PROMPT ENUMERATIONS AND RELATIVE RANDOMNESS

ANTHONY MORPHETT

Abstract. The 'dynamic' property of prompt simplicity has become an influential and important concept in the study of the c.e. Turing degrees. The equivalent property of prompt permitting is a particularly fruitful notion, arising neatly from the technique of Yates permitting. We introduce an analogous notion, *prompt non-low-for-randomness*. Prompt non-low-for-randomness is a prompt form of non-low-for-random permitting, which is the natural notion of permitting in the context of relative randomness. We begin an investigation of this notion by showing that the class of Turing degrees of promptly non-low-for-random c.e. sets is non-trivial and a proper subclass of the non-low-for-random promptly simple degrees. We also consider prompt non-low-for-randomness in the context of the LR-degrees and LR-reducibility, a weak reducibility arising naturally from relative randomness.

1. Randomness and relative randomness

Algorithmic randomness has been a popular and fruitful topic of research in recent years. Studies of the interactions between randomness and computability-theoretic notions have given rise to many new techniques, concepts and classes. In this paper we will suggest a new concept, a notion of promptness for computably enumerable sets in the context of relative algorithmic randomness, analogous to the well-known notion of prompt simplicity. We begin to study the class of such c.e. sets and investigate the relationship between this class and the promptly simple Turing degrees.

Notions of algorithmic randomness attempt to formalise the idea of data being random as far as can be detected by algorithmic methods. Although this can be done in several ways, such as by formalising gambling strategies in the form of effective martingales or via compressibility, one of the most successful notions of randomness is that proposed by Martin-Löf [19] in terms of effective nullsets (measure 0 sets). A *Martin-Löf test* is a sequence of uniformly Σ^0_1 classes $(U_i)_{i\in\mathbb{N}}$ such that

$$U_{i+1} \subseteq U_i \text{ and } \mu(U_i) \leq 2^{-i}$$

where μ denotes the usual Lebesgue measure on Cantor space 2^ω. A sequence $X \in 2^\omega$ is *Martin-Löf random* if X is not in the intersection of any Martin-Löf test, ie.,

$$X \notin \cap_i U_i \text{ for any Martin-Löf test } (U_i)_{i\in\mathbb{N}}.$$

Since there are only countably many Martin-Löf tests, and the intersection of each is a nullset, the Martin-Löf random sequences form a set of measure 1. The intersection of a Martin-Löf test can be thought of as an effective nullset; a Martin-Löf random sequence is therefore one which *avoids all effective nullsets*. We will henceforth just say 'random' to mean Martin-Löf random. A comprehensive treatment of Martin-Löf randomness and related topics may be found in [21] or [11].

An important property of Martin-Löf randomness is the existence of a universal Martin-Löf test. Let $(U^k_i)_{i\in\mathbb{N}}$ for $k \in \mathbb{N}$ be a computable listing of all Martin-Löf tests, so that U^k_i is the i'th member of the k'th test. Then

$$U_i := \bigcup_k U^k_{i+k+1} \tag{1}$$

for $i \in \mathbb{N}$ gives a universal Martin-Löf test: X is random iff $X \notin \cap U_i$.

Date: December 28, 2010.

The definitions of Martin-Löf tests and randomness may be relativised to an arbitrary oracle $A \in 2^\omega$. We can thereby study the information content of A by examining the relative randomness notions obtained by relativising to A. One particularly interesting complexity class obtained from such study of relative randomness is the class of *low-for-randoms*. An oracle $A \in 2^\omega$ is low-for-random if the set of sequences random relative to A (the A-randoms) contains the set of (unrelativised) random sequences. In other words, the randomness notion obtained by relativising to A is exactly the same as the unrelativised randomness notion; the oracle A is of no use for effectively approximating nullsets compared to the unrelativised case. A non-computable c.e. low-for-random set was constructed by Kučera and Terwijn [16], and the class of low-for-randoms has since been extensively studied. In particular, low-for-randomness was shown in [22] to co-incide with two classes arising naturally in the context of Kolmogorov complexity: the K-trivials and the low-for-K sets. A further equivalence was established in [13] with the class of bases for randomness, those sets A which are computable from a sequence X which is random relative to A. A presentation of these results may be found in [21].

Given two oracles A and B, we may compare their information content by comparing the relative randomness notions they induce. A natural way to do this is with the LR-reducibility. Say that A is LR-reducible to B, $A \leq_{LR} B$, if the set of B-random sequences is contained in the set of A-random sequences. More intuitively, $A \leq_{LR} B$ if any sequence that is not A-random is also not B-random; the oracle B is at least as good for effectively approximating nullsets as the oracle A. LR-reducibility gives rise to a degree structure in the usual way, by setting $A \equiv_{LR} B$ iff $A \leq_{LR} B$ and $B \leq_{LR} A$. An LR-degree is an equivalence class under $\equiv_{LR}$. Note that the least LR-degree consists exactly of the low-for-random sets. LR-reducibility was defined by Nies [22], and LR-reducibility and degrees were further studied in [4, 5, 20].

Since Turing reducibility implies LR-reducibility, $\leq_{LR}$ is a weakening of $\leq_T$. The existence of non-computable low-for-random sets establishes that it is a proper weakening. However, $\leq_{LR}$ retains many similarities to $\leq_T$. For instance, both are Σ^0_3 predicates (shown for LR for instance by Theorem 2 below). Also, many techniques from the study of c.e. and Δ^0_2 Turing degrees can be adapted to work in the context of LR-degrees, such as Sacks restraints and coding methods as in [4]. Permitting techniques can also be adapted to LR-degrees and non-low-for-randomness; the technique of *non-low-for-random permitting*, first used in [7], has been used by Barmpalias in [3] to show that each non-zero Δ^0_2 LR-degree bounds uncountably many LR-degrees and in [2] to show that there are no minimal pairs in the Δ^0_2 or c.e. LR-degrees. The technique of non-low-for-random permitting is outlined in section 3.

The main motivation of the work in this paper is to investigate whether promptness notions such as prompt simplicity (which has proved a very important concept in the c.e. Turing degrees) can be adapted to the context of relative randomness, non-low-for-randomness and LR-degrees. In section 3 we propose a notion of *prompt non-low-for-randomness*, and begin an investigation into the class of promptly non-low-for-random c.e. degrees. Since the notion of prompt simplicity is so fruitful in the c.e. Turing degrees, it is reasonable to ask whether the analogous notion of prompt non-low-for-randomness might be similarly fruitful in the context of relative randomness. In section 5 and 6 we investigate the extent to which results and techniques from the study of promptly simple degrees may be adapted to the promptly non-low-for-random degrees, and the connections between the two notions. In section 7 we discuss the relationship between prompt non-low-for-randomness and the LR-degrees.

We typically use the letters σ, τ, ρ etc to denote finite binary strings from $2^{<\omega}$, and A, X, etc to denote infinite binary sequences from 2^ω. We identify elements of 2^ω with subsets of $\mathbb{N}$ in the usual way, and often use the terms *sequence*, *set* or *oracle* synonymously. For strings σ, τ, the notation $\sigma \subset \tau$ denotes that σ is a proper initial segment of τ, and $\sigma \subseteq \tau$ has the obvious meaning of $\sigma \subset \tau$ or $\sigma = \tau$. We also write $\sigma \subset X$ to denote that the finite string σ is an initial segment of the sequence X. A finite string σ can represent the basic clopen set

$$[\sigma] := \{X \in 2^\omega : \sigma \subset X\},$$

and likewise a set of strings $V \subseteq 2^{<\omega}$ can represent the open set

$$[V] := \cup_{\sigma \in V}[\sigma].$$

When dealing with sets of strings and the corresponding open subsets of Cantor space, we usually omit the brackets, since it is clear from context whether we are talking about finite strings or subsets of Cantor space.

It is convenient to work with an alternate formulation of Martin-Löf randomness to that given above. For a set of strings $V \subseteq 2^{<\omega}$, define the *weight of* V,

$$\text{weight } V := \sum_{\sigma \in V} 2^{-|\sigma|}.$$

Say that V is *bounded* if weight $V < \infty$. A *Solovay test* is a bounded c.e. set of strings $V \subseteq 2^{<\omega}$. A sequence $X \in 2^{\omega}$ passes the Solovay test V if

$$\text{there are only finitely many } \sigma \in V \text{ s.t. } \sigma \subset X,$$

ie. the set U contains only finitely many initial segments of X. The following characterisation of Martin-Löf randomness is due to Solovay [25].

Theorem 1. $X \in 2^{\omega}$ is Martin-Löf random iff X passes all Solovay tests.

As with Martin-Löf tests, there is a universal Solovay test; by treating a Σ^0_1 class as a prefix-free c.e. set of strings, one could take, for example,

$$V = \cup_i U_i \tag{2}$$

where $(U_i)_{i \in \mathbb{N}}$ is the universal Martin-Löf test (1).

A key characterisation of low-for-randomness and LR-reducibility is the following, combining results of Kjos-Hanssen [14] and Kjos-Hanssen, Miller and Solomon [15]. A proof may be found in [21].

Theorem 2. Let $A, B \in 2^{\omega}$. The following are equivalent:

(i) $A \leq_{LR} B$;

(ii) for every A-Σ^0_1 class T^A with $\mu(T^A) < 1$ there is a B-Σ^0_1 class V^B with
$$\mu(V^B) < 1 \text{ and } T^A \subseteq V^B;$$

(iii) for some member U^A of a universal A-Martin-Löf test, there is a B-Σ^0_1 class V^B with
$$\mu(V^B) < 1 \text{ and } U^A \subseteq V^B;$$

(iv) for every bounded A-c.e. set of strings $W^A \subseteq 2^{<\omega}$, there is a bounded B-c.e. set of strings V^B with
$$W^A \subseteq V^B;$$

(v) for a universal A-Solovay test $U^A \subseteq 2^{<\omega}$, there is a bounded B-c.e. set of strings V^B with
$$U^A \subseteq V^B.$$

Note that if A is low-for-random, then $A \leq_{LR} \emptyset$, and so the statement that A is low-for-random is equivalent to (ii)-(v) with 'B-c.e.' or 'B-Σ^0_1' replaced with just 'c.e.' or 'Σ^0_1', respectively (this specific case of Theorem 2 was present in [16]). In particular, if A is not low-for-random, V is a c.e. set and U^A is a universal A-Solovay test, then

$$U^A \subseteq V \;\Rightarrow\; \text{weight } V = \infty.$$

In the following discussion and constructions, the suffix $[s]$ denotes the value of a parameter or quantity at the end of stage s of a construction.

