasing, unbounded function $g : \mathbb{N} \to \mathbb{N}$, there exists $n_g \in \mathbb{N}$ such that, for all $n \geq n_g$,

$$K(S[0..n-1]) < K(n) + g(n). \tag{3.1}$$

It is clear that every n-bit string w must satisfy $K(w) \geq K(n) - O(1)$. A sequence S is thus ultracompressible if, for every computable, nondecreasing, unbounded (but perhaps very slowly growing) function g, for all but finitely many n, the n-bit prefix of S has K-complexity that is within $g(n)$ bits of the minimum possible K-complexity for an n-bit string.

We now show that a rec-random sequence can be ultracompressible. Similar results have been proven by Wang [29] and Ambos-Spies and Wang [1] for the monotone Kolmogorov complexities of rec-random sequences. The present result is slightly stronger than these results in that it gives a single rec-random sequence S that has property (3.1) for every computable, nondecreasing, unbounded function g. The proof is based in part on a simpler, unpublished construction by Gasarch and Lutz [7] of a rec-random sequence that is not algorithmically random.

Theorem 3.16. There is a rec-random sequence that is ultracompressible.

Proof. Let $g_0, g_1, g_2, \ldots$ be an enumeration of all computable, nondecreasing, unbounded functions $g_k : \mathbb{N} \to \mathbb{N}$, and let $d_0, d_1, d_2, \ldots$ be an enumeration of all exact rec-martingales d_k with $d_k(\lambda) = 1$. (Both enumerations are necessarily noneffective.) For each $k \in \mathbb{N}$, fix a program prefix $\pi_{d_k} \in \{0,1\}^*$ such that, for all $w \in \{0,1\}^*$, $U(\pi_{d_k}\mathbf{sd}(w)) = d_k(w)$, where $\mathbf{sd}(w)$ is the self-delimiting encoding of w defined in section 2.1. For each $k \geq -1$, let $\pi_k^{(d)} = \langle \pi_{d_0}, \ldots, \pi_{d_k} \rangle$, where $\langle \ldots \rangle$ is the self-delimiting sequence encoding defined in section 2.1, and let $a_k = \left| \pi_k^{(d)} \right|$.

Our objective is to exhibit a rec-random sequence S that is ultracompressible. This sequence S is specified by a sequence

$$w_{-1} \sqsubsetneq w_0 \sqsubsetneq w_1 \sqsubsetneq w_2 \sqsubsetneq \ldots \sqsubseteq S$$

of prefixes w_k that are defined inductively below. There is a single Turing machine that carries out all of the extensions from w_k to w_{k+1}, given a suitable program at each stage w_k. We now describe this machine.

Fix a Turing machine M that, given a program of the form $\pi = \pi_k^{\langle w \rangle} \pi_{k+1}^{(d)} \pi_n$, where $k \geq -1$, $U(\pi_k^{\langle w \rangle}) = \langle w_0, \ldots, w_k \rangle$, $U(\pi_n) = s_n$, and $n \geq |w_k|$, outputs the encoded list $\langle w_0, \ldots, w_k, w(k,n) \rangle$, where $w(k,n) \in \{0,1\}^n$ is the string whose i^{th} bit is given by the recursion

$$w(k,n)[i] = \begin{cases} w_k[i] & \text{if } 0 \leq i < |w_k| \\ \\ [\![\widetilde{d}_k(w(k,n)[0..i-1]1) \leq \widetilde{d}_k(w(k,n)[0..i-1]0)]\!] & \text{if } |w_k| \leq i < n, \end{cases}$$

where

$$\widetilde{d}_k(w) = \sum_{j=0}^{k+1} 2^{-(j+|w_{j-1}|)} d_j(w)$$

and $w_{-1} = \lambda$. (If the program π for M is not of the above form, then $M(\pi)$, which may or may not be defined, is not used in this proof.)

In more intuitive terms, given such a program π, M extends w_k one bit at a time, choosing the bit that minimizes the composite martingale $\widetilde{d}_k$ at each step of the extension. In particular, it is evident that

$$\widetilde{d}_k(w(k,n)) \leq \widetilde{d}_k(w_k). \tag{3.2}$$

As defined below, the extended prefix w_{k+1} is precisely the string $w(k, n_k)$ for a suitable value of n_k. The rec-randomness of S is then ensured by (3.2), while the ultracompressibility of S is ensured by a judicious choice of n_k.

Fix a constant $c \in \mathbb{N}$ such that, for all $k \in \mathbb{N}$ and all $w_0, \ldots, w_k \in \{0,1\}^*$,

$$K(w_k) \leq K(\langle w_0, \ldots, w_k \rangle) + c; \tag{3.3}$$

and, for all $x \in \{0,1\}^*$,

$$K(x) \leq K_M(x) + c. \tag{3.4}$$

Define the sequence

$$w_{-1} \sqsubsetneq w_0 \sqsubsetneq w_1 \sqsubsetneq w_2 \sqsubsetneq \cdots$$

inductively as follows. First, let $w_{-1} = \lambda$. Next, assume that $w_{-1} \sqsubsetneq \cdots \sqsubsetneq w_k$ have been defined, where $k \geq -1$. For each $n \geq |w_k|$, let

$$\pi(k,n) = \pi_k^{\langle w \rangle} \pi_{k+1}^{(d)} \pi_n,$$

where $\pi_k^{\langle w \rangle}$ is a minimum-length program for $\langle w_0, \ldots, w_k \rangle$ and π_n is a minimum-length program for s_n, and let $w(k,n)$ be the (unique) string such that

$$M(\pi(k,n)) = \langle w_0, \ldots, w_k, w(k,n) \rangle .$$

Note that, for all $k \geq -1$ and $n \geq |w_k|$,

$$K\left(\langle w_0, \ldots, w_k, w(k,n) \rangle\right) \leq K\left(n\right) + K\left(\langle w_0, \ldots, w_k \rangle\right) + a_{k+1} + c. \tag{3.5}$$

