LOW UPPER BOUNDS IN THE LR DEGREES

DAVID DIAMONDSTONE

Abstract. We say that $A \leq_{LR} B$ if every B-random real is A-random—in other words, if B has at least as much derandomization power as A. The LR reducibility is a natural weak reducibility in the context of randomness, and generalizes lowness for randomness. We study the existence and properties of upper bounds in the context of the LR degrees. In particular, we show that given two (or even finitely many) low sets, there is a low c.e. set which lies LR above both. This is very different from the situation in the Turing degrees, where the Sacks splitting theorem shows that two low sets can join to $0'$. The techniques used provide new ways of working in the LR degrees.

1. Introduction

One of the main themes in the theory of algorithmic randomness has been studying the relationship between the randomness of a set of natural numbers, and its computational power. One surprising feature that has emerged is that this relationship is not monotone, but rather follows something like a bell curve. At one end, we have sets which are far from being random, which exhibit little or no computational power. These include the computable sets, and the trivial sets, which have minimal initial segment complexity. At the other end, we have sets which are extremely random, such as the 2-randoms, n-randoms, and Π^1_1-Martin-Löf-randoms. These sets also exhibit little computational power. In the middle, however, where we have 1-randoms, we find sets that can have extraordinary computational power. In fact, a theorem due independently to Kučera [9] and Gàcs [6] says that every Turing degree above the halting problem contains a set which is 1-random. This theorem tells us that 1-random sets can have arbitrarily high computational power.

In this paper, we examine the left end of this spectrum, those sets which are far from random: the K-trivials. This remarkable class of sets seems to come up again and again. Many natural lowness notions have been introduced in the theory of randomness, and, surprisingly, many of them coincide. First there are the K-trivials, those sets which have minimal initial segment Kolmogorov complexity.

Definition 1. A is K-trivial if there is some constant c such that

$$(\forall n) K(A \upharpoonright n) \leq K(n) + c.$$

Then there are sets which are low for randomness (sets A where $\mathcal{MLR}^A = \mathcal{MLR}$) or low for K (sets A where $K^A - K$ is bounded). The list goes on: we have sets which are low for weak 2-randomness, sets which are bases for Martin-Löf randomness, and others. The fact that all of these randomness-theoretic lowness properties exactly coincide is one of the more remarkable results in the theory of randomness. These equivalence theorems were not easy, and were proved in many separate parts by researchers including Denis Hirschfeldt, André Nies, and Frank

Stephan (see [11], [7], and [4]). Much current research in randomness tries to better understand this class.

2. Background

2.1. **weak reducibilities.** One way of studying a lowness notion is by studying an associated weak reducibility. A preorder $\leq_*$ on the power set of the natural numbers is called a *weak reducibility* if it is weaker than Turing reducibility, i.e. $A \leq_T B$ implies $A \leq_* B$. Weak reducibilities are often associated to lowness notions by relativizing (or partially relativizing) the lowness notion. Every weak reducibility has an associated degree structure, where the degrees are equivalence classes under bi-reducibility, and are partially ordered by the reducibility.

André Nies introduced *LR-reducibility* as another way of understanding lowness for randomness [11]. (The definition is a partial relativization of being low for random, hence the name LR.)

Definition 2. The *LR reducibility* is the weak reducibility defined by $A \leq_{LR} B$ if $\mathcal{MLR}^B \subseteq \mathcal{MLR}^A$.

The bottom degree $\{A \mid A \leq_{LR} 0\}$ is the set of low-for-random reals, i.e. the K-trivials. This is only a *partial* relativization, because $A \leq_{LR} B$ is different from A low-for-random relative to B. Actually A is low-for-random relative to B if and only if $A \oplus B \leq_{LR} B$. The reason that a partial relativization rather than a full relativization is used is that a full relativization does not give a transitive relation.