2. Prompt simplicity and Yates permitting

The main notion of this paper is prompt non-low-for-randomness (Definition 5), which will be defined in section 3. Before we give the definition however, we discuss prompt simplicity, permitting and prompt permitting, to establish some terminology that will be useful in section 3, and to make clear the parallel between prompt simplicity and prompt non-low-for-randomness. We first give some background on prompt simplicity, and then discuss prompt (Yates) permitting, a property equivalent to prompt simplicity. We will define the 'permission set' and the 'prompt permission set' for prompt permitting. In section 3 we outline the technique of non-low-for-random permitting, and after defining the permission set and prompt permission set for non-low-for-random permitting, we give a definition of prompt non-low-for-randomness (Definition 5). At the end of section 3 we give some conditions equivalent to Definition 5.

Let W_e ($e \in \mathbb{N}$) be a standard listing of all c.e. sets, with a uniformly computable enumeration $W_e[s]$ such that $W_e = \cup_s W_e[s]$. We call this the *canonical listing* of c.e. sets. By a suitable coding, we can consider the W_e as sets of numbers or sets of strings as appropriate. An enumeration of a c.e. set A is a computable sequence $A[s]$ of finite sets such that $A[0] = \emptyset$, $A[s] \subseteq A[s+1]$ and $A = \cup_s A[s]$. The number x *enters* A *at* s if $x \in A[s] - A[s-1]$. In the following, whenever we work with a c.e. set A, we actually work with a particular enumeration $A[s]$ of A. However we will usually suppress the enumeration: when we say 'a c.e. set A' we mean a c.e. set A along with an enumeration $A[s]$ of A. We similarly assume without further mention that c.e. operators U (considered as c.e. sets of axioms) come with a particular enumeration $U[s]$.

In the 1980s, certain constructions in connection with structural properties of the c.e. Turing degrees aroused interest in dynamic properties of enumerations of c.e. sets. Maass [17] defined the notion of prompt simplicity: a c.e. set A is *promptly simple* if it is co-infinite and there is a computable function f such that

$$\forall e \qquad W_e \text{ infinite } \Rightarrow \exists x, s : x \text{ enters } W_e \text{ at stage } s \text{ and } x \in A[f(s)].$$

This notion was further studied in Maass, Shore and Stob [18] and in an influential paper by Ambos-Spies, Jockusch, Shore and Soare [1] (a presentation of this work may be found in [24]). Say that a Turing degree is promptly simple if it contains a promptly simple set. Several important results were proved in [1] about the promptly simple Turing degrees. In particular, they prove that the promptly simple Turing degrees form a strong filter in the c.e. Turing degrees[1], and the non-promptly simple degrees form an ideal in the c.e. Turing degrees[2]. They also prove that the 'dynamic' property of being promptly simple is equivalent to two structural properties which are definable in the first-order theory of the c.e. Turing degrees: being noncappable[3] and being low cuppable[4]. Prompt simplicity is thus a very fruitful notion in the study of c.e. Turing degrees.

Another dynamic property equivalent to prompt simplicity is *prompt permitting*.[5] Let us say that a c.e. set A is *(Yates) permitting* if for every infinite c.e. set W_e the set

$$P^Y(W_e) = \{x \in W_e : x \text{ enters } W_e \text{ at some stage } s \text{ and } A[s] \upharpoonright x \neq A \upharpoonright x\}$$

[1] A set $\mathcal{F}$ in an upper semi-lattice $\mathcal{L}$ is a filter if $\mathcal{F}$ is closed upwards and in taking greatest lower bounds (when they exist); it is a strong filter if it is closed upwards and every pair of elements in $\mathcal{F}$ bound a third element in $\mathcal{F}$.

[2] A set $\mathcal{I}$ in an upper semi-lattice $\mathcal{L}$ is an ideal if it is closed downwards and in taking least upper bounds.

[3] A c.e. Turing degree $\mathbf{a}$ is noncappable if there is no nonzero c.e. Turing degree $\mathbf{b}$ such that $\mathbf{a} \cap \mathbf{b} = \mathbf{0}$.

[4] A c.e. Turing degree $\mathbf{a}$ is low cuppable if there is a c.e. Turing degree $\mathbf{b}$ such that $\mathbf{b}' = \mathbf{0}'$ and $\mathbf{a} \cup \mathbf{b} = \mathbf{0}'$.

[5] Permitting is a technique that, given a c.e. set A with some non-computability property (such as 'being non-computable' or 'being non-low-for-random'), allows one to build a second c.e. set $B \leq_T A$ (with some other desirable properties) by ensuring that the approximation to $B \upharpoonright n$ changes at a certain stage only if the approximation to $A \upharpoonright n$ also changes at the same stage. The non-computability property of A guarantees that $A \upharpoonright n$ will change sufficiently often for us to construct B with the desired properties. This is typically done by enumerating a 'request set' W during a construction, where the number $n \in W$ represents a request that the approximation to $A \upharpoonright n$ change. Permitting was originally used by Friedberg [12] and Yates [26], using the property that A is non-computable, and in this form is known as Yates permitting.

is infinite[6]. The set $P^Y(W)$ is the *permission set* for W; if we think of W as a set of requests enumerated during a construction (where $n \in W$ represents a request that the approximation to $A \restriction n$ change), then $P^Y(W)$ is the set of successful requests. It is easy to verify that for c.e. sets, being Yates permitting is equivalent to being non-computable; in other words, constructions involving Yates permitting can be done below any non-computable c.e. set.

The notion of prompt permitting arises from Yates permitting by requiring that the changes to the approximation of $A \restriction n$ occur within a computable time. Although the non-computability of A guarantees that infinitely many requests for changes will succeed, it provides no indication of how long we might have to wait for any particular request to succeed. The notion of prompt permitting is obtained by requiring that the successful requests succeed within a computable time interval. Fix the c.e. set A and let $f : \mathbb{N} \to \mathbb{N}$ be a computable function such that $f(s) > s$. Let

$$\mathrm{PP}^Y_f(W_e) = \{x \in W_e : x \text{ enters } W_e \text{ at some stage } s \text{ and } A[s] \restriction x \neq A[f(s)] \restriction x\}. \tag{3}$$

$\mathrm{PP}^Y_f(W_e)$ is the set of *prompt permissions* from W_e, with respect to the function f. The c.e. set A is *promptly permitting* if there is a computable function f such that $f(s) > s$ and

$$\forall e \qquad W_e \text{ infinite } \;\Rightarrow\; \mathrm{PP}^Y_f(W_e) \text{ infinite.} \tag{4}$$

In [1] it is shown that degrees of promptly permitting c.e. sets are exactly the promptly simple degrees.

Theorem 3 ([1])**.** Let A be a c.e. set. The following are equivalent:

(i) A has promptly simple degree;
(ii) there is a computable function f such that for all e,

$$W_e \text{ infinite } \;\Rightarrow\; \mathrm{PP}^Y_f(W_e) \text{ infinite ;}$$

(iii) there is a computable function g such that for all e,

$$W_e \text{ infinite } \;\Rightarrow\; \mathrm{PP}^Y_g(W_e) \neq \emptyset.$$

Although the property of prompt simplicity lends its name to this important class of degrees, the property of prompt permitting is in practice the more useful notion. Although in the remainder of the chapter we talk of prompt simplicity and promptly simple degrees, we really think of prompt permitting as the more basic property.

Note that condition (4) only concerns c.e. sets W_e from the canonical listing. Suppose X is a c.e. set. If we wish to measure promptness of permissions from X, we must do so *relative to the canonical enumeration* $W_e[s]$ of some W_e such that $W_e = X$, and not relative to a non-canonical enumeration $X[s]$. The usual way to do this is by the Slowdown Lemma.

Lemma 4 (Slowdown Lemma, [24])**.** Let X_e be a sequence of c.e. sets with a uniformly computable enumeration $X_e = \cup_s X_e[s]$. There is a computable function $q : \mathbb{N} \to \mathbb{N}$ such that for all e,

$$W_{q(e)} = X_e \text{ and } W_{q(e)}[s+1] \cap \big(X_e[s+1] - X_e[s]\big) = \emptyset.$$

That is, when we construct a sequence of c.e. sets X_e, we can computably obtain canonical indexes for c.e. sets $W_{q(e)}$ such that numbers enter $W_{q(e)}$ *strictly later* than they enter X_e.

Proof. By the Recursion Theorem with parameters we can define q by

$$W_{q(e)} = \{x : \exists s \; (x \in X_e[s] - W_{q(e)}[s])\}.$$

□

[6] In the notation $P^Y(W)$ (and similar notations used later) there is an implicit dependence on A; however since A is usually understood to be fixed, and to avoid additional subscripts we omit explicit mention of the set A. The superscript Y for 'Yates' is to distinguish it from the non-low-for-random permission set $P(W)$ we will define later.