This is because, by (3.4), $K\left(\langle w_0, \ldots, w_k, w(k,n) \rangle\right) = K\left(M\left(\pi\left(k,n\right)\right)\right) \leq K_M\left(M\left(\pi\left(k,n\right)\right)\right) + c \leq \left|\pi\left(k,n\right)\right| + c = K\left(n\right) + K\left(\langle w_0, \ldots, w_k \rangle\right) + a_{k+1} + c$.

Define $\widetilde{g} : \mathbb{N} \to \mathbb{N}$ by

$$\widetilde{g}(n) = \min_{0 \leq j \leq k+1} \left\lfloor \frac{g_j(n)}{2} \right\rfloor .$$

Then $\widetilde{g}$ is computable and unbounded, so by Theorem 2.6 there exist infinitely many $n \in \mathbb{N}$ such that $K(n) < \widetilde{g}(n)$. Thus we can fix $n_k > |w_k|$ such that

$$K(n_k) < \widetilde{g}(n_k) \tag{3.6}$$

and

$$K(\langle w_0, \ldots, w_k \rangle) + a_{k+1} + a_{k+2} + 3c < \widetilde{g}(n_k) . \tag{3.7}$$

Let $w_{k+1} = w(k, n_k)$. This completes the definition of the sequence $w_{-1} \sqsubsetneq w_0 \sqsubsetneq w_1 \sqsubsetneq \cdots$.

For all $0 \le j \le l+1$, by (3.5), (3.6), and (3.7),

$$\begin{aligned} K(\langle w_0, \ldots, w_{l+1} \rangle) + a_{l+2} + 2c &= K(\langle w_0, \ldots, w_l, w(l, n_l) \rangle) + a_{l+2} + 2c \\ &\le K(n_l) + K(\langle w_0, \ldots, w_l \rangle) + a_{l+1} + a_{l+2} + 3c \\ &< 2\widetilde{g}(n_l) \\ &= 2\widetilde{g}(|w_{l+1}|) \\ &\le g_j(|w_{l+1}|) . \end{aligned}$$

It follows by the change of variable $k = l + 1$ that, for all $0 \le j \le k$,

$$K(\langle w_0, \ldots, w_k \rangle) + a_{k+1} + 2c < g_j(|w_k|) . \tag{3.8}$$

We next show that, for all $k \ge -1$,

$$\widetilde{d}_k(w_k) \le 2 - 2^{-(k+1)} \tag{3.9}$$

We prove this by induction on k. It clearly holds for $k = -1$; assume that it holds for k. Then, by (3.2), Lemma 2.1, and (3.9),

$$\begin{aligned} \widetilde{d}_{k+1}(w_{k+1}) &= \widetilde{d}_k(w_{k+1}) + 2^{-(k+2+|w_{k+1}|)} d_{k+2}(w_{k+1}) \\ &\le \widetilde{d}_k(w_k) + 2^{-(k+2)} \\ &\le 2 - 2^{-(k+1)} + 2^{-(k+2)} \\ &= 2 - 2^{-(k+2)}, \end{aligned}$$

so it holds for $k + 1$.

Now let S be the unique sequence such that $w_k \sqsubseteq S$ for all $k \in \mathbb{N}$. We show that S is rec-random and ultracompressible.

To see that S is rec-random, let d be an exact rec-martingale with $d(\lambda) = 1$. Fix $j \in \mathbb{N}$ such that $d_j = d$. Then, for all $k > j$, (3.9) tells us that the prefix w_k of S

satisfies

$$\begin{aligned} d(w_k) &= 2^{j+|w_{j-1}|}2^{-(j+|w_{j-1}|)}d_j(w_k) \\ &\leq 2^{j+|w_{j-1}|}\widetilde{d}_k(w_k) \\ &\leq 2^{j+|w_{j-1}|}\left(2-2^{-(k+1)}\right) \\ &< 2^{j+|w_{j-1}|+1}. \end{aligned}$$

It follows by Lemma 2.4 that S is rec-random.

Finally, to see that S is ultracompressible, let $g : \mathbb{N} \to \mathbb{N}$ be computable, nondecreasing, and unbounded. Fix $j \in \mathbb{N}$ such that $g_j = g$, and let $n \geq |w_j|$. Fix $k \in \mathbb{N}$ such that $|w_k| \leq n < |w_{k+1}|$. Then, by (3.3), (3.5), (3.8), and the fact that g is nondecreasing,

$$\begin{aligned} K\left(S[0..n-1]\right) &= K(w(k,n)) \\ &\leq K\left(\langle w_0, \ldots, w_k, w(k,n)\rangle\right) + c \\ &\leq K(n) + K\left(\langle w_0, \ldots, w_k\rangle\right) + a_{k+1} + 2c \\ &< K(n) + g_j(|w_k|) \\ &\leq K(n) + g(n). \end{aligned}$$

Hence S is ultracompressible. □

We now note that rec-random sequences can be strongly deep.

