Bounded low and high sets

Bernard A. Anderson
Department of Mathematics and Computer Science
Gordon State College
419 College Drive
Barnesville, GA 30204, United States
banderson@gordonstate.edu
faculty.gordonstate.edu/banderson/

Barbara F. Csima
Department of Pure Mathematics
University of Waterloo,
Waterloo, ON N2L 3G1, Canada
csima@uwaterloo.ca
www.math.uwaterloo.ca/∼csima

Karen M. Lange
Department of Mathematics
Wellesley College
106 Central St.
Wellesley, MA 02481, United States
karen.lange@wellesley.edu
palmer.wellesley.edu/∼klange/

March 20, 2017

Abstract

Anderson and Csima [1] defined a jump operator, the *bounded jump*, with respect to bounded Turing (or weak truth table) reducibility. They showed that the bounded jump is closely related to the Ershov hierarchy and that it satisfies an analogue of Shoenfield jump inversion. We show that there are high bounded low sets and low bounded high sets. Thus, the information coded in the bounded jump is quite different from that of the standard jump. We also consider whether the analogue of the Jump Theorem holds for the bounded jump: do we have $A \leq_{bT} B$ if and only if $A^b \leq_1 B^b$? We show the forward direction holds but not the reverse.

1 Introduction

There are many models of computation based on restrictions of Turing reducibility $\leq_T$. Here we consider *bounded Turing reducibility* (also known as *weak truth table reducibility*). For two subsets A, B of natural numbers, A is *bounded Turing reducible to* B, written $A \leq_{bT} B$, if $A \leq_T B$ and there is a computable function $g : \omega \to \omega$ such that determining $A(n)$ requires only consulting the first $g(n)$ digits of B. In [1], Anderson and Csima defined an analogue of the standard jump tailored to work with $\leq_{bT}$. They called this operator the *bounded jump* (see Definition 1) and denoted the bounded jump of a set A by A^b. They explored the properties of the bounded jump and compared it to other potential jump operators for weaker forms of computation. In particular, the bounded jump is strictly increasing and order preserving on the bounded Turing degrees, just as the standard jump is on the Turing degrees. They also showed that the bounded jump hierarchy is closely connected to the Ershov hierarchy of α-c.e. sets (see Theorem 1.2). Here we compare bounded low and bounded high sets to their standard jump counterparts. Recall that a set is *low* (respectively *high*) if the set's standard jump has the minimum (respectively maximum) possible complexity. We can similarly define bounded low and bounded high sets. We show in Theorems 2.1 and 2.5 that there are low bounded high sets and high bounded low sets. Hence, the amount of information coded into the bounded jump does not help us understand the complexity of the standard jump, and vice versa.

It is natural to ask which classical results hold for the bounded Turing degrees, possibly when using the bounded jump. Anderson [2], building on work of Mohrherr [10], proved the analogue of Friedberg's jump inversion in the truth table degrees: for all $X \geq_{tt} \emptyset'$, there exists a set A such that $A' \equiv_{tt} X \equiv_{tt} A \oplus \emptyset'$. Moreover, the same proof gives the result in the bounded Turing degrees. However, Csima, Downey, and Ng [5] showed that Schoenfield (and hence Sacks) jump inversion fails when the standard jump is used. Anderson and Csima obtained the analogue with the bounded jump: given a set B such that $\emptyset^b \leq_{bT} B \leq_{bT} (\emptyset^b)^b$, there is a set $A \leq_{bT} \emptyset^b$ such that $A^b \equiv_{bT} B$. They leave unanswered whether A can be taken to be computably enumerable, i.e., whether the analogue of Sacks jump inversion holds. In §3, we consider the classical Jump Theorem, which states that for any sets A and B, we have $A \leq_T B \Leftrightarrow A' \leq_1 B'$. We wish to determine whether the analogue of the Jump Theorem holds for the bounded jump; do we have $A \leq_{bT} B \Leftrightarrow A^b \leq_1 B^b$? We find that the forward direction holds, but the converse does not.

1.1 Notation and Definitions

For notation and background not described below, see Cooper [4] and Soare [13] [12]. All sets considered are subsets of ω. We let $\varphi_0, \varphi_1, \varphi_2, \ldots$ be an effective enumeration of the partial computable functions and $\Phi_0, \Phi_1, \Phi_2, \ldots$ be an effective enumeration of the Turing functionals. We assume our enumerations are acceptable.

We let $\emptyset' = \{x \mid \varphi_x(x) \downarrow\}$, and for an arbitrary set A, let $A' = \{x \mid \Phi_x^A(x) \downarrow\}$. In the case that the enumeration $\{\varphi_n\}_{n \in \omega}$ is such that $\varphi_n = \Phi_n^{\emptyset}$, there is no confusion with the two definitions of $\emptyset'$. But under any enumeration, the two definitions are 1-equivalent.

For a set A, we let $A \upharpoonright x = \{n \in A \mid n \leq x\}$. We follow an expression with a stage number in brackets (i.e. $[s]$) to indicate the stage number applies to everything in the expression that is indexed by stage.

For sets A and B we write that $A \leq_{bT} B$, and say A is *bounded Turing reducible* to B, if there exist i and j such that φ_j is total and for all x, we have $A(x) = \Phi_i^{B \upharpoonright \varphi_j(x)}(x)\downarrow$.

Definition 1. Given a set A, the *bounded jump of A*, is

$$A^b = \{x \in \omega \mid (\exists i \leq x)[\varphi_i(x)\downarrow \,\wedge\, \Phi_x^{A \upharpoonright \varphi_i(x)}(x)\downarrow]\}.$$

We let A^{nb} denote the n-th bounded jump.

We remark that there are several other potential definitions for a bounded jump. In particular in §3, we mention another possibility A^{b_0}, which satisfies $A^{b_0} \equiv_{tt} A^b$ but in general does not satisfy $A^{b_0} \equiv_1 A^b$. We choose to use A^b as the bounded jump because of its connection with the Ershov hierarchy shown in Theorem 1.2, and because we believe that a jump operator should be as 1-complete as possible. For example, consider the class $\mathscr{C} = \{X \mid X \leq_{bT} (0')^{b_0}\}$ (the class $\mathscr{C}$ is the same for any likely bounded jump operator used in place of b_0). We show in [1] that for any $X \in \mathscr{C}$ we have $X \leq_1 (0')^b$, but a similar fact does not hold for b_0 (or other likely bounded jump operators). See [1] for other definitions and further discussion.

We list a few basic properties of the bounded jump.

Lemma 1.1 (Anderson & Csima [1])**.** *For all sets A, we have* $A \leq_1 A^b \leq_1 A'$ *and* $A^b \equiv_T A \oplus \emptyset'$. *Moreover,* $\emptyset^b \equiv_1 \emptyset'$.