3. Non-low-for-random permitting and prompt permitting

In the context of relative randomness and LR-degrees, the most natural form of permitting is *non-low-for-random permitting*. This was first used in [7], and further developed by Barmpalias in [3] and [2]. The idea of non-low-for-random permitting is as follows. Let A be a non-low-for-random c.e. set. By Theorem 2, there is an A-c.e. set of strings U^A with weight < 1 such that, for any c.e. set of strings W,

$$U^A \subseteq W \;\Rightarrow\; \text{weight } W = \infty. \tag{5}$$

We can use this fact to force changes in the c.e. set A. Suppose at stage s during some construction we wish some number $< n$ to be enumerated into A, so that the approximation to $A \upharpoonright n$ will change. We can request a change by taking a string σ from $U^A[s]$ with use $\leq n$ and enumerating σ into a c.e. set W. If we do this repeatedly, we threaten to make $U^A \subseteq W$. By (5), if we succeed in making $U^A \subseteq W$ then we must have weight $W = \infty$. Since the weight of U^A is finite, the strings in $W - U^A$ must contribute infinite weight to W. Each of these strings $\sigma \in W - U^A$ corresponds to a successful change of some initial segment of A, as we had $\sigma \in U^A[s]$ at the stage when σ was put into W. Hence we are guaranteed enough successful A-changes to ensure weight $W = \infty$. This technique was used for instance in [3] to show that every non-zero Δ^0_2 LR-degree has uncountably many predecessors in the LR-degrees.

Note the following point about the technique of non-low-for-random permitting as sketched above. If we enumerate a string σ into W at stage s, then we have $\sigma \in U^A[s]$. Let u be the use of $\sigma \in U^A[s]$. The request corresponding to σ succeeds as soon as $A \upharpoonright u$ changes at a stage after s. Suppose W is a given c.e. set (as opposed to one that we enumerate during a permitting construction). If a string σ is enumerated into W at stage s but $\sigma \notin U^A[s]$, then, as far as permitting is concerned, σ is irrelevant until σ appears in $U^A[t]$ at some $t > s$ (if ever). This observation motivates the following definition of W^*. Fix the c.e. sets A and W, and the universal A-c.e. set U^A. As we approximate $U^A[s]$ in a Σ^0_2 way, we can approximate the c.e. set

$$W^* = \{\sigma : \exists s\ (\sigma \in W[s] \cap U^A[s])\}$$

via the enumeration

$$W^*[s] = \big\{\sigma : \exists t \leq s\ (\sigma \in W[t] \cap U^A[t])\big\}. \tag{6}$$

That is, a string σ is enumerated into W^* at stage s if s is the first stage at which σ is in *both* $W[s]$ and $U^A[s]$. Note that if $U^A \subseteq W$ then $U^A \subseteq W^*$, and if A is non-low-for-random and $U^A \subseteq W$ then weight W^* is infinite. For the purposes of non-low-for-random permitting, W^* is equivalent to W.

With A fixed, let

$$P_{U^A}(W) = \{\sigma : \sigma \text{ enters } W^* \text{ at some stage } s,\\ \sigma \in U^A[s] \text{ with use } u \text{ and } A[s] \upharpoonright u \neq A \upharpoonright u\}.$$

This is the permission set for W for non-low-for-random permitting, the set of strings from W that are permitted via U^A. The 'non-low-for-random permitting principle' (5) can be expressed as

$$U^A \subseteq W \;\Rightarrow\; \text{weight } P_{U^A}(W) = \infty.$$

We can now formulate a notion of prompt non-low-for-random permitting. Let f be a computable function such that $f(s) > s$. By analogy with (3), we can define the *prompt permitting set* for W with respect to U^A and f,

$$\mathrm{PP}_{U^A,f}(W) = \big\{\sigma : \sigma \text{ enters } W^* \text{ at some stage } s,\\ \sigma \in U^A[s] \text{ with use } u \text{ and } A[s] \upharpoonright u \neq A[f(s)] \upharpoonright u\big\}. \tag{7}$$

When the function f and/or class U^A are understood to be fixed, we can omit the subscripts. With this notation established, we can now give a definition of prompt non-low-for-randomness.

Definition 5. Let A be a c.e. set. A is *promptly non-low-for-random* if there is an A-c.e. set U^A such that weight $U^A < 1$ and a computable function $f : \mathbb{N} \to \mathbb{N}$ such that, for all e,

$$U^A \subseteq W_e \ \Rightarrow \ \text{weight } \mathrm{PP}_{U^A,f}(W_e) = \infty. \tag{8}$$

We say that A *is promptly non-low-for-random via* U^A, f if U^A and f satisfy (8). This definition asserts that if U^A is contained in a c.e. request set W_e, then the requests which succeed promptly (w.r.t. f) will have infinite weight.

As with the definition of promptly (Yates) permitting, Definition 5 only concerns c.e. sets from the canonical enumeration. Note that in this case we measure promptness relative to the enumeration of W_e^*, which depends not only on the canonical enumeration of W_e but also on the Σ_2^0 approximation of U^A.

It is clearly equivalent to require only that weight $U^A < \infty$ rather than weight $U^A < 1$ in Definition 5. We now show that we can equivalently replace the condition

$$\text{weight } \mathrm{PP}_{U^A,f}(W_e) = \infty$$

with

$$\text{weight } \mathrm{PP}_{U^A,f}(W_e) \geq 1.$$

Theorem 6. Let A be a c.e. set and U be a c.e. operator such that weight $U^A < 1$. Then the following are equivalent:

(i) there is a computable function g such that for all e,
$$U^A \subseteq W_e \Rightarrow \text{weight } \mathrm{PP}_{U^A,g}(W_e) = \infty;$$
(ii) there is a computable function f such that for all e,
$$U^A \subseteq W_e \Rightarrow \text{weight } \mathrm{PP}_{U^A,f}(W_e) \geq 1.$$

Proof. (i) implies (ii) is immediate, so we prove (ii) implies (i). Assume (ii). We may assume without loss of generality that f is strictly increasing and $f(s) > s$ for all s. Let W_e^* be a delayed enumeration of a subset of W_e such that σ enters W_e^* at s iff σ appears in $W_e[s] \cap U^A[s]$ for the first time at s. We will define the computable function g satisfying (i).

Let $2^{<k}$ denote the set of all binary strings of length $< k$, and let $D_i, i \in \mathbb{N}$ be a standard computable listing of all finite sets of strings. Define the c.e. set $X_{e,k,i}$ by the enumeration

$$X_{e,k,i}[s] = D_i \cup \{\sigma : |\sigma| \geq k \ \wedge \ \sigma \in W_e^*[s]\}.$$

In the limit we have

$$X_{e,k,i} = \left(W_e^* - 2^{<k}\right) \cup D_i,$$

and strings of length $\geq k$ enter $X_{e,k,i}$ at the same stage as they enter W_e^*. Let $q(e,k,i)$ be a computable function obtained from the Slowdown Lemma 4 such that $W_{q(e,k,i)} = X_{e,k,i}$ and strings enter $W_{q(e,k,i)}$ strictly later than they enter $X_{e,k,i}$. Note that for every k there is some i such that

$$D_i = U^A \cap 2^{<k} = W_{q(e,k,i)} \cap 2^{<k}.$$

We now define g. Fix s, and let

$$Z = \{\langle e, \sigma\rangle : \sigma \text{ enters } W_e^* \text{ at s}\}.$$

For each pair $\langle e, \sigma\rangle \in Z$, for each $k < |\sigma|$ and for each i such that $D_i \subseteq 2^{<k}$, let $t_{e,\sigma,k,i}$ be the stage when σ enters $W_{q(e,k,i)}$. Define $g(s)$ to be the maximum of $f(t_{e,\sigma,k,i})$ over all those $t_{e,\sigma,k,i}$ just defined, or $g(s) = f(s) + 1$ if no $t_{e,\sigma,k,i}$ were defined (ie., if Z is empty). Note that $g(s) > t_{e,\sigma,k,i} > f(s)$ since $t_{e,\sigma,k,i} > s$ and f is increasing.

We claim that if $|\sigma| \geq k$ and $\sigma \in \mathrm{PP}_{U^A,f}(W_{q(e,k,i)})$ for some i then $\sigma \in \mathrm{PP}_{U^A,g}(W_e)$. Suppose that $\sigma \in \mathrm{PP}_{U^A,f}(W_{q(e,k,i)})$ and $|\sigma| \geq k$. Since $W_{q(e,k,i)} - 2^{<k} \subseteq W_e^*$, σ enters W_e^* at some stage s, and therefore enters $W_{q(e,k,i)}$ at some $t > s$. Let u be the use of $\sigma \in U^A[s]$. If $\sigma \in U^A[t]$ with use u, then we must have $A[t] \upharpoonright u \neq A[f(t)] \upharpoonright u$ since σ is promptly permitted, and hence $A[s] \upharpoonright u \neq A[g(s)] \upharpoonright u$ since $g(s) \geq f(t)$. So $\sigma \in \mathrm{PP}_{U^A,g}(W_e)$. Otherwise, either σ is not in $U^A[t]$

or it is in $U^A[t]$ with some other use; but either way A must have changed below u between s and t and hence $\sigma \in \mathrm{PP}_{U^A,g}(W_e)$ since $g(s) > t$.