Theorem 3.17. There is a rec-random sequence that is strongly deep.

Proof. By Theorem 3.16, there is a rec-random sequence S that is ultracompressible. To see that S is strongly deep, fix a computable function $t : \mathbb{N} \to \mathbb{N}$ and a constant $c \in \mathbb{N}$. By Theorem 2.10, it suffices to show that $S \in \widehat{\mathrm{D}}_c^t$.

Fix a real number α such that $0 < \alpha < 1$, and define $g : \mathbb{N} \to \mathbb{N}$ by

$$g(n) = \left\lfloor \frac{(1-\alpha)n}{3} \right\rfloor .$$

Then g is computable, nondecreasing, and unbounded, so by Theorem 3.15, there exists $n_1 \in \mathbb{N}$ such that, for all $n \geq n_1$,

$$K^t(S[0..n-1]) > n - g(n). \tag{3.10}$$

Also, since S is ultracompressible, there exists $n_2 \in \mathbb{N}$ such that, for all $n \geq n_2$,

$$K(S[0..n-1]) < K(n) + g(n). \tag{3.11}$$

Finally, there exists $n_3 \in \mathbb{N}$ such that, for all $n \geq n_3$,

$$K(n) \leq g(n). \tag{3.12}$$

Let $n_0 = \max\{n_1, n_2, n_3\}$. Then, for all $n \geq n_0$, (3.10), (3.11) and (3.12) tell us that

$$K^t(S[0..n-1]) - K(S[0..n-1]) > n - 3g(n) \geq \alpha n.$$

Hence, $S \in \widehat{\mathrm{D}}^t_{\alpha n} \subseteq \widehat{\mathrm{D}}^t_c$. □

The rec-random sequence S given by the above proof is not only strongly deep, but is in the class $\widehat{\mathrm{D}}^t_{\alpha n}$ for all computable time bounds t. Since the real number α may be arbitrarily close to 1, this says that S is strongly deep at very high significance levels (significance levels very close to n bits).

Theorem 3.17 contrasts sharply with Theorems 2.9 and 3.4. There is of course nothing paradoxical in this contrast. It is merely a consequence of the strong, quantitative separation of RAND(rec) from RAND given by Theorems 2.7 and 3.16.

We now have the first of the desired noninclusions.

Corollary 3.18. strDEEP $\not\subseteq$ rec-wkDEEP.

Proof. By Theorem 3.17, there is a sequence S that is rec-random and strongly deep. Since S is rec-random, Observation 3.8 tells us that S is not rec-weakly deep. □

Our proof that strDEEP $\not\subseteq$ rec-wkDEEP uses Baire category and Banach-Mazur games. We briefly review the relevant ideas.

A Banach-Mazur game is a two-player, infinite game in which the players construct a sequence $S \in \mathbf{C}$ by taking turns extending a prefix of S. There is a "payoff set" $X \subseteq \mathbf{C}$ such that player I wins a play of the game if $S \in X$, and player II wins if $S \notin X$. A *strategy* for a Banach-Mazur game is a function $\sigma : \{0,1\}^* \to \{0,1\}^*$ such that, for all $w \in \{0,1\}^*$, $w \sqsubsetneq \sigma(w)$, i.e., $\sigma(w)$ is a proper extension of w. A *play* of a Banach-Mazur game is an ordered pair (α, β) of strategies. For $t \in \mathbb{N}$, the t^{th} *partial result* of a play (α, β) is the string $R_t(\alpha, \beta) \in \{0,1\}^*$ defined by the following recursion.

(i) $R_0(\alpha, \beta) = \lambda$.

(ii) For all $i \in \mathbb{N}$, $R_{2i+1}(\alpha, \beta) = \alpha(R_{2i}(\alpha, \beta))$.

(iii) For all $i \in \mathbb{N}$, $R_{2i+2}(\alpha, \beta) = \beta(R_{2i+1}(\alpha, \beta))$.

(Player I uses strategy α, and player II uses strategy β.) The *result* of a play (α, β) is the unique sequence $R(\alpha, \beta) \in \mathbf{C}$ such that , for all $t \in \mathbb{N}$, $R_t(\alpha, \beta) \sqsubseteq R(\alpha, \beta)$. We write $G[X; \mathcal{S}_I, \mathcal{S}_{II}]$ for the Banach-Mazur game with payoff set X in which Player I is required to use a strategy from the set $\mathcal{S}_I$ of functions and player II is required to use a strategy from the set $\mathcal{S}_{II}$ of functions. In this paper, the sets of functions that we are interested in are the set rec, consisting of all computable functions from $\{0,1\}^*$ into $\{0,1\}^*$, and the set all, consisting of all functions from $\{0,1\}^*$ into $\{0,1\}^*$. We write $G[X]$ for $G[X; \text{all}, \text{all}]$.