As mentioned earlier, a set is low (high) with respect to a given jump operator if its jump encodes the least (most) possible information. A set A is *low* if $A' \leq_T \emptyset'$ and is *bounded low* if $A^b \leq_{bT} \emptyset^b$. A set $A \leq_T \emptyset'$ is *high* if $A' \geq_T \emptyset''$, and a set $A \leq_{bT} \emptyset^b$ is *bounded high* if $A^b \geq_{bT} \emptyset^{2b}$.

We briefly review the α-c.e. sets so that we can state Anderson and Csima's result connecting the bounded jump hierarchy and the Ershov hierarchy. Fix a canonical, computable coding of the ordinals less than ω^ω. Since we do not use ordinals above ω^ω in this paper, the details of the coding are not significant. We say a function on an ordinal α is (partial) computable if the corresponding function on codes for the ordinal α is (partial) computable.

For $\alpha \geq \omega$, we say that a set A is *α-c.e.* if there is a partial computable $\psi : \omega \times \alpha \to \{0,1\}$ such that for every $n \in \omega$, there exists a $\beta < \alpha$ where $\psi(n,\beta)\downarrow$ and $A(n) = \psi(n,\gamma)$ where γ is least such that $\psi(n,\gamma)\downarrow$ [9]. We remark that a set A is ω-c.e. if and only if there are computable functions $f : \omega \times \omega \to \{0,1\}$ and $g : \omega \to \omega$ such that $A(n) = \lim_{s\to\infty} f(n,s)$ and $|\{s \mid f(n,s+1) \neq f(n,s)\}| \leq g(n)$. Thus, the above definition of α-c.e. generalizes the common usage of the term "ω-c.e." Note that this definition differs slightly from the definitions in [3].

The following theorem generalizes the classical result that $A \leq_{bT} \emptyset'$ iff A is ω-c.e. Though this was not made clear in [1], the theorem assumes a notation of ω^n for which we can computably work with the Cantor normal forms of all ordinals less than ω^n. (In particular, such a notation allows us to computably view ω^n as the lexicographical order on n-tuples from ω.)

Theorem 1.2 (Anderson & Csima [1])**.** *For any set X and $n \geq 2$,*

$$X \leq_{bT} \emptyset^{nb} \iff X \text{ is } \omega^n\text{-c.e.} \iff X \leq_1 \emptyset^{nb}.$$

This theorem cannot hold for all notations, as by [7], every Δ^0_2 set is ω^2-c.e. for some notation of ω^2.

2 Comparing jump classes

2.1 Bounded Low

The next example shows that bounded low sets can code substantial information in the Turing degrees. We will need the following definitions and result (see [12]). Given functions f and g, the function f *dominates* g if $f(n) \geq g(n)$ for all but finitely many n; the function f is called *dominant* if f dominates every total computable function. Martin proved that A satisfies $\emptyset'' \leq_T A'$ iff there is a dominant function $f \leq_T A$. Given an infinite set $X = \{a_0 < a_1 < a_2 < \ldots\}$, the *principal function of* X is defined such that $p_X(n) = a_n$.

Theorem 2.1. *There exists a c.e. bounded low set that is high.*

Proof. To ensure that A is high, we build A so that $p_{\bar{A}}$ is dominant by satisfying for all $i \in \omega$:

$\mathbf{R}_i$: If φ_i is total, then $p_{\bar{A}}$ dominates φ_i, i.e., $(\exists m)(\forall l \geq m)[p_{\bar{A}}(l) \geq \varphi_i(l)]$.

Next note that A is bounded low iff $A^b \leq_{bT} \emptyset'$ iff A^b is ω-c.e. Requirements $\mathbf{R}_i$ will demand that we enumerate many elements in A. The requirement $\mathbf{Q}_x$ imposes restraint on all requirements $\mathbf{R}_i$ for $i \geq x$ so that we can ensure A^b is ω-c.e.

$\mathbf{Q}_x$: If $\varphi_{n,s}(x)\downarrow$ with $n \leq x$, then all requirements $\mathbf{R}_i$ for $i \geq x$ are restrained from enumerating any $y \leq \varphi_n(x)$ into A after stage s.

In fact, the computable function $g(x) = 2(x+1)^2$ will bound the number of changes in the natural approximation of $A^b(x)$.

Finally, we ensure that A is coinfinite by satisfying for all $e \in \omega$:

$\mathbf{N}_e$: $|\bar{A}| \geq e$.

At each stage s, we have a c.e. approximation A_s of A and a current approximation of $\bar{A}_s = \{a_{0,s} < a_{1,s} < a_{2,s} < \ldots\}$. To satisfy $\mathbf{N}_e$, no requirement $\mathbf{R}_i$ for $i \geq e$ may enumerate any element $a_{l,s}$ into A with $l \leq e$ at stage s. For φ_i associated with $\mathbf{R}_i$, we assume, without loss of generality, for all $z, s \in \omega$ that if $\varphi_i(z+1)[s]$ converges then $\varphi_i(z)[s]$ converges.

At stage $s = 0$, let $A_0 = \emptyset$ and $a_{j,0} = j$ for all $j \in \omega$.

At stage $s+1$, we first calculate the restraints imposed by requirements $\mathbf{Q}_x$ and $\mathbf{N}_e$ for each requirement $\mathbf{R}_i$.

Let $r(x,s) = \max_{n \le x} \varphi_n(x)[s]$. Let

$$r_{i,s} = \max_{x \le i}\{r(x,s), a_{i,s}\} = \max_{n \le x \le i}\{\varphi_n(x)[s], a_{i,s}\}. \tag{1}$$

Note that $r(x,s) \le r(x+1,s)$, $r_{i,s} \le r_{i,s+1}$, and $r_{i,s} \le r_{i+1,s}$ for all $x,i,s \in \omega$. We allow requirement $\mathbf{R}_i$ to enumerate any element greater than $r_{i,s}$ into A.

We check whether any $\mathbf{R}_i$ may act while obeying the restraints. We ask whether

$$(\exists i \le s)(\exists x \le s)[x > r_{i,s} \ \&\ \varphi_i(x)[s] \downarrow > a_{x,s}]. \tag{2}$$

If so, let $\langle i,x \rangle$ be the least pair satisfying (2). We act for $\mathbf{R}_i$ to guarantee that $p_{\bar{A}}(x) \ge \varphi_i(x)$. Let l be the least number such that $a_{l,s} \ge \max_{m,y \le s}\{\varphi_m(y)[s]\} \ge \varphi_i(x)$. We enumerate $a_{x,s}, a_{x+1,s}, \dots, a_{l-1,s}$ into $A_{s+1} \supseteq A_s$ so that $a_{x,s+1} = a_{l,s} \ge \varphi_i(x)$. This action ends stage $s+1$. This completes our description of the construction.