Now we can prove that g satisfies (i). Let ϵ be such that $0 < \epsilon < 1 - \text{weight } U^A$. Suppose that $U^A \subseteq W_e$. By (ii), $\mathrm{PP}_{U^A,f}(W_e) - U^A$ has weight $> \epsilon$, and so $\mathrm{PP}_{U^A,g}(W_e) - U^A$ has weight $> \epsilon$ also since $\mathrm{PP}_{U^A,f}(W_e) \subseteq \mathrm{PP}_{U^A,g}(W_e)$. Let k be such that

$$\left(\mathrm{PP}_{U^A,g}(W_e) - U^A\right) \cap 2^{<k} \tag{9}$$

has weight $> \epsilon$. For some i we have $D_i = U^A \cap 2^{<k}$. Then $U^A \subseteq W_{q(e,k,i)}$, and therefore

$$\mathrm{PP}_{U^A,f}(W_{q(e,k,i)}) - U^A \tag{10}$$

has weight $> \epsilon$. By the earlier claim, $\mathrm{PP}_{U^A,f}(W_{q(e,k,i)}) \subseteq \mathrm{PP}_{U^A,g}(W_e)$. Since the sets (9) and (10) are disjoint, $\mathrm{PP}_{U^A,g}(W_e)$ has weight $> 2\epsilon$. We may now repeat the argument with k such that

$$\text{weight } \left(\left(\mathrm{PP}_{U^A,g}(W_e) - U^A\right) \cap 2^{<k}\right) > 2\epsilon$$

to show that weight $\mathrm{PP}_{U^A,g}(W_e) > 3\epsilon$. We can repeat this argument arbitrarily many times, each time adding ϵ to weight $\mathrm{PP}_{U^A,g}(W_e)$. Hence weight $\mathrm{PP}_{U^A,g}(W_e)$ is unbounded, establishing (i). □

4. Prompt non-low-for-random sets

One usually obtains an example of a promptly simple set by analysing the standard simple set construction. That is, Post's example of a simple set (see Cooper [10], Theorem 6.2.3) turns out in fact to be promptly simple via the identity function. The same thing occurs in the case of prompt non-low-for-randomness; a standard construction of a non-low-for-random c.e. set in fact yields a set which is promptly non-low-for-random. We now give a version of the non-low-for-random construction, adapted slightly to simplify the promptness verification.[7]

Theorem 7. There is a c.e. set A which is promptly non-low-for-random. Moreover, we can have $A \geq_T \emptyset'$.

We will construct the set A, as well as a sequence of c.e. operators T_e to satisfy

$$P_e: \qquad T_e^A \subseteq W_e \Rightarrow \text{weight } \mathrm{PP}_{T_e^A,\mathrm{id}}(W_e) \geq 1$$

where id is the identity function. We ensure that weight $T_e^A \leq 2^{-e-1}$ and that the T_e^A are pairwise disjoint (as sets of strings). Hence we may set $T^A = \cup_e T_e^A$ to obtain a bounded A-c.e. set such that A is promptly non-low-for-random by Theorem 6 via T^A and the identity function. We will argue that $A \geq_T \emptyset'$ automatically.

The basic strategy for P_e will be to put a string σ of fixed length l into T_e^A and wait for $\sigma \in W_e[s]$. As soon as this occurs, we remove σ from T_e^A by enumerating into A below the use of $\sigma \in T_e^A[s]$, thus promptly permitting σ. Then we repeat with the next string of length l. After at most 2^l many repetitions, we will either have some string σ which never appears in W_e and is permanently in T_e^A, and hence $T_e^A \not\subseteq W_e$, or we will have promptly permitted all strings of length l, which have weight 1. At any stage, there is at most one string in T_e^A of length l; thus T_e^A has weight at most 2^{-l}. Let $W_e^*[s]$ be (as in equation (6)) a delayed enumeration of a subset of W_e, such that a string σ is enumerated into W_e^* at stage s iff s is the least such that $\sigma \in W_e[s] \cap T_e^A[s]$.

To simplify the verification, each time a higher priority requirement acts we will make P_e start over with a new length l. Let $j(e,s)$ be the number of times that any requirement $P_{e'}$ with $e' < e$ has acted by stage s, and let $l_e[s] = p_e^{j(e,s)+1}$, where p_e is the e'th prime. At stage s, P_e will use strings of length $l_e[s]$. This simplifies the verification by ensuring that no two requirements will ever use strings of the same length.

[7] Usually one would obtain a non-low-for-random c.e. set indirectly, for instance by using the fact that all low-for-randoms are low and hence that any non-low c.e. set is non-low-for-random. If one wished to explicitly construct a non-low-for-random c.e. set, the construction of Theorem 7 would be the standard method.

The construction consists of the P_e requirements in a finite injury setting. Initially we have A, T_e all empty. At stage 0, do nothing. At stage $s+1$, let e be the least such that $T_e^A[s] \subseteq W_e^*[s]$, or $T_e^A[s] = \emptyset$ and there is some string σ of length $l_e[s]$ not in $W_e^*[s]$.

If there is some $\sigma' \in T_e^A[s]$, then enumerate the use of $\sigma' \in T_e^A[s]$ into $A[s+1]$ (removing σ' from T_e^A). We say that P_e *acts* at $s+1$.

Let σ be the lexicographically least string of length $l_e[s]$ which is not in $W_e^*[s]$, if it exists. If σ exists, declare $\sigma \in T_e^A[s+1]$ with fresh use u. If σ does not exist, do nothing more.

End of construction.

Lemma 8. For each e, P_e acts only finitely often.

Proof. This is a standard finite injury argument. Assume inductively that the claim holds for $e' < e$, and let s_0 be the least stage such that no P_e' for $e' < e$ acts at any $s \geq s_0$. Then $l_e[s]$ is fixed after s_0. Note that if $\sigma \in T_e^A[s]$ and $\sigma' \in T_{e'}^A[s]$ for $e' > e$ then the latter has larger use than the former. Thus if P_e enumerates into A at stage s then all $T_{e'}^A[s+1]$ become empty, for $e' > e$. If some string is put into $T_{e'}^A$ after s then it will have use larger than that of any string in T_e^A. So enumerations into A by lower priority requirements will not disturb strings in T_e^A, and any strings put into T_e^A after s_0 will remain there until removed by P_e. After s_0, P_e can act at most $2^{l_e[s_0]}$ times until every string of length $l_e[s_0]$ is in V_e. □

Lemma 9. A is promptly non-low-for-random.

Proof. Set $T = \cup_e T_e$. Note that each T_e^A contains at most one string, of length $\geq e+1$, so weight $T^A < 1$. Suppose that $T^A \subseteq W_e$. Let s_0 be the least stage such that no requirement $P_{e'}$ for $e' < e$ acts after s_0. Then $l := l_e[s]$ is fixed after s_0. At s_0, P_e will put the first string σ of length l into T^A. Since $T^A \subseteq W_e$, there will be a least $s_1 > s_0$ with $\sigma \in W_e[s_1]$. Since no higher priority requirement ever acts after s_0, P_e will act at s_1 and will enumerate into A below the use of σ. Thus $\sigma \in PP(W_e)$. P_e will then put the next string of length l into $T_e^A[s_1+1]$. Since $T_e^A \subseteq W_e^*$, $W_e^*[s_0]$ contains no strings of length l, and no other requirement uses strings of length l, this will happen 2^l many times, until every string of length l is in $PP(W_e)$. But then weight $PP(W_e) \geq 1$. Hence A is promptly non-low-for-random by Theorem 6, via the operator T and the identity function. □

Lemma 10. $A \geq_T \emptyset'$.

Proof. Let f be a computable function such that

$$W_{f(n)} = \begin{cases} \{0, 00, 000, 0000, \ldots\} & \text{if } n \in \emptyset'; \\ \emptyset & \text{otherwise.} \end{cases}$$

That is, $W_{f(n)}$ enumerates $\emptyset'$, and if it finds $n \in \emptyset'$ then it enumerates strings of zeros of every length. From the proof of Lemma 8, it is clear that using an oracle for A we can find the least stage after which $P_{f(n)}$ is never injured. Let s_0 be that stage. At s_0, $P_{f(n)}$ will put a string of zeros into $T_{f(n)}^A$ with some use u. If $n \in \emptyset'$, then P_e will enumerate into A below u. Thus it suffices to find the least stage $s_1 \geq s_0$ such that $A[s_1] \upharpoonright u = A \upharpoonright u$. Then we have $n \in \emptyset'$ iff $n \in \emptyset'[s_1]$. □

The method used above can clearly be combined with other finite-injury strategies. For instance, we could make A low by using the usual lowness strategy, or we could use Sacks restraints (as used in [4]) to avoid a non-trivial upper cone of Turing or LR-degrees. In the presence of restraints, we can no longer ensure that T_e^A contains at most one string, since a higher-priority requirement might impose a restraint on P_e and prevent it from removing a string from T_e^A. However, since P_e uses longer strings each time it is injured, we can argue that T_e^A contains at most one string of each length, and thus the weight of T^A is bounded.

We sketch the construction of a low promptly non-low-for-random c.e. set. In addition to the requirements P_e as above, we also have the lowness requirements

$$N_e: \qquad \exists^\infty s\ \Phi_e^A(e)[s]\downarrow \Rightarrow\ \Phi_e^A(e)\downarrow$$

where $(\Phi_e)_{e\in\mathbb{N}}$ is a listing of all Turing functionals. We rank the requirements in the priority ordering $P_0, N_0, P_1, N_1, \dots$ in a finite injury setting. To satisfy N_e, if at stage s we see a new computation $\Phi_e^A(e)[s]\downarrow$ with use u, we restrain $A \restriction u$ to prevent that computation from being spoiled by enumerations into A by lower priority P-requirements. This might prevent the lower-priority P-requirements from removing their strings from T^A; any strings in $T^A_{e'}[s]$ for $e' > e$ at the time when N_e imposes its restraint will remain permanently in T^A as junk. However, at any time each requirement $P_{e'}$ will be working with at most one string of length $l_e[s]$, so the restraint will cause at most $\sum_{e'>e} l_{e'}[s]$ weight of junk to be captured in T^A. When N_e imposes its restraint, the counters $j(e', s)$ for $e' > e$ are incremented, so those P-requirements will start work again with longer strings. For every e' and every value of $l_{e'}[s]$ there is thus at most one string of length $l_{e'}[s] = p_{e'}^{j(e',s)+1}$ permanently in T^A, and so the weight of T^A is bounded by

$$\sum_e \sum_j 2^{p_e^{j+1}} < 1.$$

The rest of the details are a straightforward finite injury construction similar to that of Theorem 7 and are omitted.