A *winning strategy for player I* in a Banach-Mazur game $G[X; \mathcal{S}_I, \mathcal{S}_{II}]$ is a strategy $\alpha \in \mathcal{S}_I$ such that, for every strategy $\beta \in \mathcal{S}_{II}$, $R(\alpha, \beta) \in X$. A *winning strategy for player II* in a Banach-Mazur game $G[X; \mathcal{S}_I, \mathcal{S}_{II}]$ is a strategy $\beta \in \mathcal{S}_{II}$ such that, for every strategy $\alpha \in \mathcal{S}_I$,$R(\alpha, \beta) \notin X$.

Definition (Mazur and Banach [21]). Let $X \subseteq \mathbf{C}$.

1. X is *meager* if there is a winning strategy for player II in the Banach-Mazur game $G[X]$.
2. X is *comeager* if X^{c} is meager.

A meager set is sometimes called a *set of first category*, or a *set of first category in the sense of Baire*.

Definition (Lutz [17]). Let $X \subseteq \mathbf{C}$.

1. X is rec-*meager* if there is a winning strategy for player II in the Banach-Mazur game $G[X; \text{all}, \text{rec}]$.
2. X is rec-*comeager* if X^{c} is rec-meager.

Definition (Lisagor [16], Lutz [17]). Let $X \subseteq \mathbf{C}$.

1. X is *meager in* REC if $X \cap \text{REC}$ is rec-meager.
2. X is *comeager in* REC if X^{c} is meager in REC.

For $X \subseteq \mathbf{C}$, the implications

$$\begin{array}{ccc} X \text{ is rec-meager} & \Longrightarrow & X \text{ is meager} \\ \Big\Downarrow & & \\ X \text{ is meager in REC} & & \end{array}$$

are clear. It is also clear that every subset of a meager set is meager and that every countable set $X \subseteq \mathbf{C}$ is meager. In fact, it is well known that every countable union of meager sets is meager [21]. On the other hand, the Baire Category Theorem [21] says that no cylinder is meager. These facts justify the intuition that meager sets are *negligibly small in the sense of Baire category.* Thus, if a set $X \subseteq \mathbf{C}$ is comeager, we say that X contains *almost every* sequence *in the sense of Baire category.*

The situation is analogous for sets that are meager in REC. Every subset of a set that is meager in REC is clearly meager in REC. Lisagor [16] has also shown that every *recursive union* (a natural, effective notion of countable union) of sets that are meager in REC is meager in REC and, more importantly, that no cylinder is meager in REC. These facts justify the intuition that, if $X \in \mathbf{C}$ is a set that is meager in REC, then $X \cap \mathrm{REC}$ is a *negligibly small subset* of REC *in the sense of Baire category.* Similarly, if X is comeager in REC, then X contains *almost every* sequence in REC *in the sense of Baire category.*

It is well-known [21, 16] that a set may be large in the sense of measure but small in the sense of Baire category, or vice versa.

The following known theorem says that the set of strongly deep sequences is small in the sense of Baire category.

Theorem 3.19 (Juedes, Lathrop, and Lutz [10]). The class strDEEP is meager.

We show that rec-wkDEEP $\not\subseteq$ strDEEP by showing that rec-wkDEEP is comeager. Our proof of this fact is somewhat more involved than the proof by Juedes, Lathrop, and Lutz [10] that wkDEEP is comeager.

Theorem 3.20. For each uniform reducibility F, the class rec-F-deep is rec-comeager, hence comeager in REC.

Proof. Let F be a uniform reducibility. For each $n \in \mathbb{Z}^+$, let $a(n) = \frac{1}{6}n(n-1)(2n-1)$, so that $a(n) + n^2 = a(n+1)$. For each $n \in \mathbb{Z}^+$ and $0 \leq k < n$, let

$$I_n(k) = \left\{ a(n) + kn + m \;\middle|\; 0 \leq m < n \right\}.$$

Note that the intervals

$$I_1(0), I_2(0), I_2(1), I_3(0), I_3(1), I_3(2), I_4(0), \ldots$$

partition $\mathbb{N}$ into successive blocks, with each $|I_n(k)| = n$.

For each $n \in \mathbb{Z}^+$, $0 \le k < n$, $x \in \{0,1\}^{\le n}$, and $A \in \mathbf{C}$, say that A *agrees with* x *on* $I_n(k)$ if

$$A[\, a(n) + kn \,..\, a(n) + kn + |x| - 1 \,] = x.$$

For each $n \in \mathbb{Z}^+$, $0 \le k < n$, and $x \in \{0,1\}^{\le n}$, define the event

$$\mathcal{E}_{k,n,x} = \left\{ B \in \mathbf{C} \,\middle|\, F_k(B) \text{ agrees with } x \text{ on } I_n(k) \right\}.$$

For each $n \in \mathbb{Z}^+$ and $0 \le k < n$, let $y_n(k)$ be the n-bit string whose l^{th} bit is defined by the recursion

$$y_n(k)[l] = [\![\Pr(\mathcal{E}_{k,n,z1}) < \Pr(\mathcal{E}_{k,n,z0})]\!]$$

for all $0 \le l < n$, where $z = y_n(k)[0..l-1]$. This definition ensures that

$$\Pr(\mathcal{E}_{k,n,y_n(k)[0..l]}) \le \frac{1}{2} \Pr(\mathcal{E}_{k,n,y_n(k)[0..l-1]}). \tag{3.13}$$