Lemma 2.2. *For all $x,e,i \in \omega$, the limits $\lim_{s\to\infty} r(x,s)$, $\lim_{s\to\infty} a_{e,s} = a_e$, and $\lim_{s\to\infty} r_{i,s} = r_i$ all exist. Hence, $\mathbf{N}_e$ is satisfied for all $e \in \omega$, so A is coinfinite.*

Proof. Given $x \in \omega$, there is a stage s_0 by which any $\varphi_n(x)$, with $n \le x$, converges if it ever will. Hence, $\lim_{s\to\infty} r(x,s)$ exists.

Given $e \in \omega$, suppose for all $j < e$, $\lim_{s\to\infty} a_{j,s} = a_j$ is defined. We show that $\lim_{s\to\infty} a_{e,s} = a_e$ is defined. Let s_0 be a stage such that, for all $j < e$ and later stages $s' \ge s_0$, we have $a_{j,s'} = a_j$. Hence, no elements below a_{e,s_0} are enumerated into A after stage s_0. Note that if requirement $\mathbf{R}_i$ acts on behalf of a given pair $\langle i,x \rangle$ as in (2), requirement $\mathbf{R}_i$ never satisfies (2) for pair $\langle i,x \rangle$ again. Thus, there is a stage $s_1 \ge s_0$ by which any $\varphi_i(e)$, with $i \le e$, converges if it ever will and after which no $\mathbf{R}_i$ acts on behalf of $\langle i,x \rangle$ for $x \le e$.

We claim $\lim_{s\to\infty} a_{e,s} = a_{e,s_1}$. If not, at a later stage $s' \ge s_1$, some $\mathbf{R}_{i'}$ must have enumerated $a_{x,s'}$ into A for some $x \le e$. Since $a_{j,s} = a_{j,s_1} = a_{j,s_0}$ for all $j < e$ and $s \ge s_0$, we must have that $x = e$. Then, $e > r_{i',s'}$ and $\varphi_{i'}(e)[s'] \downarrow > a_{e,s'}$. By definition, $r_{i,s} \ge a_{i,s} \ge i$ for all i and s, so $i' < e$. By construction, $\mathbf{R}_{i'}$ would act at stage s' for the pair $\langle i',e \rangle$, contradicting our choice of s_1. Hence, $a_{e,s} = a_{e,s_1}$ for all $s \ge s_1$, so $\lim_{s\to\infty} a_{e,s}$ exists.

Since $r_{i,s} = \max_{x \le i}\{r(x,s), a_{i,s}\}$ and both $\lim_{s\to\infty} r(x,s)$ and $\lim_{s\to\infty} a_{i,s} = a_i$ exist, $\lim_{s\to\infty} r_{i,s} = r_i$ exists.

□

Lemma 2.3. *The requirement $\mathbf{R}_i$ is satisfied for all $i \in \omega$. Thus, $p_{\bar{A}}$ is a dominant function, so A is high.*

Proof. Suppose φ_i is total. Let $r_i = \lim_{s\to\infty} r_{i,s}$ and $a_l = \lim_{s\to\infty} a_{l,s}$. We show that $p_{\bar{A}}$ dominates φ_i. Take $l > r_i$. Let s_0 be a stage such that $r_{i,s} = r_i$ and $a_{l,s} = a_l$ for all $s \ge s_0$. Since $a_{l,s}$ never changes after stage s_0, we have that $\mathbf{R}_i$ never acts for $\langle i,l \rangle$ after stage s_0. Thus, $\varphi_i(l) \le a_l = p_{\bar{A}}(l)$ since $l > r_i$.

□

Lemma 2.4. *The set A^b is ω-c.e.*

Proof. We show that

$$A^b = \{x \in \omega \mid (\exists n \leq x)[\Phi_x^{A \upharpoonright \varphi_n(x)\downarrow}(x)\downarrow]\} \tag{3}$$

is ω-c.e. by showing that its (computable) natural approximation

$$f(x,s) = \begin{cases} 1 & \text{if } (\exists n \leq x)[\Phi_x^{A \upharpoonright \varphi_n(x)\downarrow}[s](x)\downarrow] \\ 0 & \text{else} \end{cases}$$

satisfies:

1. $\lim_{s\to\infty} f(x,s) = A^b(x)$ and
2. $f(x,s) \neq f(x,s+1)$ for at most $g(x) = 2(x+1)^2$ many stages s.

Suppose $f(x,s) = 1$, i.e., there is some $n \leq x$ such that $\Phi_x^{A \upharpoonright \varphi_n(x)\downarrow}[s](x)\downarrow$. By construction, the only requirements that may enumerate elements below $\varphi_n(x)$ into A after stage s are $\mathbf{R}_i$ for $i < x$. Hence, only these x-many requirements may injure the computation $\Phi_x^{A \upharpoonright \varphi_n(x)\downarrow}[s](x)\downarrow$, potentially causing $f(x,t)$ to equal 0 for some $t > s$. We show that each of these x-many requirements may injure a computation $\Phi_x^{A \upharpoonright \varphi_n(x)\downarrow}(x)$ on a given use $\varphi_n(x)$ with $n \leq x$ at most once.

Suppose requirement $\mathbf{R}_k$ for $k < x$ enumerates elements below $\varphi_n(x)$ into A at stage $s' > s$ while acting for the pair $\langle k, y\rangle$. By construction, $\mathbf{R}_k$ enumerates $a_{y,s'}, \ldots, a_{l-1,s'}$ into A where $a_{y,s'} < \varphi_n(x)$ and l is the least value such that

$$a_{l,s'} \geq \max_{m,z \leq s'} \{\varphi_m(z)[s']\}. \tag{4}$$

So, we have that $a_{y,s'+1} = a_{l,s'} \geq \varphi_n(x)[s']$.

We assumed for all $i, z, s \in \omega$ that if $\varphi_i(z+1)[s]$ converges then $\varphi_i(z)[s]$ converges. It is also easy to see that the restraint $r_{i,s}$ on $\mathbf{R}_i$ is nondecreasing in s. Moreover, at any given stage of the construction, the least pair $\langle i, z\rangle$ satisfying (2) receives attention. These observations imply that if requirement $\mathbf{R}_i$ acts on behalf of the pairs $\langle i, z\rangle$ and $\langle i, z'\rangle$ with $z < z'$ at stages s and s' respectively, then $s < s'$. Since requirement $\mathbf{R}_k$ acts at stage $s' > s$ for the pair $\langle k, y\rangle$, requirement $\mathbf{R}_k$ will never act for a pair $\langle k, y'\rangle$ where $y' < y$ after stage s'. If $\mathbf{R}_k$ acts at a later stage $t > s'$ for a pair $\langle k, y''\rangle$ where $y < y''$, the requirement will only enumerate elements a into A such that

$$a \geq a_{y'',t} \geq a_{y'',s'+1} > a_{y,s'+1} \geq \varphi_n(x)[s']. \tag{5}$$

Therefore, a single requirement $\mathbf{R}_k$ with $k < x$ can injure a given use $\varphi_n(x)$ for the computation $\Phi_x^{A \upharpoonright \varphi_n(x)\downarrow}(x)$ at most once.