5. Prompt non-low-for-randomness, prompt simplicity and Turing degrees

We now show that the promptly non-low-for-random Turing degrees are a subset of the promptly simple Turing degrees. In section 6 we show that the subset is proper. Although the promptly simple Turing degrees form a filter in the c.e. Turing degrees, it is not known if the promptly non-low-for-random Turing degrees form a filter. However we show that the promptly non-low-for-random degrees are closed upwards under $\leq_T$, and discuss the obstacles to establishing the remaining filter condition.

Theorem 11. Let A be a c.e. set. If A is promptly non-low-for-random then A is of promptly simple degree.

Proof. Suppose A is promptly non-low-for-random via U and f. We construct a computable function g such that

$$W_e \text{ infinite } \Rightarrow \exists x, s \,:\, x \in W_e[s] - W_e[s-1] \text{ and } A[s] \restriction x \neq A[g(s)] \restriction x.$$

That is, A promptly permits via g and hence has promptly simple degree by Theorem 3. We also construct c.e. sets V_e for $e \in \mathbb{N}$, and assume that we have a computable function q given by the Slowdown Lemma 4 such that $W_{q(e)} = V_e$ and strings enter $W_{q(e)}$ strictly later than they enter V_e.

Set $g(0) = 0$. At stage $s+1$, do the following for each $e \leq s$. Let x be the largest number which entered W_e at $s+1$, if any. If x exists, and if there is a string $\rho \in U^A[s]$ with use $\leq x$ and $\rho \notin V_e[s]$, then enumerate the oldest such ρ into $V_e[s+1]$. Let t_e be the stage when ρ appears in $W_{q(e)}$. If x or ρ do not exist then t_e is undefined. Finally, let $g(s+1)$ be the maximum of $f(t_e)$ for all those t_e defined at this stage (or $g(s+1) = g(s)$ if no t_e were defined).

Verification. Suppose that W_e is infinite. Then there will be infinitely many stages s when some x enters W_e and there is a $\rho \in U^A[s]$ with use $\leq x$ and $\rho \notin V_e[s]$. (To see this, observe that for every $\rho \in U^A$ there will be such a stage.) So we will enumerate infinitely many strings into V_e, and since we always choose the oldest string to enumerate, we will have $U^A \subseteq V_e$. But then weight $\mathrm{PP}(W_{q(e)}) = \infty$ since A is promptly non-low-for-random. In particular, there is at least one x, s and string ρ such that x enters W_e at s, $\rho \in U^A[s]$ with use $< x$, and $A[s] \restriction x \neq A[g(x)] \restriction x$. □

We show in section 6 that the converse of this does not hold.

We now show that prompt non-low-for-randomness is closed upwards under Turing reducibility. The proof is an adaptation of that of Theorem XIII.1.6 from Soare [24].

Theorem 12. If A, B are c.e. sets, $A \leq_T B$ and A is promptly non-low-for-random, then B is promptly non-low-for-random.

Proof. Suppose that $A = \Phi^B$ for a Turing functional Φ, and that A is promptly non-low-for-random via U, f. Assume without loss of generality that f is nondecreasing, and assume the convention that if $\Phi^B(n)[s]\downarrow$ with use u then $\Phi^B(z)[s]\downarrow$ for all $z < n$ with use $\leq u$. We define T, g so that B is promptly non-low-for-random via T, g. We also construct auxiliary c.e. sets V_e. Let q be a computable function given by the Slowdown Lemma 4 such that $W_{q(e)} = V_e$, and strings enter $W_{q(e)}$ strictly later than they enter V_e.

We can define T in advance: when we see a string $\rho \in U^A[s]$ with use u such that $\Phi^B[s] \upharpoonright u = A[s] \upharpoonright u$, then declare $\rho \in T^B[s]$ with use $v = \mathit{use}\, \Phi^A(u)[s]$ (if it is not already in $T^B[s]$).

Initially all V_e are empty and $g(s)$ undefined for all s. At stage 0, set $g(0) = 0$. At stage $s+1$, do the following for each $e < s+1$. Let

$$X = \left(W_e[s] \cap T^B[s]\right) - V_e[s]$$

be the strings in W_e and T^B but not in V_e at $s+1$. If X is empty, then do nothing for e at this stage. Otherwise, put each string σ from X into $V_e[s+1]$. For each $\sigma \in X$, let t_σ be the stage when σ was put into T^B with the current computation (ie, the least $t < s+1$ such that $\sigma \in T^B[t]$ with use u such that $B[t] \upharpoonright u = B[s] \upharpoonright u$), and let u_σ be the use of $\sigma \in U^A[t_\sigma]$. Let t_e be the least stage such that for all $\sigma \in X$:

- σ appeared in $W_{q(e)}$ at some $t' > s+1$ and $t_e \geq f(t')$, and
- $\Phi^B[t_e] \upharpoonright u_\sigma = A[t_e] \upharpoonright u_\sigma$.

Finally, at the end of stage $s+1$ declare $g(s+1)$ to be the maximum of those t_e defined at this stage (or $g(s+1) = g(s)$ if no t_e were defined). End of construction.

Verification. First we observe that $T^B = U^A$: certainly if $\sigma \in U^A$ then there will be a stage when $\sigma \in U^A[s]$ via a permanent computation, and $\Phi^B[s]$ correctly computes A on the use. At this stage σ will be put permanently into T^B (if not already). Further, since strings are only put into T^B at s if they are already in $U^A[s]$ and $\Phi^B[s]$ agrees with $A[s]$ on the use, if a string leaves U^A after it has been put into T^B then B must change below the use of the corresponding Φ^B computation, which will remove the string from T^B also.

Suppose now that $T^B \subseteq W_e$. Then $U^A = T^B \subseteq V_e \subseteq W_e$. Since A is promptly non-low-for-random, we must have weight $\mathrm{PP}_{U^A}(W_{q(e)}) = \infty$. We claim that $\mathrm{PP}_{U^A}(W_{q(e)}) \subseteq \mathrm{PP}_{T^B}(W_e)$, and thus $\mathrm{PP}_{T^B}(W_e)$ has infinite weight also.

Suppose $\sigma \in \mathrm{PP}_{U^A}(W_{q(e)})$. Let s_0 be the stage at which we put σ into V_e. At s_0 we have $\sigma \in W_e[s_0]$ and $\sigma \in T^B[s_0]$ with some use v. Moreover, s_0 is the first stage at which σ is in both W_e and T^B (or else we would have put σ into V_e at an earlier stage). Thus it suffices to show that $B[s_0] \upharpoonright v \neq B[g(s_0)] \upharpoonright v$.

Let t_σ and u_σ be as in the construction, and let t' be the stage when σ enters $W_{q(e)}$. We have $A[t_\sigma] \upharpoonright u_\sigma \neq A[f(t')] \upharpoonright u_\sigma$; either because $A[t'] \upharpoonright u_\sigma \neq A[t_\sigma] \upharpoonright u_\sigma$, or if $A[t'] \upharpoonright u_\sigma = A[t_\sigma] \upharpoonright u_\sigma$ then because $\sigma \in \mathrm{PP}_{U^A}(W_{q(e)})$. But

$$\begin{aligned}
\Phi^B[s_0] \upharpoonright u_\sigma &= A[t_\sigma] \upharpoonright u_\sigma \\
&\neq A[g(s_0)] \upharpoonright u_\sigma && \text{since } g(s_0) \geq f(t') \\
&= \Phi^B[g(s_0)] \upharpoonright u_\sigma && \text{by choice of } g(s_0).
\end{aligned}$$

Therefore $B[s_0] \upharpoonright v \neq B[g(s_0)] \upharpoonright v$, and $\sigma \in \mathrm{PP}_{T^B}(W_e)$. □

The equivalent of Theorem 12 for LR-reducibility instead of Turing reducibility does not hold. We describe in section 7 that there is an LR-complete c.e. set B which is not promptly non-low-for-random. In particular, B is $\geq_{LR}$ all promptly non-low-for-random sets including the Turing-complete set from Theorem 7.

To show that the promptly non-low-for-random degrees form a filter in the c.e. Turing degrees, it would suffice to show that given any two promptly non-low-for-random sets A, B there is a promptly non-low-for-random set C which is computable in both A, B. Given A and B, one would typically use double permitting below both A and B to construct the required C. In the case of the promptly simple degrees, the argument is as follows. Suppose A and B are of promptly simple degree and we want to make $C \leq_T A, B$ promptly simple. Suppose at some stage we see a number x enter a c.e. set W_e, and we would like to make C promptly permit x by changing below x. We could enumerate x into a set V_e, and see if x is promptly permitted by A. If it is not promptly permitted, then we abandon x and try again later with some other number from W_e. If A does promptly permit x, then we enumerate x into a second set V'_e and see if B promptly permits x. If not, again we abandon x. Otherwise, we have received prompt permissions from both A and B, so we can enumerate x into C, satisfying the prompt simplicity requirement for W_e. The fact that A and B are promptly simple ensures that some x will eventually receive both prompt permissions, although arbitrarily many other x may have to be discarded first.

In the case of prompt non-low-for-randomness, we would like to perform a similar construction. Suppose we are given promptly non-low-for-random c.e. sets A, B and we want to construct C, T^C and g such that $C \leq_T A, C \leq_T B$ and C is promptly non-low-for-random via T^C, g. Suppose at some stage we see a string $\sigma \in W_e$ which we would like C to promptly permit. We put σ into T^C with some use u and then attempt to get A and B-permissions to change C below u. If the A and B permissions both succeed then we can change $C \upharpoonright u$, promptly permitting σ. However, if one of the A or B permissions fail, then we cannot remove σ from T^C. It becomes junk and contributes unwanted weight to T^C. The risk is that weight T^C will become infinite. The fact that A and B are promptly non-low-for-random guarantees that the set of strings that do receive both A and B permissions has infinite weight, but makes no guarantee about the strings that do not receive both permissions. In particular we cannot ensure that weight $T^C < \infty$. It is unknown if the promptly non-low-for-random degrees form a filter in the c.e. Turing degrees.