For each $n \in \mathbb{Z}^+$ and $0 \le k < n$, define the event

$$\mathcal{E}_{k,n} = \mathcal{E}_{k,n,y_n(k)}.$$

Then, by (3.13), for all $n \in \mathbb{Z}^+$ and $0 \le k < n$,

$$\Pr(\mathcal{E}_{k,n}) \le 2^{-n}. \tag{3.14}$$

Let

$$Y = \left\{ A \in \mathbf{C} \,\middle|\, (\forall k)(\exists^\infty n) A \text{ agrees with } y_n(k) \text{ on } I_n(k) \right\}.$$

It suffices to prove that

$$Y \subseteq \text{rec-}F\text{-DEEP} \tag{3.15}$$

and

$$Y \text{ is rec-comeager.} \tag{3.16}$$

We first prove (3.15). For each $k, n \in \mathbb{N}$, define the function $d_{k,n} : \{0,1\}^* \to [0,1]$ by

$$d_{k,n}(w) = \begin{cases} \Pr(\mathcal{E}_{k,n} | \mathbf{C}_w) & \text{if } 0 \le k < n \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to check that each $d_{k,n}$ is a martingale, and that the function $(k, n, w) \mapsto d_{k,n}(w)$ is total recursive (with rational values). Also, by (3.14),

$$d_{k,n}(\lambda) \leq 2^{-n} \tag{3.17}$$

for all $k, n \in \mathbb{N}$. It follows by Theorem 2.2 that

$$\mu_{\text{rec}}\left(\bigcup_{k=0}^{\infty}\bigcap_{m=0}^{\infty}\bigcup_{n=m}^{\infty} S^1[d_{k,n}]\right) = 0. \tag{3.18}$$

To prove (3.15), let $A \in Y$. Let $B \in F^{-1}(A)$. Fix $k \in \mathbb{N}$ such that $A = F_k(B)$. Since $A \in Y$, the set

$$J_k = \left\{ n > k \,\middle|\, A \text{ agrees with } y_n(k) \text{ on } I_n(k) \right\}$$

is infinite. Let $n \in J_k$. Then $B \in \mathcal{E}_{k,n}$. In fact, since F_k is a tt-reduction, there is a prefix $w \sqsubseteq B$ such that $\mathbf{C}_w \subseteq \mathcal{E}_{k,n}$. Then $d_{k,n}(w) = \Pr(\mathcal{E}_{k,n}|\mathbf{C}_w) = 1$, so $B \in S^1[d_{k,n}]$. Since J_k is infinite, this argument shows that

$$F^{-1}(A) \subseteq \bigcup_{k=0}^{\infty}\bigcap_{m=0}^{\infty}\bigcup_{n=m}^{\infty} S^1[d_{k,n}]. \tag{3.19}$$

It follows from (3.18) and (3.19) that $\mu_{\text{rec}}(F^{-1}(A)) = 0$, i.e., that $A \in$ rec-F-DEEP. This proves (3.15).

Finally, to prove (3.16), define a strategy β for player II in the Banach-Mazur game $G[Y^c; \text{all}; \text{rec}]$ as follows. Given $w \in \{0,1\}^*$, fix the least $n \in \mathbb{Z}^+$ such that $a(n) \geq |w|$, and set

$$\beta(w) = w0^{a(n)-|w|}y_n(0)y_n(1)\cdots y_n(n-1).$$

It is clear that $\beta \in$ rec and, for every strategy α that player I might use, $R(\alpha, \beta) \in Y$. Hence, β is a winning strategy for player II in $G[Y^c;\text{all,rec}]$. It follows that Y^c is rec-meager, whence (3.16) holds. □

Theorem 3.21. The class rec-wkDEEP is comeager.

Proof. The class rec-wkDEEP is a countable intersection of classes rec-F-DEEP, each of which is rec-comeager, hence comeager, by Theorem 3.20. □

Corollary 3.22. rec-wkDEEP $\not\subseteq$ strDEEP.

Proof. This follows immediately from Theorems 3.19 and 3.21. □

We now have the main result of section 3.3.

Theorem 3.23. The following diagram of proper inclusions holds.

$$\begin{array}{ccccc} & & \text{rec-wkDEEP} & & \\ & \rotatebox{45}{$\subsetneq$} & & \rotatebox{-45}{$\subsetneq$} & \\ \text{rec-strDEEP} & & & & \text{wkDEEP} \\ & \rotatebox{-45}{$\subsetneq$} & & \rotatebox{45}{$\subsetneq$} & \\ & & \text{strDEEP} & & \end{array}$$

Proof. This follows immediately from Theorem 3.14, and Corollaries 3.18 and 3.22. □

By Theorem 3.23, there exist sequences that are strongly deep, but not rec-strongly deep. Let S be such a sequence. Since S is not rec-strongly deep, there exist a *fixed* computable time bound $t_0 : \mathbb{N} \to \mathbb{N}$ and a *fixed* constant $c_0 \in \mathbb{N}$ such that, for *every* computable time bound $l : \mathbb{N} \to \mathbb{N}$, there are infinitely many prefixes of S that *cannot* be described c_0 bits more succinctly with the time bound l than with the time bound t_0. Nevertheless, since S is strongly deep, it must be the case that, for *every* constant $c \in \mathbb{N}$ (even when c is much greater than c_0), all but finitely many prefixes of S can be described at least c bits more succinctly without a time bound than with the time bound t_0. *None* of this additional succinctness (beyond c_0 bits) can be realized within any computable time bound; *all* of it requires greater-than-computable running time. The depth of such a sequence S appears not to come from so much from a nontrivial causal (computational) history as from something utterly noncomputational.