There are $(x+1)$-many φ_n for $n \leq x$ that may compute a use $\varphi_n(x)$ for the computation $\Phi_x^{A \upharpoonright \varphi_n(x)\downarrow}(x)$. Thus, $f(x,s) \neq f(x,s+1)$ for at most $g(x) = 2(x+1)^2$ stages s, since for each of the $(x+1)$-many potential uses, there are x-many requirements that may injure that use at most once (after the first convergence with that use), and each injury may lead to two changes in $f(x,s)$ (divergence followed by reconvergence). □

□

2.2 Bounded high

We show a set's bounded jump may be maximally complex while its standard jump codes the minimum amount.

Theorem 2.5. *There exists a low set $A \leq_{bT} \emptyset'$ that is bounded high.*

This result also gives an example of a low set that is not bounded low as well as an example of an incomplete bounded high set.

Proof. We construct an ω-c.e. set A (and hence $A \leq_{bT} \emptyset'$) that satisfies $A' \leq_T \emptyset'$ and $A^b \geq_{bT} \emptyset^{2b} = (\emptyset^b)^b$. To guarantee that A is ω-c.e., we give a computable approximation $A[s]$ of A so that $A(n)[s]$ changes at most n times. We satisfy the standard requirements to guarantee that A is low.

$\mathbf{N}_e$: $\quad (\exists^\infty s)\left[\Phi_e^A(e)[s]\downarrow\right] \implies \Phi_e^A(e)\downarrow.$

To meet $\mathbf{N}_e$, we define the *restraint function* $r(e,s) = \max_{t\leq s} u(e,t)$, where $u(e,s)$ is the use of the computation $\Phi_e^A(e)[s]$ if it converges and 0 otherwise. At any stage in the construction, we do not allow any $\mathbf{P}_i$ for $i \geq e$ to change $A(x)$ for any $x < r(e,s)$.

To show that $A^b \geq_{bT} \emptyset^{2b}$, we produce an effective procedure Ψ and a computable function f so that Ψ computes whether $n \in \emptyset^{2b}$ using only the oracle $A^b \upharpoonright f(n)$. We break this task into the following positive requirements, one for each $n \in \omega$.

$\mathbf{P}_n$: $\quad \Psi^{A^b \upharpoonright f(n)}(n)\downarrow = \emptyset^{2b}(n).$

We take advantage of the fact that $\emptyset^{2b}$ is ω^2-c.e. by Theorem 1.2 to satisfy these requirements. Since $\emptyset^{2b}$ is ω^2-c.e., there is a partial computable approximation function $\psi : \omega \times \omega^2 \to \{0,1\}$ for $\emptyset^{2b}$. For a fixed n, at each stage s, the function ψ determines an ordinal $\beta_{n,s} < \omega^2$ that witnesses an approximation $\emptyset^{2b}(n)[s]$ to $\emptyset^{2b}(n)$. The sequence $\{\beta_{n,s}\}_{s\in\omega}$ decreases finitely often, with $\emptyset^{2b}(n)[s]$ changing only at ordinal decreases and giving a correct guess after the last decrease. To fix notation, say $\beta_{n,s}$ has the form $\omega \cdot l_{n,s} + k_{n,s}$, where $l_{n,s}, k_{n,s} \in \omega$.

We begin with a brief overview of how we encode $\emptyset^{2b}(n)$ into $A^b \upharpoonright f(n)$. We code $\emptyset^{2b}(n)$ into a dedicated block of indices, called the *A^b-block of indices for* $\mathbf{P}_n$. We change the coding only when there is a decrease in $\beta_{n,s}$, which itself results in a decrease in the limit term $l_{n,s}$ or the finite term $k_{n,s}$. Our coding procedure will treat a decrease in the limit term $l_{n,s}$ differently from a decrease only in the finite term.

Each time the limit term $l_{n,s}$ in $\beta_{n,s}$ decreases, we will encode $\emptyset^{2b}(n)[s]$ into a new subblock of the block dedicated to $\mathbf{P}_n$. We know at stage 0 that we will need at most $l_{n,0}$ such subblocks. Moreover, we will see that we can determine the size and location of these subblocks at the start of the construction so that we can obtain the computable bound $f(n)$ in $\mathbf{P}_n$. When $l_{n,s}$ decreases to $l := l_{n,s+1}$, we get an upper bound (specifically $k_{n,s+1}$) on the number of times $\emptyset^{2b}(n)[s]$ can change while the limit term of the witnessing ordinal equals l. We ensure that the reduction from $A^b[s]$ on the fixed subblock depends on the membership in A of a (dynamically allocated) value $c_{n,s}$, which is greater than this upper bound and other restraints. This choice will allow us to change $A^b[s]$ to match the approximation $\emptyset^{2b}(n)[s]$ while maintaining that A is ω-computably enumerable.

We now turn to the details of the construction. To make the notation above precise, we set $\beta_{n,s}$ to be the least $\beta < \omega^2$ such that $\psi(n,\beta)[s]\downarrow$ if such a β exists. If not, we let $\beta_{n,s}$ be the first ordinal less than ω^2 (when dovetailing computations) for which we see $\psi(n,\beta_n)\downarrow$. Then, the function $(n,s) \to \beta_{n,s}$ is computable. Since $\beta_{n,s} < \omega^2$, $\beta_{n,s} = \omega \cdot l_{n,s} + k_{n,s}$ for some $l_{n,s}, k_{n,s} \in \omega$ where $l_{n,s}$ and $k_{n,s}$ are computable from the notation for $\beta_{n,s}$. We write $\emptyset^{2b}(n)[s] = \psi(n,\beta_{n,s})$. Without loss of generality, we may assume that $\emptyset^{2b}(n)[s]$ changes whenever $\beta_{n,s}$ decreases.

In order to construct the reduction Ψ from A^b to $\emptyset^{2b}$, we will need to control whether or not infinitely many elements reside in A^b. To that end, we fix computable injective functions g and h such that for all $x \in \omega$ we control $\Phi_{g(x)}$ and $\varphi_{h(x)}$ where $h(x) < g(x)$. The procedure Ψ will only use the membership of (finitely many) indices $g(\langle n,l,y\rangle)$ in A^b to compute $\emptyset^{2b}(n)$. As mentioned above, we call this collection of indices the *A^b-block of indices for $\mathbf{P}_n$*, and it is is made up of disjoint subblocks of indices indexed by $l \leq l_{n,0}$. We call the l^{th} such subblock the *location l subblock of the A^b-block for* $\mathbf{P}_n$. Recall that we use this subblock to code $\emptyset^{2b}(n)$ into A^b when $l_{n,s} = l$. The location l subblock for $\mathbf{P}_n$ consists of

1. A single index, called the *location index*, used to indicate whether we ever stopped coding in this subblock, i.e., whether $l_{n,s} < l$ at some stage s.
2. A collection of indices, called the *coding indices*, used to code $\emptyset^{2b}(n)$ if $l = \lim_s l_{n,s}$; and
3. A collection of indices, one for each coding index, called the *injury accounting indices*, used to change the decoding functional Ψ when $\mathbf{P}_n$ is injured. We give a computable function $q(n)$ in Lemma 2.7 and prove there that reserving $q(n)$ many coding indices and $q(n)$ many injury indices suffices for the construction.