6. Non-prompt non-low-for-randomness

We now present some c.e. Turing degrees that do not contain promptly non-low-for-random sets. One class of c.e. degrees that are not promptly non-low-for-random are the cappable degrees: by a theorem of [1], the cappable c.e. Turing degrees are exactly the non-promptly simple c.e degrees, and by Theorem 11, every non-promptly simple Turing degree is not promptly non-low-for-random. Cappable degrees are known to occur widely, for instance, in every class $\mathbf{Low_n}$ and $\mathbf{High_n}$ [18].

Certainly low-for-random degrees cannot be promptly non-low-for-random. These include both promptly simple and non-promptly simple degrees: it is easy to construct a cappable low-for-random c.e. set by adapting the usual minimal pair construction, and the standard cost function construction from [16] of a non-computable low-for-random c.e. set yields a promptly simple low-for-random set.

Hence there are c.e. Turing degrees which are promptly simple but not promptly non-low-for-random. This example of a promptly simple low-for-random is not so interesting though, as a low-for-random c.e. set does not (non-low-for-random) permit at all, let alone permit promptly. A more interesting question is whether there is a non-low-for-random c.e. set which is promptly simple but not promptly non-low-for-random. We now give a direct construction of such a set. The strategy for making a set non-promptly non-low-for-random is very similar to that for making a c.e. set cappable in the c.e. Turing degrees (ie. the minimal pair method).

Theorem 13. There is a non-low-for-random c.e. set A which is of promptly simple degree but is not promptly non-low-for-random.

We will construct the required c.e. set A. Let $\langle U_e, \phi_e \rangle_{e \in \mathbb{N}}$ be a listing of all pairs of a c.e. bounded operator U and a (possibly partial) computable function ϕ. We assume the convention that if $\phi(x)[s]\downarrow = z$ then $z < s$ and $\phi(y)[s]\downarrow$ for all $y < z$. To ensure that A is not promptly non-low-for-random, it will suffice to construct a c.e. set X_e (with a canonical enumeration $W_{q(e)}$ given by the Slowdown Lemma 4) for each pair U_e, ϕ_e such that if ϕ_e is total then $U_e^A \subseteq W_{q(e)}$ but weight $\mathrm{PP}_{U_e,\phi_e}(W_{q(e)}) < 1$. That is, we will satisfy each requirement

$$N_e: \qquad \phi_e \text{ total and } U_e^A \text{ infinite } \Rightarrow U_e^A \subseteq X_e \text{ but weight } \mathrm{PP}_{U_e,\phi_e}(W_{q(e)}) < 1.$$

To ensure that A is of promptly simple degree, it suffices by Theorem 3 to make A promptly permitting via the function $f(s) = s + 2$. Thus we have the promptness requirements

$$PS_e : \qquad W_e \text{ infinite } \Rightarrow \exists x, s : x \text{ enters } W_e \text{ at } s \text{ and } A[s] \upharpoonright x \neq A[s+2] \upharpoonright x$$

for each $e \in \mathbb{N}$.

Let V_e be a listing of all bounded c.e. sets; that is, c.e. sets such that weight $V_e < 1$. To make sure that A is not low-for-random we also meet the non-low-for-randomess requirements

$$P_e : \qquad T^A \not\subseteq V_e$$

for $e \in \mathbb{N}$, where T^A is an A-c.e. set with weight < 1 that we construct. By Theorem 2 this ensures that A is not low-for-random. The construction will take place on a tree; in fact we will uniformly construct a c.e. operator T_α for each P-node α on the tree, and we will set $T^A = \cup_\alpha T^A_\alpha$.

The strategy for meeting P_e is essentially that used in Theorem 7. A P_e-node α will place a string σ of fixed length k into T^A_α and wait for $\sigma \in V_e[s]$. When α sees $\sigma \in V_e[s]$, it removes σ from T^A_α by enumerating into A, and repeats with the next string σ of length k. Since weight $V_e < 1$, after at most 2^k many repetitions we must have some σ such that $\sigma \notin V_e$. This σ will be permanently in T^A_α, but no other string will be permanently in T^A_α. By suitable choice of k we can ensure that weight T^A_α is as small as necessary. Unlike Theorem 7, the tree framework will mean that σ will not be removed from T^A_α as soon as it appears in V_e, but only when the node α is next visited. Hence the strategy will make A non-low-for-random, without making it promptly so.

Occasionally it will be necessary to impose a restraint on A; when this happens, α may be unable to enumerate into A to remove a string from T^A_α. In this case, α must abandon its old string σ (which remains permanently in T^A). To ensure that weight $T^A < 1$, α will need to use longer strings in future. It will increment k and restart its strategy. Each time α is injured, the additional amount of junk contributed to T^A halves. In a finite injury setting, α will be able to satisfy its requirement while still contributing an arbitrarily small permanent weight to T^A.

The strategy for N_e will be as follows. We will try to build a c.e. set X such that if ϕ_e is total then $U^A_e \subseteq X$ but $\mathrm{PP}_{U^A_e,\phi_e}(X)$ has finite weight. Strictly, we should actually be concerned with weight $\mathrm{PP}(W)$ where W is a canonical version of X given by the Slowdown Lemma 4, but we overlook this technicality in the following discussion. Suppose at stage s we have some string $\tau \in U^A_e[s]$ with use u, but τ is not yet in X. Put τ into $X[s]$, and note that $\tau \in X[s] \cap U^A_e[s]$ for the first time at s. We want to ensure τ is not promptly permitted with respect to the function ϕ_e. This will happen if A changes below u before stage $\phi_e(s)$. If $A \upharpoonright u$ does change before stage $\phi_e(s)$ then τ will contribute to $\mathrm{PP}_{U_e,\phi_e}(X)$, which we want to keep small. So after we have enumerated τ into X, we will restrain $A \upharpoonright u$ until a stage t such that $\phi_e(s)[t]\downarrow$ (by convention if $\phi_e(s)[t]\downarrow = x$ then $x < t$). At stage t we may drop the restraint, since any change in $A \upharpoonright u$ after t will not contribute to $\mathrm{PP}_{U_e,\phi_e}(X)$. If $\phi_e(s)\uparrow$, then we will never drop the restraint, but nor will we ever impose any additional restraint for N_e. In this case, we have a permanent finite restraint. Otherwise, when ϕ_e is total, the restraint will be dropped infinitely often, providing infinitely many windows for the positive P-requirements to enumerate into A.

This situation is reminiscent of the construction of a minimal pair of c.e. Turing degrees. As in the minimal pair strategy, the negative requirement either drops its restraint infinitely often, or eventually imposes a single finite permanent restraint. With multiple N-requirements working together, the (potential) difficulty for the P-requirements is that the different N-requirements may not drop their restraints at the same time. This is solved exactly as in the minimal pair case by performing the construction on a tree.

For the prompt simplicity requirements, we have to promptly enumerate some number into A as soon as we see a larger number enter the c.e. set W_e. This would appear to be in direct conflict with the negative requirements, which want us to delay enumerations. However, the N_e strategy outlined above would in fact construct a c.e. set X such that $U^A_e \subseteq X$ but $\mathrm{PP}_{U^A_e,\phi_e}(X) = \emptyset$. This is stronger than we need to satisfy N_e; we don't need $\mathrm{PP}(X)$ to be empty, but merely to have small weight. We can allow some weaker priority PS requirements to ignore a higher priority restraint, enumerate into A, and promptly permit some string in X, as long as the total weight that is promptly permitted is small.

The tree will consist of nodes labelled N_e and P_e for $e \in \mathbb{N}$. The prompt simplicity reqirements PS_e do not reside on the tree. Nodes of even length $2e$ are labelled N_e, and nodes of odd length $2e+1$ are labelled P_e. N-nodes have two outcomes $\infty < f$, representing, respectively, the infinitary outcome where ϕ_e is total and U_e^A is infinite, and the finitary outcome where ϕ_e is partial or U_e^A is finite. P-nodes have a single outcome 0. The ordering $\infty < f$ induces an ordering on tree nodes as usual. We denote nodes of the tree by α, β etc. A node β has lower priority than α if β extends α or is to the right of α.

We write U_α, ϕ_α to denote U_e, ϕ_e when α is an N_e-node. Each N-node α will build a c.e. set X_α. Each P-node α has parameters k_α which is a number, and σ_α which is a string of length k_α. Initially, assign each P-node α on the tree a unique parameter k_α such that $\sum_\alpha 2^{-k_\alpha} < 2^{-2}$.

Let q be a function from N-nodes to $\mathbb{N}$ given by the Slowdown Lemma 4 such that $W_{q(\alpha)} = X_\alpha$ and strings enter $W_{q(\alpha)}$ strictly later than they enter X_α.

During the construction we will declare some nodes *injured*. When an P-node α is injured, we increment k_α and declare $\sigma_\alpha \uparrow$. We needn't do anything when an N-node is injured except to note the fact.

We will say that an N-node is *expansionary* at a stage s if it is not waiting for a computation $\phi(t)$ to halt, so it is safe to enumerate into A without promptly permitting any strings in X_α (the formal definition is given below). Note that all nodes belonging to a single N-requirement share the same pair U, ϕ and reside on the same level of the tree. Suppose that α, β are N_e-nodes with α to the left of β. If at stage s the node α is waiting for a computation $\phi(t)$ to halt, then it appears to α at that stage that ϕ is partial. Since α has stronger priority than β, β may safely adopt α's judgement and also assume at that stage that ϕ is partial. Thus β need not act while α is waiting for a computation. We can thereby co-ordinate the nodes of each N-requirement so that *at most one node on each level is imposing restraint at any time.* This simplifies calculating the cost of enumerations for PS requirements. Note that this is the same principle used in 'measure-guessing' constructions such as [8] and [7]. The following definition of expansionary stage captures this principle.