Kjos-Hanssen, Miller, and Solomon proved in [8] that this structure is equivalent to another structure, the LK degrees, which are defined by giving a partial relativization of being low for K: $A \leq_{LK} B$ if $K^A \leq K^B + O(1)$. This generalizes the fact that being low for K is the same as being low for randomness, making this a relatively robust structure. However, it is not a true reducibility, because there are no reduction procedures. In fact, Barmpalias, Lewis, and Soskova showed in [2] that $0'$ has uncountably many predecessors under LR. This was improved by Joseph Miller in [10], where he showed that the same is true for each set which is not low for Ω.

This paper introduces a new technique for dealing with the LR reducibility, and applies that technique to exhibit a surprising divergence between the LR degrees and the Turing degrees:

Theorem 1. *Given low sets A, B, there is a low c.e. set C such that $A, B \leq_{LR} C$.*

This is in stark contrast with Sacks' splitting theorem, which shows that $0'$ is the least upper bound in the Turing degrees of two low (c.e.) sets.

2.2. **Properties of $\leq_{LR}$.** A good way to think about the LR reducibility is by using the contrapositive definition: $A \leq_{LR} B$ if and only if $\overline{\mathcal{MLR}}^A \subseteq \overline{\mathcal{MLR}}^B$. In other words, $A \leq_{LR} B$ if B has at least as much power as A to de-randomize, i.e. to compute Martin-Löf tests covering any given real. So it does make sense to regard this as a reducibility, but instead of a reduction from B to A, instead you have something like a reduction from a Martin-Löf test relative to B to a Martin-Löf rest relative to A. The LR reducibility is a covering notion: $A \leq_{LR} B$ if the universal ML-test relative to B covers the universal ML-test relative to A.

Because LR is a covering notion dealing with sets of bounded measure, it relates to the reverse mathematics of measure theory. It has connections with other definitions which have arisen in that context (see [12]):

Definition 3. B is *almost-everywhere dominating* if, for almost all reals X (w.r.t. Lebesgue measure), every X-computable function is dominated by some B-computable function. B is *uniformly almost-everywhere dominating* if there is a single B-computable function f such that for almost all X, every X-computable function is dominated by f.

Bjørn Kjos-Hanssen, Joseph Miller, and Reed Solomon showed in [8] that the following are equivalent for a real B:

- B is almost-everywhere dominating
- B is uniformly almost-everywhere dominating
- $0' \leq_{LR} B$.

In addition to showing the connection with reverse mathematics, this theorem has a useful corollary: if $0' \leq_{LR} B$, then B is high. This is because of Martin's high domination theorem: a uniformly almost-everywhere dominating set can compute a dominant function, so by Martin's theorem, it must be high.

The LR reducibility will inevitably be compared to Turing reducibility, which after all is the prototypical example of a reducibility for computability theorists. However, $\leq_{LR}$ is very different from $\leq_T$. One of the first examples of this is that the usual Turing join, $\oplus$, is not a join in the LR degrees.

Observation. There are reals A, B such that $A \leq B$, but $A \oplus B \not\leq_{LR} B$.

Proof. Let A be a promptly simple K-trivial (which exists by a cost function construction; see [5]). Let B be a low set such that $A \oplus B \equiv_T 0'$, by the well known result of Ambos-Spies, Juckusch, Shore, and Soare [1]. Then $A \leq_{LR} 0 \leq_{LR} B$. However, since B is low, it is not uniformly almost-everywhere dominating, so $A \oplus B \not\leq_{LR} B$. □

It is not known whether there is a join in the LR degrees; it may be that there are two LR degrees with no least upper bound. It is also not known whether greatest lower bounds must always exist. Nevertheless, one can easily see that $A \oplus B$ is an upper bound for A and B, even if it is not a least upper bound. The main result of this paper shows that this upper bound may not even be close to being a least upper bound. By Sacks' splitting theorem, there are two low sets A and B with $A \oplus B \equiv_T 0'$, but by the main result of this paper, there is a low c.e. set C which is an LR-upper-bound for A and B.

One important technical tool for dealing with $\leq_{LR}$ is the following theorem, due to Bjørn Kjos-Hanssen [3], which gives a characterization of $A \leq_{LR} B$.