We can (uniformly in n and $l \leq l_{n,0}$) computably assign $1+2q(n)$ indices of the form $g(\langle n,l,y\rangle)$ to the location l subblock for $\mathbf{P}_n$. Given this assignment of indices, one can find a computable function $f:\omega \to \omega$ such that $f(n)$ is larger than all indices in the A^b-block for $\mathbf{P}_n$. Note that the A^b-block of indices for $\mathbf{P}_n$ remains constant throughout the construction and that the A^b-blocks for $\mathbf{P}_n$ and $\mathbf{P}_m$ with $n \neq m$ are disjoint.

At each stage s, we let $A^b[s] \upharpoonright \mathbf{P}_n$ be the restriction of the characteristic function of $A^b[s]$ to the A^b-block for $\mathbf{P}_n$. During the construction, we let $v_{n,s}$ denote the stage s coding index for $\emptyset^{2b}(n)$ in the location $l_{n,s}$ subblock of the A^b-block for $\mathbf{P}_n$. As mentioned above, the coding of $\emptyset^{2b}(n)[s]$ into $v_{n,s}$ will depend only on whether a (dynamic) index $c_{n,s}$ is a member of A. We let $A[s] \upharpoonright \mathbf{P}_n$ denote the restriction of the characteristic function of $A[s]$ to $\{c_{n,s}\}$.

During the construction, when we say "define $\Psi^{(A^b[s]\upharpoonright \mathbf{P}_n)}(n)$ to halt" (respectively "define $\Phi_j^{(A[s]\upharpoonright \mathbf{P}_n)}(x) = y$"), we mean that the computation holds for all oracles that agree with the stage s approximation of the A^b-block for $\mathbf{P}_n$ (respectively the stage s approximation of A at $c_{n,s}$).

In addition to coding $\emptyset^{2b}(n)[s]$ into $v_{n,s}$, we will also enumerate other indices into A^b during the construction. To enumerate some index $g(x)$ other than $v_{n,s}$ into A^b, we will simply let $\varphi_{h(x)}(g(x))\downarrow= 0$ and let $\Phi^X_{g(x)}(g(x))\downarrow$ for all oracles X.

2.2.1 Initializing $\mathbf{P}_n$

We describe how we *initialize for* $\mathbf{P}_n$ *at stage* s. This module encodes $\emptyset^{2b}(n)[s]$ into A^b at location index $v_{n,s}$ in the $l_{n,s}$-subblock for $\mathbf{P}_n$. First, let $c_{n,s}$ be a fresh element larger than $k_{n,s}$ (where $\beta_{n,s} = \omega \cdot l_{n,s} + k_{n,s}$), $\max_{e \le s} r(e,s)$, and all previously used elements in A.

The module depends on whether $\mathbf{P}_n$ has been initialized previously. If it has, we take action to destroy our previous encoding of $\emptyset^{2b}(n)[s-1]$ and choose the new coding location. We consider whether $l_{n,s} = l_{n,s-1}$. If $l_{n,s} < l_{n,s-1}$, we set $v_{n,s}$ to be the first coding index in the location $l_{n,s}$ subblock for $\mathbf{P}_n$. We guarantee that the reduction $\Psi^{A^b \restriction \mathbf{P}_n}$ consults the $l_{n,s}$ subblock by enumerating into A^b the location index of each location l subblock for $\mathbf{P}_n$ where $l > l_{n,s}$.

Now suppose $l_{n,s} = l_{n,s-1}$. We will see that we only initialize $\mathbf{P}_n$ in this case when $\mathbf{P}_n$ is injured. Hence, we enumerate into A^b the next unused injury accounting index in the location $l_{n,s}$ subblock for $\mathbf{P}_n$, and we let $v_{n,s}$ be the next unused coding index in this subblock. Regardless of whether $l_{n,s} = l_{n,s-1}$, these actions destroy any previously defined computation $\Psi^{(A^b[s-1] \restriction \mathbf{P}_n)}(n)\downarrow$. (If stage s is the first stage at which we initialize $\mathbf{P}_n$, we have no such computations to destroy.)

In all cases, we encode $\emptyset^{2b}(n)[s]$ into A^b at the coding index $v_{n,s}$ in the location $l_{n,s}$ subblock for $\mathbf{P}_n$. We enumerate $v_{n,s}$ into A^b if and only if $\emptyset^{2b}(n)[s] = 1$. We accomplish this enumeration by picking z so that $g(z) = v_{n,s}$, setting $\varphi_{h(z)}$ equal to the constant function $c_{n,s}$, and defining $\Phi_{v_{n,s}}^{(A[s] \restriction \mathbf{P}_n)}$ to halt on all inputs. Finally, we define the reduction Ψ, when given an oracle agreeing with $(A^b[s] \restriction \mathbf{P}_n)$, to output the value of $A^b[s]$ at $v_{n,s}$ on input n. Thus,

$$\Psi^{(A^b[s] \restriction \mathbf{P}_n)}(n) = (A^b[s] \restriction \mathbf{P}_n)(v_{n,s}).$$

$$\text{So, } \Psi^{(A^b[s] \restriction \mathbf{P}_n)}(n) = \emptyset^{2b}(n)[s].$$

2.2.2 The construction

At stage 0, we initialize $\mathbf{P}_0$ for the first time and set $A(n)[0] = 0$ for all $n \in \omega$. At stage $s+1$, we suppose the following statements hold for all $n \le s$.

1. $\Psi^{(A^b[s] \restriction \mathbf{P}_n)}(n) := (A^b[s] \restriction \mathbf{P}_n)(v_{n,s}) = \emptyset^{2b}(n)[s]$.

2. If $v_{n,s} \in A^b[s]$, then $\varphi_{j_{n,s}} = c_{n,s}$ provides the required oracle bound and $c_{n,s} \notin A[s]$. Moreover, the computation witnessing $v_{n,s} \in A^b[s]$ only asks about the membership of $c_{n,s}$ in A.

3. $c_{n,s} > \max_{e \le n} r(e,s)$.