Let α be an N_e-node. Say that a stage s is *α-expansionary* if $X_\alpha[s] = \emptyset$, or

- t is the greatest β-expansionary stage $< s$ for any N_e-node $\beta \le \alpha$, and some string ρ was put into $X_\beta[t+1]$,
- ρ has appeared in each $W_{q(\gamma)}$ for all N_e-nodes $\gamma \ge \beta$ by some $t', t < t' \le s$, and t' is the least such,
- $\phi_e(t')[s]\downarrow$, and
- $U_\alpha^A[s] - X_\alpha[s] \neq \emptyset$.

The first three clauses state that α (or a higher priority β) isn't still delaying enumerations to prevent a string from being promptly permitted; the last clause states that there is a new string ready to be added to X_α.

To satisfy a promptly simple requirement PS_e, we will need to enumerate a number into A as soon as some larger number appears in W_e. Such an enumeration might cause some strings from the X_α's to be promptly permitted. We will allow PS_e to injure the lower priority N-requirements $N_{e'}$ for $e' > e$, but we need to ensure that PS_e will cause only a small weight of prompt permissions in the sets X_α belonging to higher-priority N-requirements. Let $i, x, s \in \mathbb{N}$; we define $\text{cost}(i, x, s)$ which is the weight that would be promptly permitted into the sets X_α belonging to N_i-nodes α if x were enumerated into A at stage s. Let β be the leftmost N_i-node such that s is not β-expansionary, if it exists; let ρ be the string most recently added to X_β at some $t+1 < s$, and let u be the use of $\rho \in U^A[t]$. If $x > u$ or if there is no such β then let $\text{cost}(i, x, s) = 0$. Otherwise let $\text{cost}(i, x, s) = 2^{-|\rho|}$.

To prevent the P and PS requirements from interfering with each other, we will reserve the odd numbers for satisfying PS requirements and the even numbers for P. We will assume that when a string σ is put into some T^A_α with use u it remains there until the number u is enumerated into A. In particular, σ remains in T^A_α even if numbers $< u$ enter A.

A P_e-node α *requires attention* at stage $s+1$ if $\sigma_\alpha\uparrow$, or $\sigma_\alpha\downarrow$, $\sigma_\alpha \in T^B_e[s]$ and $\sigma_\alpha \in V_e[s]$.

The construction. At even stages we will take action for N and P-requirements; at odd stages we will take action for PS requirements. At stage 0 and 1, do nothing. At stage $s+1>1$, we are given $A[s], \phi_e[s]$ etc and we define $A[s+1]$.

If $s+1$ is odd, then let e be the least such that PS_e is not yet satisfied and there exists z and an odd number $x \le z$ satisfying

- $z \in W_e[s+1] - W_e[s-1]$,
- $x \notin A[s]$, and
- $\mathrm{cost}(i,x,s) < 2^{-e-1}$ for all $i \le e$.

If there is no such e then go to the next stage. Otherwise enumerate the least such x into $A[s+1]$ and injure all nodes α of length $|\alpha| > 2e+1$ (these are all the $N_{e'}$ nodes for $e'>e$ or $P_{e'}$-nodes for $e' \ge e$). Declare PS_e to be satisfied.

If $s+1$ is even, then perform steps 1 and 2 below in order.

Step 1. Let α be shortest P-node that requires attention at stage $s+1$ and such that if β is an N-node with $\beta \subset \alpha$ then $\beta^\frown\infty \subseteq \alpha$ iff $s+1$ is β-expansionary. (Note that previously unvisited P-nodes will always require attention, so such a node always exists). Let the current approximation of the true path $\mathrm{TP}_{s+1} = \alpha$.

- If $\sigma_\alpha[s]\uparrow$, then let $\sigma_\alpha[s+1]$ be the lexicographically least string of length $k_\alpha[s]$ which is not in $V_\alpha[s]$. Put σ_α into $T^B_e[s+1]$ with large even use. (Note that such a string must exist since weight $V_\alpha < 1$.)
- Otherwise, $\sigma_\alpha\downarrow$, $\sigma_\alpha \in T^B_\alpha[s]$ with some use u and $\sigma_\alpha \in V_\alpha[s]$. Enumerate u into $A[s+1]$ to remove σ_α from $T^A_\alpha[s+1]$, and declare $\sigma_\alpha[s+1]\uparrow$.

Step 2. For each N-node $\beta \subset \mathrm{TP}_{s+1}$ such that s is β-expansionary, in order of length, do the following. Let ρ be the oldest string in $U^A_\beta[s] - X_\beta[s]$. That is, the unique $\rho \in U^A_\beta[s] - X_\beta[s]$ such that ρ was enumerated into U^A_β at some $s' \le s$ with use u, $A[s] \upharpoonright u = A[s'] \upharpoonright u$, and if $\rho' \in U^A_\beta[s]$ then ρ' was enumerated into U^A_β after s' or $\rho' \ge \rho$ (in the usual length/lexicographic order). Enumerate ρ into $X_\beta[s+1]$ and into $X_\gamma[s+1]$ for all N_e-nodes γ to the right of β. If ρ does not exist then do nothing for β. Injure all P-nodes of lower priority than TP_{s+1}.

End of construction.

Verification. Say that s is an α-stage if s is even and $\alpha \subseteq \mathrm{TP}_s$. α is *accessible* at s if s is an α-stage. Define the true path $\mathrm{TP} = \liminf_{\text{even } s} \mathrm{TP}_s$. We verify simultaneously that the true path TP is infinite, each node on TP is injured only finitely often, and that each P-node on TP requires attention only finitely often.

Lemma 14. For each n, there is a unique node α of length n such that

(i) α is accessible infinitely often, and is the leftmost such node of length n;
(ii) α is injured only finitely often;
(iii) if α is a P-node, then α requires attention only finitely often and there is a string $\sigma \in T^A_\alpha$ permanently but $\sigma \notin V_\alpha$.

Proof. Induction on the length n. The claim holds trivially for the root node. Assume inductively that β is the node of length n satisfying the claim, and that s_0 is a stage such that β is never injured or receives attention after s_0. If β is an N-node, then for every β-stage $s > s_0$ we have either $\beta^\frown\infty \subseteq \mathrm{TP}_s$ or $\beta^\frown 0 \subseteq \mathrm{TP}_s$. If β is a P-node, then it has only one child $\beta^\frown 0$, and since β never receives attention after s_0, we have $\beta^\frown 0 \subseteq \mathrm{TP}_s$ for all β-stages $s > s_0$. Hence some child of β is accessible infinitely often, and so (i) holds. Let α be the leftmost such.

The node α can only be injured when $\mathrm{TP}_s < \beta$ or when some requirement PS_e with $2e+1 < |\alpha|$ enumerates into A. The former occurs only finitely often by the induction hypothesis, and there are only finitely many PS requirements with $2e+1 < |\alpha|$, and each acts at most once. So α is injured only finitely often.

Suppose that α is a P-node. Let s_1 be the least α-stage such that α is never injured at any $s \geq s_1$. Then k_α is fixed after s_1, and α receives attention at s_1. Since weight V_α is bounded, it cannot contain all strings of length k_α. Hence there is a lexicographically least string $\sigma \notin V_\alpha$ of length k_α. After receiving attention finitely many times after s_1, α will set $\sigma_\alpha = \sigma$ and will put σ into $T^A_\alpha[s]$. After this, σ is in T^A_α permanently and α will never require attention again. □

We can now verify that the P and N requirements are satisfied by the nodes on the true path.

Lemma 15. Each requirement P_e is satisfied. Therefore A is not low-for-random.

Proof. By Lemma 14, $T^A = \cup_\alpha T^A_\alpha \not\subseteq V_e$ for all e. We just need to verify that weight $T^A < 1$. Let α be a P-node and j_α be the initial value of k_α. Since k_α is increased each time α is injured, and at most one string is left in T^A_α with each injury, we have weight $T^A_\alpha \leq \sum_n 2^{-j_\alpha - n - 1}$. By the choice of the j_α, weight $T^A \leq \sum_\alpha 2^{-j_\alpha} < 1$. □

Lemma 16. Each requirement N_e is satisfied. Therefore A is not promptly non-low-for-random.

Proof. Let α be the N_e-node on TP, and let s_0 be the least α-stage such that α is never injured at any $s \geq s_0$. If ϕ_α is partial or U^A_α is finite, then N_e is satisfied and there are only finitely many α-expansionary stages after s_0. Suppose that ϕ_α is total and U^A_α is infinite. Then there are infinitely many α-expansionary stages after s_0, and at each such stage s we put some string from $U^A_\alpha[s]$ into X_α. Since we always choose the oldest string from $U^A_\alpha[s] - X_\alpha[s]$, if $\rho \in U^A_\alpha$ permanently then eventually we will put ρ into X_α. Hence $U^A_\alpha \subseteq X_\alpha = W_{q(\alpha)}$. We argue that weight $\mathrm{PP}_{U_\alpha,\phi_\alpha}(W_{q(\alpha)}) < \infty$.

Each time α is injured, some strings may be promptly permitted into $\mathrm{PP}(W_{q(\alpha)})$. However α is injured only finitely often so this contributes only finite weight to $\mathrm{PP}(W_{q(\alpha)})$. The tree layout ensures that lower-priority P-requirements will not contribute to $\mathrm{PP}(W_{q(\alpha)})$, and after s_0 nor will higher priority P or PS requirements. So the only contributions to $\mathrm{PP}(W_{q(\alpha)})$ after s_0 can come from requirements PS_i with $i > e$. But each of these acts at most once, and contributes at most $\mathrm{cost}(e, x, s) < 2^{-i-2}$ to $\mathrm{PP}(W_{q(\alpha)})$. Thus the total contribution to $\mathrm{PP}(W_{q(\alpha)})$ after s_0 is at most $\sum_{i>e} 2^{-i-2} = 2^{-e-2}$, which establishes the claim. □

Lemma 17. Each requirement PS_e is satisfied. Hence, A is of promptly simple degree.