Theorem 2 (Kjos-Hanssen [3]). *The following are equivalent for reals A, B:*

- $A \leq_{LR} B$
- *every $\Sigma^0_1(A)$ class of measure less than 1 is contained in some $\Sigma^0_1(B)$ class of measure less than 1*
- *some member of a universal Martin-Löf test relative to A is contained in a $\Sigma^0_1(B)$ class of measure less than 1.*

2.3. **Notation and conventions.** Given a binary strings σ and a binary string (or sequence) τ, write $\sigma \subset \tau$ if τ extends σ. Given a binary string σ, we write $[[\sigma]]$ for the basic open set determined by σ, i.e.

$$[[\sigma]] = \{X \in 2^\omega \mid \sigma \subset X\}.$$

Given a set of binary strings U, we write $[[U]]$ for the open set determined by U, i.e.

$$[[U]] = \bigcup_{\sigma \in U} [[\sigma]].$$

Let σ_n be the nth string in the standard length-lexicographic ordering λ, 0, 1, 00, 01, 10, 11, ... of binary strings. Let

$$D_y = \{\sigma_n \mid \text{a '1' appears at the } n\text{th position in the binary representation of } y\}.$$

We will also write this as $D[y]$, when y is given by a complicated expression, and adopt the convention that D_y is the empty set when y is undefined (i.e. when it is given by a divergent computation). If U is a set of binary strings, $w(U) = \sum_{\sigma \in U} 2^{-|\sigma|}$ is the weight of U. Thus $w(U) \geq \mu([[U]])$, with the two being equal if and only if U is prefix-free. In order to work with c.e. sets of strings rather than Σ^0_1 classes of reals as often as possible, we will recast the definition of a universal oracle ML-test as follows:

Definition 4. A *universal oracle Martin-Löf test* is a uniform sequence of c.e. operators $U_n : 2^\omega \to 2^{2^{<\omega}}$ such that for all $X \in 2^\omega$ and for all n we have

- The measure of $[[U^X_n]]$ is at most 2^{-n}, and
- for all Martin-Löf tests relative X, some member of that test is a subset of $[[U^X_n]]$.

Remark. The universal oracle ML-tests used in this paper will meet the slightly stronger condition that $w(U^X_n) < 2^{-n}$. In other words, the *weight* is bounded by 2^{-n}, rather just the measure being bounded. Universal oracle ML-tests exist which meet this condition: simply take the usual existence proof, and force the sets to remain prefix-free, which will ensure that the weight equals the measure.)

3. Exhibiting LR upper bounds

In this section, we outline the strategy behind the proof of the main theorem:

Theorem 1. *Given two low sets A, B, there is a low c.e. set C such that $A, B \leq_{LR} C$.*

Consider the problem, given a Δ^0_2 set X, of producing a Δ^0_2 set Y, with $Y \geq_{LR} X$, possibly with some additional properties. Since X is Δ^0_2, we have an approximation to X. By taking a member of the universal oracle Martin-Löf test and setting the oracle to X, we obtain a $\Sigma^0_1(X)$ set U^X. Furthermore, the Δ^0_2 approximation to X naturally gives a Σ^0_2 approximation to U^X. To ensure that $Y \geq_{LR} X$, we will build a $\Sigma^0_1(Y)$ set V^Y with $w(V^Y) < 1$ covering U^X. If we succeed, then by the theorem of Kjos-Hanssen, we will have $X \leq_{LR} Y$.

The most obvious thing to do when σ enters U^X is to immediately put σ in V^Y. However, if the approximation to X changes and σ leaves U^X, it is less obvious what to do. There are two choices, each having advantages and disadvantages. The first choice is to do nothing, which leaves σ in V^Y. This has the advantage of preserving Y, so is compatible with other strategies which attempt to restrain

Y, for example to make Y low. However, it keeps superfluous weight in V^Y. If this strategy is used too often, and the set X is not itself K-trivial, the result will be $w(V^Y) = 1$, so V^Y will not be a witness to $X \leq_{LR} Y$, as desired. The other option is to change Y, which allows us to remove σ from V^Y. The advantage of this option is that it keeps superfluous weight out of V^Y, making it easy to ensure $w(V^Y) < 1$. The disadvantage is that changing Y frequently can result in a set which is computationally powerful. If this is done too frequently, we will have $Y \geq_T X$.