At stage $s+1$, we consider whether we need to take action for $\mathbf{P}_n$ for each $n \le s$ (in order of priority). First, if restraint impedes $\mathbf{P}_n$ from acting (specifically if $c_{n,s} \le \max_{e \le n} r(e, s+1)$), we consider $\mathbf{P}_n$ *injured*, and we initialize $\mathbf{P}_n$ at stage $s+1$. If not, we check whether the approximation of $\emptyset^{2b}(n)$ changed. If the approximation remains the same, we do nothing beyond updating all values, e.g., setting $v_{n,s+1} = v_{n,s}$, $(A^b[s+1] \upharpoonright \mathbf{P}_n) = (A^b[s] \upharpoonright \mathbf{P}_n)$, etc. Now suppose that the approximation changed, so

$$\beta_{n,s+1} = \omega \cdot l_{n,s+1} + k_{n,s+1} < \beta_{n,s} = \omega \cdot l_{n,s} + k_{n,s}.$$

Our action depends on whether $l_{n,s+1} = l_{n,s}$. If $l_{n,s+1} < l_{n,s}$, we initialize $\mathbf{P}_n$. Otherwise $l_{n,s+1} = l_{n,s}$ so $k_{n,s+1} < k_{n,s}$. We continue working with the coding indices $v_{n,s+1} := v_{n,s}$ and $c_{n,s+1} := c_{n,s}$. We encode $\emptyset^{2b}(n)[s+1]$ into location $v_{n,s+1}$ of $A^b[s+1]$ and ensure that the reduction Ψ, given an oracle agreeing with $(A^b[s+1] \upharpoonright \mathbf{P}_n)$, outputs the value encoded at $v_{n,s}$ on input n.

By our inductive hypotheses, if $v_{n,s+1} \in A^b[s]$, then $\emptyset^{2b}(n)[s] = 1$ and $\emptyset^{2b}(n)[s+1] = 0$. Furthermore, $c_{n,s+1} \notin A[s]$. We change the approximation of A so that $c_{n,s+1} \in A[s+1]$. This action destroys the computation $\Phi_{v_{n,s+1}}^{(A[s] \upharpoonright \mathbf{P}_n)}(v_{n,s+1}) \downarrow$, so $v_{n,s+1} \notin A^b[s+1]$. Finally, we define $\Psi^{(A^b[s+1] \upharpoonright \mathbf{P}_n)}(n) = (A^b[s+1] \upharpoonright \mathbf{P}_n)(v_{n,s+1})$ (if it is not already). Then, $\Psi^{(A^b[s+1] \upharpoonright \mathbf{P}_n)}(n) = 0 = \emptyset^{2b}(n)[s+1]$.

By our inductive hypotheses, if $v_{n,s+1} \notin A^b[s]$, then $\emptyset^{2b}(n)[s] = 0$ and $\emptyset^{2b}(n)[s+1] = 1$. If $c_{n,s+1} \notin A[s]$, we have not changed the approximation of A at $c_{n,s+1}$ since $\mathbf{P}_n$ was last initialized. In this case, we let $A[s+1] = A[s]$, and we enumerate $v_{n,s+1}$ into A^b at stage $s+1$ in the same manner as before. Finally, we define $\Psi^{(A^b[s+1] \upharpoonright \mathbf{P}_n)}(n) = (A^b[s+1] \upharpoonright \mathbf{P}_n)(v_{n,s+1})$. Otherwise, $c_{n,s} \in A[s]$, and we have changed the approximation of A at $c_{n,s+1}$ since $\mathbf{P}_n$ was last initialized. We change the approximation of A so that $c_{n,s+1} \notin A[s+1]$. This action reinstates the previously defined computations $\Phi_{v_{n,s+1}}^{(A[s+1] \upharpoonright \mathbf{P}_n)}(v_{n,s+1}) \downarrow$ and $\Psi^{(A^b[s+1] \upharpoonright \mathbf{P}_n)}(n) = (A^b[s+1] \upharpoonright \mathbf{P}_n)(v_{n,s+1})$. Regardless of whether $c_{n,s} \in A[s]$, we have $\Psi^{(A^b[s+1] \upharpoonright \mathbf{P}_n)}(n) = 1 = \emptyset^{2b}(n)[s+1]$ in the case $v_{n,s+1} \notin A^b[s]$.

The final step of stage $s+1$ is to initialize $\mathbf{P}_{s+1}$. This completes the construction.

2.2.3 Verification

As usual, we say requirement $\mathbf{N}_e$ is *injured* at stage $s+1$ whenever some $\mathbf{P}_n$ with $n < e$ enacts a change $A[s](x) \neq A[s+1](x)$ for some $x < r(e,s)$. We say $\mathbf{P}_n$ *acts* at stage $s+1$ if $\mathbf{P}_n$ is injured at that stage or $\emptyset^{2b}(n)[s+1] \neq \emptyset^{2b}(n)[s]$.

Lemma 2.6. *For all $e \in \omega$, requirement $\mathbf{N}_e$ is met (and is injured at most finitely often) and $r(e) = \lim_s r(e,s)$ exists.*

Proof. We prove the lemma for $\mathbf{N}_e$ by induction. By the inductive hypothesis, there is a stage s such that for all $t \geq s$ and $n < e$, requirement $\mathbf{N}_n$ is not injured at stage t and $r(n) = r(n,t) = r(n,s)$. Since ψ is an approximation function for the ω^2-c.e. set $\emptyset^{2b}$, there is a stage $s' > s$ such that $\emptyset^{2b}(e)[s'] = \emptyset^{2b}(e)[t]$ for all $t \geq s'$ and $n < e$. By construction and choice of s', no $\mathbf{P}_n$ with $n < e$ acts after stage s'. (Note that $c_{n,s'} > \max_{j \leq n} r(j)$ since $s' > s$.) Hence, $\mathbf{N}_e$ is never injured after stage s'. Suppose $\Phi_e^A(e)[t'] \downarrow$ at some stage $t' > s'$. Then $r(e,t) = r(e,t')$ for all $t \geq t'$ so $A[t]$ never changes below $r(e,t')$ and $\Phi_e^A(e) \downarrow$. $\square$

Lemma 2.7. *There is a computable function $q(e)$ such that, for all $i \in \omega$, requirement $\mathbf{P}_e$ is injured at most $q(e)$ many times. Each requirement $\mathbf{P}_e$ acts finitely often and is eventually satisfied. Moreover, A is ω-c.e.*

Proof. We inductively give a computable upper bound $w(e)$ on the number of times the restraint $r(e,s)$ increases as well as a computable upper bound $d(e)$ on the number of times $\mathbf{P}_e$ redefines the dynamic index $c_{e,s}$. By construction, $\mathbf{P}_e$ is injured at most $\sum_{n=0}^{e} w(n)$ many times, so we set $q(e)$ to be this sum.