If F is a uniform reducibility that is (like all standard reducibilities) reflexive, then the measure and category of the class rec-F-DEEP are of some interest. First, rec-F-DEEP must be disjoint from RAND(rec), so rec-F-DEEP must be a measure 0 subset of **C**. Also, by Theorem 3.20, rec-F-DEEP must be comeager. Thus, the class rec-F-DEEP is small in the sense of measure, but large in the sense of Baire category. This state of affairs is not unusual and would not be worth mention, were it not for the fact that the situation changes when we look at the measure and category of rec-F-DEEP in REC. By Theorems 3.10 and 3.20, rec-F-DEEP is large in REC in the senses of both measure and category. The class rec-F-DEEP is thus one concerning which measure and category agree in REC, but disagree in **C**.

3.4 Weakly Useful Sequences

Juedes, Lathrop, and Lutz [10] defined the class of *weakly useful* sequences and proved that every weakly useful sequence is strongly deep. Fenner, Lutz, and Mayordomo [6] subsequently proved that every weakly useful sequence is rec-weakly deep. In this section, we strengthen both these results by proving that every weakly useful sequence is rec-strongly deep. Our argument closely follows that of [10], but it is short and central, so we present it in full.

Definition (Juedes, Lathrop, and Lutz [10]). A sequence $A \in \mathbf{C}$ is *strongly useful*, and we write $A \in$ strUSEFUL, if there is a computable time bound $s : \mathbb{N} \to \mathbb{N}$ such that $\mathrm{REC} \subseteq \mathrm{DTIME}^A(s)$. A sequence $A \in \mathbf{C}$ is *weakly useful*, and we write $A \in$ wkUSEFUL, if there is a computable time bound $s : \mathbb{N} \to \mathbb{N}$ such that $\mathrm{DTIME}^A(s)$ does not have measure 0 in REC.

Thus a sequence is strongly useful if it enables one to solve all decidable sequences in some fixed, computable amount of time. A sequence is weakly useful if it enables one to solve all elements of a nonnegligible set of decidable sequences in some fixed, computable amount of time.

Recall that the diagonal halting problem is the sequence K whose n^{th} bit is

$$K[n] = [\![M_n(n) \text{ halts}]\!],$$

where $M_0, M_1, \ldots$ is a standard enumeration of all deterministic Turing machines. It is well-known that K is polynomial-time many-one complete for the set of all recursively enumerable subsets of $\mathbb{N}$, so K is strongly useful.

It is clear that every strongly useful sequence is weakly useful. Fenner, Lutz, and Mayordomo [6] used martingale diagonalization to construct a sequence that is weakly useful but not strongly useful, so strUSEFUL $\subsetneqq$ wkUSEFUL.

Our proof that every weakly useful sequence is rec-strongly deep uses the following theorem, which is a recursive strengthening of Theorem 5.8 of [10]. Recall the class ${\mathbf{K}^t}_{\text{i.o.}}[< g(n)]$ defined in section 2.2.

Theorem 3.24. If $t : \mathbb{N} \to \mathbb{N}$ is computable and $0 < \alpha < \beta < 1$, then

$$\mathrm{REC} \subseteq \widehat{\mathrm{D}}^{t,\,\mathrm{rec}}_{\alpha n} \cup {\mathbf{K}^t}_{\text{i.o.}}[< \beta n].$$

Proof. Assume the hypothesis and let

$$S \in \mathrm{REC} - {\mathbf{K}^t}_{\text{i.o.}}[< \beta n].$$

We will show that $S \in \widehat{\mathrm{D}}^{t,\,\mathrm{rec}}_{\alpha n}$.

Since $S \notin \mathbf{K}^t{}_{\text{i.o.}}[< \beta n]$, it must be the case that, for all but finitely many n,

$$K^t(S[0..n-1]) \geq \beta n.$$

Since S is recursive, there is a Turing machine M' such that, for all $n \in \mathbb{N}$, $M'(\mathbf{sd}(s_n)) = S[0..n-1]$, where $\mathbf{sd}(s_n)$ is the self-delimiting version of s_n, the n^{th} string in the standard enumeration of $\{0,1\}^*$.

Now let $\pi_{M'}$ be a program prefix for U such that for all $\pi \in \{0,1\}^*$,

$$U(\pi_{M'}\pi) = M'(\pi).$$

In particular, we have

$$U(\pi_{M'}\mathbf{sd}(s_n)) = M'(\mathbf{sd}(s_n)) = S[0..n-1].$$

Let $l : \mathbb{N} \to \mathbb{N}$ give the running time of U on these programs, i.e.,

$$l(n) = \mathit{time}_U(\pi_{M'}\mathbf{sd}(s_n)).$$

Then l is computable and, for all but finitely many $n \in \mathbb{N}$,

$$\begin{aligned} K^l(S[0..n-1]) &\leq |\pi_{M'}\mathbf{sd}(s_n)| \\ &= 2\lfloor \log(n+1) \rfloor + 2 + |\pi_{M'}| \\ &< \beta n - \alpha n \\ &\leq K^t(S[0..n-1]) - \alpha n, \end{aligned}$$

so $S \in \widehat{\mathrm{D}}^{t,\,\mathrm{rec}}_{\alpha n}$. □

Corollary 3.25. For every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every $0 < \gamma < 1$,

$$\mu\left(\mathrm{D}^{t,\,\mathrm{rec}}_{\gamma n} \middle| \mathrm{REC}\right) = 1.$$