Even though the usual definition of Martin-Löf test refers to measure, as does the theorem of Kjos-Hanssen we are making use of, this strategy instead deals with weight. As discussed in the remark at the end of the introduction to this paper, the definition of an Martin-Löf test may be recast in this matter, and this will be equivalent for the purposes of defining randomness. Similarly, the theorem of Kjos-Hanssen is still true when measure is replaced with weight, and the proof is identical. So we do not lose anything by talking about weight instead of measure. The main advantage is that nonempty sets can have measure 0, but nonempty sets have positive weight. Thus if one can ensure that $w(U \setminus V) = 0$, then $U \subseteq V$.

3.1. **Our strategy: divide and conquer.** We are now ready to adapt the general strategy for producing an LR-upper-bound Y for a single set X outlined above to our specific case, where we want to bound two separate low sets A and B, and must also make the set $C \geq_{LR} A, B$ be low. In order to ensure that C is low, we will satisfy the usual lowness requirements

$$\mathcal{N}_e : \qquad (\exists^\infty s)\Phi_e^C(e)\downarrow [s] \quad \implies \quad \Phi_e^C(e)\downarrow .$$

We will use the standard strategy to satisfy these negative requirements: whenever we see a computation $\Phi_e^C(e)$ converge with use u, we restrain $C\upharpoonright u$ and prevent it from changing unless a higher priority positive requirement takes precedence.

The positive requirements, however, need a new idea. We have two ensure two things: that $U^A \subseteq V^C$, and that $U^B \subseteq V^C$, where V^C is a $\Sigma_1^0(C)$ set we build during the construction, with $w(V^C) < 1$. In order to deal with this requirement at the same time as we deal with the negative requirements, we will split it up into an infinite collection of more manageable requirements, so that satisfying all of them will mean that this one requirement is satisfied. We do this by replacing the requirement $U^X \subseteq V^C$ with a requirement that this is "almost" true: true up to a set of small weight. So the positive requirements are as follows:

$$\mathcal{P}_{2e} : \quad w(U^A \setminus V^C) \leq 2^{-e-4}$$
$$\mathcal{P}_{2e+1} : \quad w(U^B \setminus V^C) \leq 2^{-e-4}$$

The exponent $-e-4$ will be needed during the verification to show that $w(V^C) < 1$. We will also fix a somewhat unusual priority list of requirements: in order of decreasing priority, we will have $\mathcal{N}_0, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{N}_1, \mathcal{P}_4, \mathcal{P}_5, \mathcal{P}_6, \mathcal{P}_7, \mathcal{N}_2, \ldots$, interposing four positive requirements between every pair of negative requirements. Again, this will be needed during the verification to show that $w(V^C) < 1$.

This choice of positive requirements already suggests the beginnings of a strategy. Since A and B are low, there are Δ_2^0 approximations to A' and B'. The strategy for the positive requirement $\mathcal{P}_{2e}$ is to use low guessing to find a finite set D with weight $w(D) = 2^{-e-4}$ that the A' approximation believes will remain in U^A, disjoint from sets held by higher priority positive requirements for A, provided such a set exists.

(The strategy for $\mathcal{P}_{2e+1}$ is the same, with B replacing A.) When D is found, it is enumerated into V^C. If the D guess later appears to incorrect, C is changed to remove D from V^C, unless changing C violates a higher priority restraint.