We now define $w(e)$ and $d(e)$. Since $\mathbf{N}_0$ is never injured once it sets a restraint, we set $w(0) = 1$. For any $e \in \omega$, requirement $\mathbf{P}_e$ changes the index $c_{e,s}$ only when $l_{e,s}$ changes or when injured by some higher priority $\mathbf{N}_n$. Thus, we set $d(e) = l_e + \sum_{n=0}^{e} w(n)$. Finally, we consider when $r(e,s)$ can increase. Any $\mathbf{P}_n$ with $n < e$ can injure $\mathbf{N}_e$ by changing the approximation of whether $c_{n,s} \in A$. The restraint $r(e,s)$ can increase before and after each such injury. Consider a fixed collection of indices $\{c_{n,s}\}_{n<e}$. By construction, the approximation of A below $\max\{c_{n,s}\}_{n<e}$ only changes on these indices. There are 2^e many configurations of the values of A at these indices. Since $r(e,s)$ is increasing, $\mathbf{N}_e$ is injured at most 2^e many times, for each fixed collection of indices $\{c_{n,s}\}_{n<e}$. Hence, $w(e) := 2^{e+1} \prod_{n=0}^{e} d(n)$ serves as the desired upper bound. This completes the definitions of $q(e), d(e)$, and $w(e)$ and the proof of the first claim.

Now suppose all $\mathbf{P}_n$ with $n < e$ act finitely often and are eventually satisfied. Choose some s such that no $\mathbf{P}_n$ with $n < e$ nor $\mathbf{N}_n$ with $n \leq e$ ever acts after stage s, $r(n,s) = \lim_{t\to\infty} r(n,t)$ for all $n \leq e$, and $\mathbf{P}_e$ is not injured after stage s. We may further choose s large enough so that $\emptyset^{2b}(e)[s] = \emptyset^{2b}(e)[t]$ for all $t \geq s$. By the choice of s, we also have that $v_{e,s}$, $c_{e,s}$, and $\alpha^b_{e,s}$ never change after stage s. Moreover, $\Psi^{\alpha^b_{e,s}}(e) := \alpha^b_{e,s}(v_{e,s}) = \emptyset^{2b}(e)[s]$, which equals $\emptyset^{2b}(e)$. Thus, $\Psi^{A^b \upharpoonright f(e)}(e)\downarrow = \emptyset^{2b}(e)$. (Recall that $f(e)$ is larger than any element in the A^b block for $\mathbf{P}_e$.)

Finally, observe that we always redefine $c_{n,s}$ whenever $l_{n,s}$ changes and choose $c_{n,s}$ larger than $k_{n,s}$ (where $\beta_{n,s} = \omega \cdot l_{n,s} + k_{n,s}$). For a fixed index $c_{n,s}$, we redefine the approximation of whether $c_{n,s} \in A[s]$ only when $\beta_{n,s}$, and hence $k_{n,s}$, decrease. Thus, $A(c_{n,s})[s]$ changes at most $k_{n,s} < c_{n,s}$ times. So, A is ω-c.e. □

□

3 The Jump Theorem

The Jump Theorem states in part that for any sets A and B we have $A \leq_T B \Leftrightarrow A' \leq_1 B'$. In Theorem 3.1, we prove the forward direction of the bounded jump analogue of the Jump Theorem. In particular, for all sets A and B, we have $A \leq_{bT} B$ implies $A^b \leq_1 B^b$. We give a counterexample to the converse in Theorem 3.2. Anderson and Csima [1] showed the forward direction holds for the b_0 jump, which is defined, for any set A, as

$$A^{b_0} = \{\langle e,i,j\rangle \in \omega \mid \varphi_i(j)\downarrow \; \wedge \Phi_e^{A \upharpoonright \varphi_i(j)}(j)\downarrow\}.$$

The converse to that result is also false. It follows from work by Downey and Greenberg [6] that there are sets A and B such that $A^{b_0} \leq_1 B^{b_0}$ but $A \not\leq_{bT} B$. (Note that Downey and

Greenberg use the notation $A^{\dagger}$ for a jump operator they define that satisfies $A^{\dagger} \equiv_1 A^{b_0}$.) The results for the bounded jump and the b_0 jump do not follow from one another; although $A^{b_0} \leq_1 A^b$ and $A^b \leq_{tt} A^{b_0}$, we do not in general have $A^b \equiv_1 A^{b_0}$ (see [1]). It remains an open question whether one can construct c.e. examples for Downey and Greenberg's result.

Theorem 3.1. *Let A and B be sets such that $A \leq_{bT} B$. Then $A^b \leq_1 B^b$.*

Proof. Let Γ and g witness that $A \leq_{bT} B$. Let j be an injective computable function defined by $\varphi_{j(x,i)}(z) = g(\varphi_i(x))$ (z is a dummy variable). Let f be a computable injective function defined by

$$\Phi^Y_{f(x)}(z)\downarrow \Leftrightarrow \big(\exists i \leq x\big)\big[\varphi_i(x)\downarrow \ \wedge \Phi_x^{\Gamma^{Y \upharpoonright g(\varphi_i(x))} \upharpoonright \varphi_i(x)}(x)\downarrow\big].$$

By the padding lemma, we may assume $f(x) \geq j(x,i)$ for all x and all $i \leq x$. We will show that $x \in A^b \Leftrightarrow f(x) \in B^b$ to prove the theorem.

For the forward direction we apply definitions and substitute $n = j(x,i)$.

$$x \in A^b \Leftrightarrow \big(\exists i \leq x\big)\big[\varphi_i(x)\downarrow \ \wedge \Phi_x^{A \upharpoonright \varphi_i(x)}(x)\downarrow\big] \Leftrightarrow$$

$$\big(\exists i \leq x\big)\big[\varphi_i(x)\downarrow \ \wedge \Phi_x^{(\Gamma^{B \upharpoonright g(\varphi_i(x))}) \upharpoonright \varphi_i(x)}(x)\downarrow\big] \Leftrightarrow \Phi_{f(x)}^{B \upharpoonright g(\varphi_i(x))}(f(x))\downarrow \Leftrightarrow$$

$$\varphi_{j(x,i)}(f(x))\downarrow \ \wedge \Phi_{f(x)}^{B \upharpoonright \varphi_{j(x,i)}(f(x))}(f(x))\downarrow \Rightarrow$$

$$\big(\exists n \leq f(x)\big)\big[\varphi_n(f(x))\downarrow \ \wedge \Phi_{f(x)}^{B \upharpoonright \varphi_n(f(x))}(f(x))\downarrow\big] \Leftrightarrow f(x) \in B^b$$

For the backwards direction, we ignore the original witness that $f(x) \in B^b$ and apply the use principle.