Proof. Let $t : \mathbb{N} \to \mathbb{N}$ be computable, and let $0 < \gamma < \alpha < \beta < 1$. Choose a computable time bound $t_1 : \mathbb{N} \to \mathbb{N}$ for t and constants $c_1, c_2 \in \mathbb{N}$ as in Lemma 3.5, so that for all $n \in \mathbb{N}$,

$$\widehat{\mathrm{D}}^{t_1,\,\mathrm{rec}}_{\gamma n + c_2 + c_1}(n) \subseteq \widetilde{\mathrm{D}}^{t,\,\mathrm{rec}}_{\gamma n + c_2}(n) \subseteq \mathrm{D}^{t,\,\mathrm{rec}}_{\gamma n}(n).$$

For all sufficiently large n, we have

$$\widehat{\mathrm{D}}^{t_1,\,\mathrm{rec}}_{\alpha n}(n) \subseteq \widehat{\mathrm{D}}^{t_1,\,\mathrm{rec}}_{\gamma n + c_2 + c_1}(n),$$

so $\widehat{\mathrm{D}}^{t_1,\,\mathrm{rec}}_{\alpha n} \subseteq \mathrm{D}^{t,\,\mathrm{rec}}_{\gamma n}$.

By Theorem 2.8 $\mathbf{K}^{t_1}_{\text{i.o.}}[< \beta n]$ has measure 0 in REC. Combined with Theorem 3.24, this implies that $\widehat{\mathrm{D}}^{t_1,\,\mathrm{rec}}_{\alpha n}$ has measure 1 in REC. Since $\widehat{\mathrm{D}}^{t_1,\,\mathrm{rec}}_{\alpha n} \subseteq \mathrm{D}^{t,\,\mathrm{rec}}_{\gamma n}$, it follows that $\mathrm{D}^{t,\,\mathrm{rec}}_{\gamma n}$ has measure 1 in REC. □

Corollary 3.26. For every computable time bound $t : \mathbb{N} \to \mathbb{N}$ and every constant $c \in \mathbb{N}$,

$$\mu\left(\mathrm{D}^{t,\,\mathrm{rec}}_{c} \middle| \mathrm{REC}\right) = 1.$$

We now establish the rec-strong depth of weakly useful sequences.

Theorem 3.27. Every weakly useful sequence is rec-strongly deep.

Proof. Let $A \in \mathbf{C}$ be weakly useful. To see that A is rec-strongly deep, let $t : \mathbb{N} \to \mathbb{N}$ be a computable time bound, and let $c \in \mathbb{N}$. It suffices to show that $A \in \mathrm{D}^{t,\,\mathrm{rec}}_{c}$.

Since A is weakly useful, there is a computable time bound $s : \mathbb{N} \to \mathbb{N}$ such that $\mathrm{DTIME}^A(s)$ does not have measure 0 in REC. Since every computable function is bounded above by a strictly increasing, time-constructible function, we can assume without loss of generality that s is strictly increasing and time-constructible.

Let $\tilde{t}(n) = n\cdot(1+\tau(n)\lceil \log \tau(n)\rceil)$, where τ is defined from t and s as in Lemma 3.12, and let $\gamma = \frac{1}{2}$. Since $\tilde{t}$ is recursive, Corollary 3.25 tells us that $\mathrm{D}^{\tilde{t},\,\mathrm{rec}}_{\gamma n}$ has measure 1 in REC. Since $\mathrm{DTIME}^A(s)$ does not have measure 0 in REC, it follows that $\mathrm{D}^{\tilde{t},\,\mathrm{rec}}_{\gamma n} \cap \mathrm{DTIME}^A(s) \neq \emptyset$. Fix a sequence $B \in \mathrm{D}^{\tilde{t},\,\mathrm{rec}}_{\gamma n} \cap \mathrm{DTIME}^A(s)$. Then there is an s-time-bounded oracle Turing machine M such that $B \leq^{\mathrm{DTIME}(s)}_{\mathrm{T}} A$ via M. Fix a constant c_M as in Lemma 3.12. Define $g(n) = c$ for all $n \in \mathbb{N}$ and define the functions $\tau, \widehat{t}$, and $\widehat{g}$ from t and g as in Lemma 3.12. Since $\widehat{g}$ and c_M are constant, we have $\tilde{t}(n) > \widehat{t}(n)$ and $\gamma n > \widehat{g}(n)$ for all but finitely many n, so $B \in \mathrm{D}^{\tilde{t},\,\mathrm{rec}}_{\gamma n}(\subseteq)\mathrm{D}^{\widehat{t},\,\mathrm{rec}}_{\widehat{g}}$. It follows by Lemma 3.12 that $A \in \mathrm{D}^{t,\,\mathrm{rec}}_{c}$. □

Juedes [9] asked whether every strongly deep sequence is weakly useful. We can now answer this question negatively.