This strategy informs us how to make the difficult decision between changing C, to keep the measure of V^C small, or preserving C, to make C low. Whenever some string σ was enumerated into V^C, it was done so for a particular requirement $\mathcal{P}_n$. If σ later leaves the set U^A which caused us to add σ to V^C, we would like to change C to remove σ from V^C, but this might violate the restraint for $\mathcal{N}_e$. We can then choose to change C, which allows us to satisfy $\mathcal{P}_n$ without enlarging V^C, but injured $\mathcal{N}_e$. Or we can choose to preserve C, which forces us to enlarge V^C to satisfy $\mathcal{P}_n$, but avoids injuring $\mathcal{N}_e$. Which one we choose depends on which requirement comes first in the priority list. Lowness ensures that our guess at the set D will eventually be correct, which means that positive requirements eventually become satisfied. This in turns means that the positive requirements each act only finitely often, so the negative requirements eventually become satisfied. Finally, some careful bookkeeping will show that each string that ends up in V^C without being in either U^A or U^B can be tied to a specific negative requirement, in such a way that no negative requirement can hold strings totaling too much weight. This will show that $w(V^C) < 1$, finishing the proof.

4. Low upper bounds

In this section we will give the full proof of the main theorem.

Theorem 1. *If A, B are low, then there is a low c.e. set C such that $A, B \leq_{LR} C$.*

Proof. Let A, B be low, and let U^X be the 3rd member of a universal oracle Martin-Löf test (so for any X we have $w(U^X) \leq \frac{1}{8}$). We introduce the following requirements:

$$\mathcal{N}_e: \quad (\exists^\infty s)\Phi_{e,s}^{C_s}(e)\downarrow \implies \Phi_e^C(e)\downarrow$$
$$\mathcal{P}_{2e}: \quad w(U^A \setminus V^C) \leq 2^{-e-4}$$
$$\mathcal{P}_{2e+1}: \quad w(U^B \setminus V^C) \leq 2^{-e-4}$$

Fix the priority list $\mathcal{N}_0, \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{N}_1, \mathcal{P}_4, \mathcal{P}_5, \mathcal{P}_6, \mathcal{P}_7, \mathcal{N}_2, \ldots$ which inserts four positive requirements between each pair of negative requirements. If $\mathcal{R}$ appears before $\mathcal{R}'$ in this list, we write $\mathcal{R} < \mathcal{R}'$.

To satisfy the positive requirement $\mathcal{P}_e$, we introduce an auxiliary functional Ψ_e^X, defined during the course of the construction. In order to stabilize these functionals with oracle A or B, we make use of the fact that A and B are low. Let $A'(x) = \lim_s f_A(x,s)$ with f computable, and likewise for B. Whenever a computation $\Psi_e^{X_s}(n)\downarrow = y$, we "confirm" the computation by testing whether $f_X(i,s) = 1$, where i is some index such that $\Phi_i^X(i) = \Psi_e^X(n)$. If $f_X(i,s) = 1$ we mark the computation as confirmed, and only then take actions reliant upon this computation being valid.

Construction. Stage 0: Start the construction with $C = \emptyset$ and $\mathrm{graph}\Psi_i = \emptyset$ for each i.

Stage $s+1$:

(1) For each $i \leq s$, let n_i be the number of times requirement $\mathcal{P}_i$ has been injured. Let $e \leq s$ be minimal such that $\Psi_e^{X_s}(n_e)$ undefined (throughout the construction, X is A if e is odd, and is B if e is even) and there is a finite set of strings $D \subseteq U^X$, disjoint from any sets held by higher priority

positive requirements for X, with $w(D) = 2^{-\lfloor e/2 \rfloor -4}$, if such e exists. Let y be the least index of such a set D, and let u be the maximum use of a computation $\sigma \in U^X$ where $\sigma \in D_y$. Enumerate the triple $(n_e, X_s \upharpoonright u, y)$ into the graph of Ψ_e.

(2) Let $e \leq s$ be minimal such that $\Psi_e^{X_s}(n_e)\downarrow$, this computation has not been confirmed, and $f_X(i,s) = 1$, where i is some index such that $\Phi_i^X(i) = \Psi_e^X(n_e)$, if such e exists. Mark the computation as confirmed. Declare all strings in $D[\Psi_e^{X_s}(n_e)]$ 'held' by $\mathcal{P}_e$, and all lower priority positive requirements injured by $\mathcal{P}_e$. Enumerate $D_y \setminus V^C$ into V^C with use $C\upharpoonright s$.