Proof. Let e be such that W_e is infinite. We show that eventually the cost conditions $\mathrm{cost}(i, x, s) < 2^{-e-2}$ hold for each $i < e$ and all sufficiently large x and s. Fix $i < e$, and let α be the N_i-node on the true path. Because an N-node must wait for the leftward nodes before it can have an expansionary stage, we have the following fact. Either (i) α has infinitely many expansionary stages, or (ii) there are only finitely many stages when any N_i-node is expansionary. If (ii) holds, let s_0 be a stage such that no N_i-node is expansionary after s_0. Then $\mathrm{cost}(i, x, s) = 0$ for all $x, s > s_0$. If (i) holds, let s_0 be a stage such that no string shorter than $e + 2$ is added to W_β after s_0 for any N_i-node β. Then $\mathrm{cost}(i, x, s) < 2^{-e-2}$ for all $x, s > s_0$.

Let s_1 be such that $\mathrm{cost}(i, x, s) < 2^{-e-2}$ for all $i < e$ and all $x, s > s_1$, and such that no requirement PS_j for $j < i$ acts after s_1. Let $z, s > s_1 + 1$ be such that z is enumerated into W_e at s. Then some $x \leq z$ will be enumerated into A at the first odd stage $\geq s$ and PS_e will be satisfied, if it is not already satisfied. This establishes Lemma 17 and Theorem 13. □

As noted earlier, the nodes belonging to each requirement N_i co-operate in such a way that there is at most one node on each level imposing restraint at any time. This is exactly the condition required for constructing a c.e. set that is LR-complete ($\geq_{LR} \emptyset'$) via the measure-guessing technique, used originally in [8] and further in [6, 7, 20]. We can in fact modify the above construction to make the set A be LR-complete, by replacing the P-requirements with the LR-completeness strategy exactly as in Theorem 1.1 of [7]. Since the N-strategy already satisfies the condition that at most one node on each level imposes restraint at a time, the modifications needed to make A LR-complete are straightforward and we omit the details.

7. Prompt non-low-for-randomness and LR-degrees

So far we have been investigating prompt non-low-for-randomness within the Turing degrees. We might ask whether the LR-degrees of promptly non-low-for-randoms form a nice class within the LR-degrees also. Although it is possible that the class of LR-degrees of promptly non-low-for-randoms might have some nice properties, the LR-degrees do not seem as natural a setting for the study of prompt non-low-for-randomness as the Turing degrees. Theorem 12 shows that prompt non-low-for-randomness is Turing degree-invariant: if A is promptly non-low-for-random and $A \equiv_T B$ for c.e. sets A, B, then B is also promptly non-low-for-random. This does not hold for LR-degrees however; an LR-degree can contain both prompt and non-prompt non-low-for-randoms. By Theorem 7 there is a Turing complete, and therefore LR-complete, promptly non-low-for-random c.e. set. By the result from [6] that there is a cappable LR-complete c.e. Turing degree (or the stronger result of [7] of a non-cuppable LR-complete c.e. Turing degree), and the implications

$$\text{cappable} \;\Rightarrow\; \text{not promptly simple} \;\Rightarrow\; \text{not promptly non-low-for-random}$$

there is an LR-complete c.e. set that is not promptly non-low-for-random. Hence the LR-degree of $\emptyset'$ contains promptly and non-promptly non-low-for-random c.e. sets. Moreover, by the remarks at the end of section 6, the LR-degree of $\emptyset'$ contains a promptly simple but not promptly non-low-for-random c.e. set. Thus the LR-degree of $\emptyset'$ contains c.e. sets of all possibilities: promptly non-low-for-random, promptly simple but not promptly non-low-for-random, and not promptly simple.

It is not known whether there is a nonzero c.e. LR-degree that contains no promptly non-low-for-random c.e. sets, or whether there is a c.e. LR-degree in which all the c.e. sets are promptly non-low-for-random. One possible approach to the latter is via jump inversion. If $\mathbf{a}$ is a Turing degree which is $\geq_T \emptyset'$ and c.e. in $\emptyset'$, the *atomic jump class* of $\mathbf{a}$ is the set of those c.e. Turing degrees $\mathbf{b}$ such that $\mathbf{b}' = \mathbf{a}$. The Sacks jump inversion theorem (see Soare [24] VIII.3.1) states that every atomic jump class is nonempty. Cooper [9] showed that there is an atomic jump class that contains only noncappable c.e. Turing degrees: there is a degree $\mathbf{a}$ c.e. in and $\geq_T \emptyset'$ such that if a c.e. degree $\mathbf{b}$ has $\mathbf{b}' = \mathbf{a}$ then $\mathbf{b}$ is noncappable. Since $B \equiv_{LR} C$ implies $B' \equiv_{tt} C'$, the LR-degree of such a $\mathbf{b}$ contains only noncappable, and hence promptly simple, c.e. sets. If Cooper's theorem could be strengthened to produce an atomic jump class of promptly non-low-for-randoms, ie $\mathbf{b}' = \mathbf{a}$ implies $\mathbf{b}$ is promptly non-low-for-random, then this would give a c.e. LR-degree in which all the c.e. sets are promptly non-low-for-random.

A similar jump inversion argument cannot produce a nontrivial c.e. LR-degree without promptly non-low-for-randoms however, because promptly non-low-for-randoms occur in every atomic jump class. Robinson [23] proved that the Sacks jump inversion theorem can be done above any low c.e. Turing degree. That is, given a low c.e. Turing degree $\mathbf{d}$ and a Turing degree $\mathbf{a}$ c.e. in and $\geq_T \emptyset'$, there is a c.e. Turing degree $\mathbf{b} \geq_T \mathbf{d}$ with $\mathbf{b}' = \mathbf{a}$. By the comments after Theorem 7, there is a low promptly non-low-for-random c.e. set D. By Robinson's theorem, every atomic jump class has a representative $\geq_T D$, and hence promptly non-low-for-random by Theorem 12. (The same argument works for showing that promptly simples occur in every atomic jump class.)

References

[1] K. Ambos-Spies, C. G. Jockusch Jr., R. A. Shore and R. I. Soare. An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees. *Transactions of the American Mathematical Society* **281** (1984), 109–128.

[2] G. Barmpalias. Elementary differences between the degrees of unsolvability and the degrees of compressibility. *Annals of Pure and Applied Logic* **161** (2010), 923–934.

[3] G. Barmpalias. Relative randomness and cardinality. *Notre Dame Journal of Formal Logic* **51** (2010), 195–205.

[4] G. Barmpalias, A. E. M. Lewis and M. Soskova. Randomness, lowness and degrees. *Journal of Symbolic Logic* **73** (2008), 559–577.

[5] G. Barmpalias, A. E. M. Lewis and F. Stephan. Π^0_1 classes, LR degrees and Turing degrees. *Annals of Pure and Applied Logic* **156** (2008), 21–38.

[6] G. Barmpalias and A. Montalbán. A cappable almost everywhere dominating computably enumerable degree. *Electronic Notes in Theoretical Computer Science* **167** (2007).
[7] G. Barmpalias and A. Morphett. Non-cupping, measure and computably enumerable splittings. *Mathematical Structures in Computer Science* **19** (2009), 25–43.
[8] P. Cholak, N. Greenberg and J. S. Miller. Uniform almost everywhere domination. *Journal of Symbolic Logic* **71** (2006), 1057–1072.
[9] S. B. Cooper. A jump class of noncappable degrees. *Journal of Symbolic Logic* **54** (1989), 324–353.
[10] S. B. Cooper. *Computability Theory*. Chapman & Hall/CRC Press, Boca Raton, New York, London, Washington D.C., 2004.
[11] R. G. Downey and D. R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, to appear.
[12] R. M. Friedberg. The fine structure of degrees of unsolvability of recursively enumerable sets. *Summaries of Cornell University Summer Institute for Symbolic Logic*, Communications Research Division, Inst. for Def. Anal., Princeton, 1957, 404–406.
[13] D. R. Hirschfeldt, A. Nies and F. Stephan. Using random sets as oracles. *Journal of the London Mathematical Society* **75** (2007), 610–622.
[14] B. Kjos-Hanssen. Low for random reals and positive-measure domination. *Proceedings of the American Mathematical Society* **135** (2007), 3703–3709.
[15] B. Kjos-Hanssen, J. S. Miller and D. R. Solomon. Lowness notions, measure and domination. To appear.
[16] A. Kučera and S. A. Terwijn. Lowness for the class of random sets. *Journal of Symbolic Logic* **64** (1999), 1396–1402.
[17] W. Maass. Recursively enumerable generic sets. *Journal of Symbolic Logic* **47** (1982), 809–823.
[18] W. Maass, R. A. Shore and M. Stob. Splitting properties and jump classes. *Israel Journal of Mathematics* **39** (1981), 210–224.
[19] P. Martin-Löf. The definition of random sequences. *Information and Control* **9** (1966), 602–619.
[20] A. Morphett. *Degrees of Computability and Randomness*. PhD. thesis, University of Leeds, 2009.
[21] A. Nies. *Computability and Randomness*. Oxford University Press, Oxford, 2009.
[22] A. Nies. Lowness properties and randomness. *Advances in Mathematics* **197** (2005), 274–305.
[23] R. W. Robinson. Jump restricted interpolation in the recursively enumerable degrees. *The Annals of Mathematics* **93** (1971), 586–596.
[24] R. I. Soare. *Recursively Enumerable Sets and Degrees*. Springer-Verlag, Berlin, London, 1987.
[25] R. Solovay. Draft of a paper (or series of papers) on Chaitin's work. Unpublished handwritten notes, 215 pages, 1975.
[26] C. E. M. Yates. Three theorems on the degree of recursively enumerable sets. *Duke Mathematical Journal* **32** (1965), 461–468.

Address

Anthony Morphett. School of Mathematics, University of Leeds, LS2 9JT, U.K.; *Email:* awmorp@gmail.com.