Corollary 3.28. wkUSEFUL $\subsetneqq$ strDEEP

Proof. This follows immediately from Theorems 3.23 and 3.27. □

4 Conclusion

The results of this paper, together with earlier results of Bennett [4], Juedes, Lathrop, and Lutz [10], and Fenner, Lutz, and Mayordomo [6], establish the following relationships.

$$\text{strUSEFUL} \subsetneqq \text{wkUSEFUL} \subseteq \text{rec-strDEEP} \subsetneqq \text{rec-wkDEEP} \subsetneqq \text{wkDEEP}$$
$$\text{rec-strDEEP} \subsetneqq \text{strDEEP} \subsetneqq \text{wkDEEP}$$

We conjecture that the inclusion wkUSEFUL $\subseteq$ rec-strDEEP is also proper, i.e., that rec-strong depth is not a sufficient condition for weak usefulness.

As noted in section 3.1, we do not know whether a sequence that is not truth-table reducible to any rec-random sequence must be rec-weakly deep. The question here is whether the upper truth-table span of a sequence S can avoid the set RAND(rec) while $F^{-1}(S)$ fails to have rec-measure 0 for some uniform reducibility F. The answer to this question may shed new light on recursive measure.

Beyond these specific questions, it is to be hoped that further investigation of computational depth will lead to a better understanding of the role that information plays in the complexity of computation.

Acknowledgments

We thank Bas Terwijn, David Juedes, Bill Gasarch, and Giora Slutzki for useful discussions. We also thank David Juedes for helpful remarks on a preliminary draft of this paper.

References

[1] K. Ambos-Spies and Y. Wang. Algorithmic randomness concepts: a comparison. Talk by K. Ambos-Spies at the Workshop on Information and Randomness in Complexity Classes, Schloss Dagstuhl, Germany, July 17, 1996.

[2] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I (second edition).* Springer-Verlag, Berlin, 1995.

[3] C. H. Bennett. Dissipation, information, computational complexity and the definition of organization. In D. Pines, editor, *Emerging Syntheses in Science, Proceedings of the Founding Workshops of the Santa Fe Institute*, pages 297–313, 1985.

[4] C. H. Bennett. Logical depth and physical complexity. In R. Herken, editor, *The Universal Turing Machine: A Half-Century Survey*, pages 227–257. Oxford University Press, 1988.

[5] G. J. Chaitin. A theory of program size formally identical to information theory. *Journal of the Association for Computing Machinery*, 22:329–340, 1975.

[6] S. A. Fenner, J. H. Lutz, and E. Mayordomo. Weakly useful sequences. In *Proceedings of the* 22^{nd} *International Colloquium on Automata, Languages, and Programming*, pages 393–404. Springer–Verlag, 1995.

[7] W. I. Gasarch and J. H. Lutz. Unpublished manuscript, 1991.

[8] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[9] D. W. Juedes. *The Complexity and Distribution of Computationally Useful Problems.* PhD thesis, Department of Computer Science, Iowa State University, 1994.

[10] D. W. Juedes, J. I. Lathrop, and J. H. Lutz. Computational depth and reducibility. *Theoretical Computer Science*, 132:37–70, 1994.

[11] D. W. Juedes and J. H. Lutz. Weak completeness in E and E_2. *Theoretical Computer Science*, 143:149–158, 1995.

[12] J. I. Lathrop. Compression depth and the behavior of cellular automata. *Complex Systems*, 1997. To appear.

[13] L. A. Levin. On the notion of a random sequence. *Soviet Mathematics Doklady*, 14:1413–1416, 1973.

[14] L. A. Levin. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problems of Information Transmission*, 10:206–210, 1974.

[15] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag, 1993.

[16] L. R. Lisagor. The Banach-Mazur Game. Translated in *Math. USSR Sbornik*, 38:201–206, 1981.

[17] J. H. Lutz. Category and measure in complexity classes. *SIAM Journal on Computing*, 19:1100–1131, 1990.

[18] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44:220–258, 1992.

[19] P. Martin-Löf. On the definition of random sequences. *Information and Control*, 9:602–619, 1966.

[20] E. Mayordomo. *Contributions to the study of resource-bounded measure.* PhD thesis, Universitat Politècnica de Catalunya, 1994.

[21] J. C. Oxtoby. *Measure and Category.* Springer-Verlag, second edition, 1980.

[22] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw - Hill, New York, 1967.

[23] H. L. Royden. *Real Analysis,* third edition. Macmillan Publishing Company, New York, 1988.

[24] C. P. Schnorr. A unified approach to the definition of random sequences. *Mathematical Systems Theory*, 5:246–258, 1971.

[25] C. P. Schnorr. Zufälligkeit und Wahrscheinlichkeit. *Lecture Notes in Mathematics*, 218, 1971.

[26] C. P. Schnorr. Process complexity and effective random tests. *Journal of Computer and System Sciences*, 7:376–388, 1973.

[27] R. J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7:1–22, 224–254, 1964.

[28] M. van Lambalgen. *Random Sequences.* PhD thesis, Department of Mathematics, University of Amsterdam, 1987.

[29] Y. Wang. *Randomness and Complexity.* PhD thesis, Department of Mathematics, University of Heidelberg, 1996.

[30] A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25:83–124, 1970.