(3) For each $e \leq s$, if some string σ held by $\mathcal{P}_e$ is in $U^{X_{t-1}} \setminus U^{X_t}$, any string held by $\mathcal{P}_e$ is no longer held by $\mathcal{P}_e$, nor is it held by any $\mathcal{N}_i > \mathcal{P}_e$. We say $\mathcal{P}_e$ injures all lower priority negative requirements.

(4) For $i \leq s$, if $\Phi_{i,s}^{C_s}(i)$ becomes defined, declare every string $\sigma \in V^C$ with use $u \leq \phi_{i,s}^{C_s}(i)$ currently held by some requirement $\mathcal{R} > \mathcal{N}_i$ to be held by the requirement $\mathcal{N}_i$. If $\mathcal{R}$ was a positive requirement, σ is still (concurrently) held by $\mathcal{R}$; if $\mathcal{R}$ was a negative requirement, then σ is no longer held by $\mathcal{R}$. Thus any string is held by at most one positive requirement and at most one negative requirement, but may be concurrently held by one of each.

(5) Let u be minimal such that there is some $\tau \in V_s^{C_s\upharpoonright u}$ which is not held by any requirement (positive or negative), if such u exists. Enumerate $u-1$ into C. This has the result of removing a set of strings D from V^C comprising those strings which were in $V_s^{C_s}$ with use at least u. For each string $\sigma \in D$ still held by some positive requirement, enumerate σ into V^C with use s. For each string in D held by a negative requirement, it is no longer held by that requirement.

This ends the construction.

Verification.

Lemma 3. *All positive and negative requirements are injured at most finitely often.*

Proof. We show by induction that the positive requirement $\mathcal{P}_e$ only injures lower priority requirements finitely often. Assume all higher priority requirements only injure $\mathcal{P}_e$ finitely often. The requirement $\mathcal{P}_e$ only injures lower priority positive requirements when $\Psi_e^X(n_e)$ becomes confirmed, which means $\Psi_e^X(n_e) \downarrow [s]$ and $f_X(i,s) = 1$, where i is some index such that $\Phi_i^X(i) = \Psi_e^X(n_e)$. If this happens infinitely often, then by the definition of f_X, we must have $\Psi_e^X(n_e)\downarrow$, so at some stage s, the apparent computation must be permanent. After such a stage, $\mathcal{P}_e$ will no longer injure lower priority positive requirements.

Similarly, the requirement $\mathcal{P}_e$ only injures lower priority negative requirements when some string held by $\mathcal{P}_e$ (representing a confirmed computation) leaves U^X. This can only happen if X changes below the use of $\Psi_e^X(n_e)$, meaning that a confirmed computation becomes divergent. If this happens infinitely often, we have $f_X(i,s) = 1$ infinitely often but $\Psi_e^X(n_e)\uparrow$, which contradicts the definition of f_X. Thus $\mathcal{P}_e$ can injure lower prior negative requirements only finitely often. □

Corollary 4. *If $\Phi_e^C(e)$ becomes defined at stage s with use u, and $\mathcal{N}_e$ is not subsequently injured, then $C\upharpoonright u = C_s\upharpoonright u$.*

Proof. We make the usual assumption that the stage bounds the use of all computations, so we must have $s > u$. An element $x < u$ is only enumerated into C when some string $\tau \in V^C$ with use $x+1$ is removed from V^C, which can only happen if τ is not held by $\mathcal{N}_e$. If τ was enumerated into V^C at some stage after stage s, then the use is at least $s > u$, so τ must have already been present in V^C at stage s. Thus, since the use $x+1$ is at most u, the string τ would have been held by $\mathcal{N}_e$ at stage s. If $\mathcal{N}_e$ is not subsequently injured, then τ will always be held by $\mathcal{N}_e$ or some higher priority negative requirement (which cannot be injured without $\mathcal{N}_e$ also being injured), and so will not be removed from V^C. Thus x will not be enumerated into C. The same holds for all $x < u$, so $C\restriction u = C_s\restriction u$. □

Corollary 5. *C is low.*

Lemma 6. *$U^A = \bigcup_e D[\Psi^A_{2e}(n_{2e})]$, where n_{2e} is the number of times requirement $\mathcal{P}_{2e}$ is injured during the construction.*

Proof. Let e_k be the position of the kth one in the binary expansion of $w(U^A)$, minus 4. Note that if $e < e_1$, then eventually $w(U^A) < 2^{-e-4}$, so $\Psi^A_{2e}(n_{2e})\uparrow$. On the other hand, $w(U^A) \geq 2^{-e_1-4}$, so eventually $\Psi^A_{2e_1}(n_{2e_1})\downarrow = y$, with $D_y \subseteq U^A$. The same holds with $U^A \setminus \bigcup_{k=1}^N D[\Psi^A_{2e_k}(n_{2e_k})]$ in place of U^A and e_{N+1} in place of e_1. So by induction on k, for all k we will have $\Psi^A_{2e_k}(n_{2e_k})\downarrow = y_k$ where

$$D_{y_k} \subseteq U^A \setminus \bigcup_{e=1}^{e_k-1} D[\Psi^A_{2e}(n_{2e})].$$

Furthermore, the sets D_{y_k} will be pairwise disjoint, and we will have $\Psi^A_{2e}(n_{2e})\uparrow$ for each e which is not equal to any of the e_k, i.e. when e corresponds to the position of a 0 in the binary expansion of $w(U^A)$.

From the above, we see that $\bigcup_e D[\Psi^A_{2e}(n_{2e})] \subseteq U^A$. To show the reverse inclusion holds, first note that $w(D[\Psi^A_{2e}(n_{2e})])$ is either 0 or 2^{-e-4}, depending on whether e is equal to one of the e_k. Since these sets are disjoint for different e, we must have

$$w\left(\bigcup_e D[\Psi^A_{2e}(n_{2e})]\right) = w(U^A),$$

since both have the same binary expansion. Since $\bigcup_e D[\Psi^A_{2e}(n_{2e})]$ is a subset of U^A and has the same weight, the two must be equal. □

Lemma 7. *$U^A \subseteq V^C$.*

Proof. Since each requirement is injured finitely often, there is some first stage s by which time requirement $\mathcal{P}_e$ has been injured as many times as it will ever be injured, and $\Psi^{A_s}_{2e,s}(n_{2e}) \downarrow = \Psi^A_{2e}(n_{2e})$. At this stage, all of the set $D[\Psi^A_{2e}(n_{2e})]$ is enumerated into V^C, and never removed. Since $U^A = \bigcup_e D[\Psi^A_{2e}(n_{2e})]$, we have $U^A \subseteq V^C$. □

Note that analogues of the above lemmas hold with B in place of A, by the same proofs.

Lemma 8. *At any stage t, the negative requirement $\mathcal{N}_i$ can hold strings totaling weight at most $3/2^{2i+3}$.*

Proof. We argue by induction on the stage t. Certainly at stage 0, no negative requirement holds any string, so the claim holds at stage 0. Assume the claim holds for all stages prior to some stage $t > 0$. Then at stage t, the total weight of strings held by $\mathcal{N}_i$ can only increase when $\Phi_i^C(i)$ becomes defined, in which case $\mathcal{N}_i$ only holds strings previously held by some $\mathcal{R} > \mathcal{N}_i$. This comprises the strings held by some $\mathcal{P}_e$ with $e > 4i$ (weight at most $2^{-\lfloor e/2\rfloor -4}$ for $\mathcal{P}_e$) and strings held by some $\mathcal{N}_e$ with $e > i$ (weight at most $3/e^{2e+3}$, assuming the inductive hypothesis). Thus the total weight held by $\mathcal{N}_i$ at stage t is at most

$$\sum_{e>4i} 2^{-\lfloor e/2\rfloor -4} + \sum_{e>i} \frac{3}{2^{2e+3}} = \frac{2}{2^{2i+3}} + \frac{3}{2^{2i+3}} \sum_{e>i} \frac{1}{4^{e-i}}$$

$$= \frac{2}{2^{2i+3}} + \frac{3}{2^{2i+3}} \sum_{e=1}^{\infty} \frac{1}{4^e} = \frac{2}{2^{2i+3}} + \frac{3}{2^{2i+3}} \frac{1}{3} = \frac{3}{2^{2i+3}}.$$

By induction, the claim holds for all t. □

Lemma 9. $w(V^C) < 1$.

Proof. A string is only enumerated into V^C when it is held by $\mathcal{P}_e$ for some e, and it is removed (by changing C) when it is no longer held by any requirement. The set of strings permanently held by $\mathcal{P}_e$ is $D[\Psi_e^X(n_e)]$ (where X is either A or B depending on the parity of e), so by the above lemmas the set of strings permanently held by the positive requirements is exactly $U^A \cup U^B$, which has total weight at most $1/4$. By the above lemma, the set of strings permanently held by requirement $\mathcal{N}_i$ has weight at most $3/2^{2i+3}$. Thus the total weight (and hence measure) of V^C is at most

$$\frac{1}{4} + \sum_{i=0}^{\infty} \frac{3}{2^{2i+3}} = \frac{1}{4} + \frac{1}{2} = \frac{3}{4} < 1.$$

□

Corollary 10. $A, B \leq_{LR} C$

Thus the low sets A, B, are LR-reducible to the low set C, as desired. □

5. References

References

[1] K. Ambos-Spies, C. Jockusch, R. Shore, and R. Soare, An algebraic decomposition of recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees, *Trans. Amer. Math. Soc.* 281 (1984) 109–128.

[2] G. Barmpalias, A. Lewis, and M. Soskova, Randomness, Lowness and Degrees, *J. Symbolic Logic*, 73 (2008) 559–577.

[3] B. Kjos-Hanssen, Low for random reals and positive measure domination, *Proc. Amer. Math. Soc.*, 135 (2007) 3703–3709.

[4] R. Downey and D. R. Hirschfeldt, *Algorithmic Randomness and Complexity* (Springer-Verlag, Berlin, 2010).

[5] R. Downey, D. Hirschfeldt, A. Nies, and F. Stephan, Trivial reals, in: *Proceedings of the 7th and 8th Asian Logic Conferences* (World Scientific, Singapore, 2003).

[6] Peter Gács, Every sequence is Turing-reducible to a random sequence, *Information and Control* 70 (1986) 186–192.

[7] D. R. Hirschfeldt, A. Nies, and F. Stephan, Using random sets as oracles, *J. London Math. Soc., 2nd series* 75 (2007) 610–622.

[8] B. Kjos-Hanssen, J. Miller, and R. Solomon, Lowness notions, measure, and domination, *J. Symbolic Logic*, 74 (2009) 517–534.

[9] Antonin Kučera, Measure, Π_1^0 classes, and complete extensions of PA, in: *Recursion Theory Week* (Oberwolfach 1984), volume 1141 of *Lecture Notes in Mathematics* (Springer–Verlag, Berlin, 1985).

[10] Joseph Miller, The K-degrees, low for K degrees, and weakly low for K sets, *Notre Dame J. Form. Log.*, 50 (2010) 381–391.
[11] André Nies, Lowness properties and randomness, *Adv. Math.*, 197 (2005) 274–305.
[12] Stephen G. Simpson. Almost everywhere domination and superhighness. *Math. Logic Quarterly*, 53 (2007) 462–482.

E-mail address: ded@msor.vuw.ac.nz

Department of Mathematics, Statistics, and Operations Research, Victoria University of Wellington, P.O. Box 600, Wellington, 6140, New Zealand