$$f(x) \in B^b \Leftrightarrow \big(\exists m \leq f(x)\big)\big[\varphi_m(f(x))\downarrow \ \wedge \Phi_{f(x)}^{B \upharpoonright \varphi_m(f(x))}(f(x))\downarrow\big] \Leftrightarrow$$

$$\big(\exists m \leq f(x)\big)\big[\varphi_m(f(x))\downarrow \ \wedge \big(\exists i \leq x\big)\big[\varphi_i(x)\downarrow \ \wedge \Phi_x^{(\Gamma^{(B \upharpoonright \varphi_m(f(x))) \upharpoonright g(\varphi_i(x))}) \upharpoonright \varphi_i(x)}(x)\downarrow\big]\big] \Rightarrow$$

$$\big(\exists i \leq x\big)\big[\varphi_i(x)\downarrow \ \wedge \Phi_x^{(\Gamma^{B \upharpoonright g(\varphi_i(x))}) \upharpoonright \varphi_i(x)}(x)\downarrow\big] \Leftrightarrow$$

$$\big(\exists i \leq x\big)\big[\varphi_i(x)\downarrow \ \wedge \Phi_x^{A \upharpoonright \varphi_i(x)}(x)\downarrow\big] \Leftrightarrow x \in A^b$$

Thus, f witnesses $A^b \leq_1 B^b$. □

Theorem 3.2. *There are c.e. sets A and B such that $A^b \leq_1 B^b$ and $A \not\leq_{bT} B$.*

Proof. Let g and h be injective computable functions such that for all x and all $k \leq 6^x$ we will control $\Phi_{g(x)}$ and $\varphi_{h(x,k)}$ and we have $h(x,k) < g(x)$.

We construct c.e. sets A and B that satisfy the following requirements:

$\mathbf{R}_{\langle e,i \rangle}$: $\big(\exists x\big)\big[A(x) \neq \Phi_e^{B \upharpoonright \varphi_i(x)}(x) \ \vee \ \varphi_i(x)\uparrow \ \vee \Phi_e^{B \upharpoonright \varphi_i(x)}(x)\uparrow\big]$

$\mathbf{Q}_n$: $n \in A^b \leftrightarrow g(n) \in B^b$

It is clear that these requirements suffice to prove the theorem. We build the sets A and B via a finite injury construction where the requirements are given priority $\mathbf{R}_0$, $\mathbf{Q}_0$, $\mathbf{R}_1$, $\mathbf{Q}_1$, $\mathbf{R}_2$, $\mathbf{Q}_2$, ... We will use a movable marker $x_{e,i}$ that eventually settles on the witness for $\mathbf{R}_{\langle e,i\rangle}$. We use the indices $k_n \leq 6^n$ for bookkeeping, and z represents a dummy variable throughout the construction. We will assume the convention that for any stage s and any c,d if $\varphi_c(d)[s]\downarrow = t$ then $t,c,d \leq s$.

To satisfy $\mathbf{R}_{\langle e,i\rangle}$, we will wait until we see $\varphi_i(x_{e,i})\downarrow$ and $\Phi_e^{B \upharpoonright\upharpoonright \varphi_i(x_{e,i})}(x_{e,i})\downarrow = 0$ (if this never happens, we are done). We then enumerate $x_{e,i}$ into A and attempt to preserve the now incorrect computation by fixing $x_{e,i}$ and $B \upharpoonright\upharpoonright \varphi_i(x_{e,i})$. If this restraint is injured by a stronger requirement, we choose a new $x_{e,i}$ and start over.

To satisfy $\mathbf{Q}_n$, we will wait until we see n enter our current estimate for A^b. We then define $\varphi_{h(n,k_n)}(z)$ to be a fresh large number (the same number for all z) and set $\Phi_{g(n)}^{B \upharpoonright\upharpoonright \varphi_{h(n,k_n)}(z)}(z)$ to converge for all z so that $g(n)$ is now in B^b. We try to fix enough of A and B to preserve the statements $n \in A^b$ and $g(n) \in B^b$. If either statement is injured, we enumerate $\varphi_{h(n,k_n)}(z)$ into B so that our earlier declaration that $g(n)$ is in B^b no longer applies. We then increment k_n by one and start over.

We note that the computable function g is not allowed to have any errors. Hence, we may be required to void declarations placing $g(n)$ into B^b by enumerating new elements into B, even if doing so injures stronger requirements. To prevent such injuries, we act preemptively. As soon as we see $\varphi_i(x_{e,i})$ converge (but before we place any restraint on B for $\mathbf{R}_{\langle e,i\rangle}$), we immediately void any active declarations from weaker requirements. To simplify the construction, whenever we act for a requirement, we will injure all weaker requirements, regardless of whether such an injury appears necessary. For brevity, we call this process the *injury procedure*, and we describe it at the end of the construction.

We start with $x_{e,i}$ as the least number in the $\langle e,i\rangle$ column and all $k_n = 0$. We set $A(0) = 0$ and $B(0) = 1$ and never change those values (this ensures our declarations for B^b do not change A^b). No other elements begin in A and B.

At each stage s we perform the following procedures:

Preemptive injury of $\mathbf{R}_{\langle e,i\rangle}$: If some convergence $\varphi_i(x_{e,i})[s]\downarrow$ has just occurred, then we carry out the injury procedure.

Diagonalization for $\mathbf{R}_{\langle e,i\rangle}$: If we observe that $\Phi_e^{B \upharpoonright\upharpoonright \varphi_i(x_{e,i})}(x_{e,i})[s]\downarrow = 0$ holds, then we enumerate $x_{e,i}$ into A and execute the injury procedure.

$\mathbf{Q}_n$ Strategy : If $n \in A^b[s]$ and $\mathbf{Q}_n$ has not acted (since its last injury), we first run the injury procedure. We then set $\varphi_{h(n,k_n)}(z)$ (for all z) to be the least unused number greater than s, and we declare that $\Phi_{g(n)}^{B \upharpoonright\upharpoonright \varphi_{h(n,k_n)}(z)}(z)\downarrow$ for all z.

We describe the injury procedure on all requirements weaker than the one under consideration:

- For all weaker $\mathbf{R}_{\langle e,i\rangle}$ requirements with $x_{e,i} \leq s$, we assign a new unused value greater than s to $x_{e,i}$.

- For all weaker $\mathbf{Q}_n$ requirements, if $\varphi_{h(n,k_n)}(z)$ has been defined, we enumerate $\varphi_{h(n,k_n)}(z)$ into B and increment k_n by one.

This completes the construction. It is clear that A and B are c.e. The usual argument that the n^{th} requirement in a standard finite injury argument acts at most 2^n times can be routinely altered to show that every $\mathbf{Q}_n$ acts at most 6^n many times (we multiply by an additional 3^n since the requirement $\mathbf{R}_{\langle e,i\rangle}$ between $\mathbf{Q}_m$ and $\mathbf{Q}_{m+1}$ may act twice). Hence, we have sufficiently many possible values for k_n to run the construction.

We show $\mathbf{R}_{\langle e,i\rangle}$ is satisfied, as is $\mathbf{Q}_n$. Suppose that s_0 is the last stage at which any requirement of higher priority than $\mathbf{R}_{\langle e,i\rangle}$ acts or is injured. Let $x_{e,i}$ now denote the final value of the marker $x_{e,i}$, that is, its value at stage s_0. If $\varphi_i(x_{e,i})$ diverges or $\Phi_e^{B \upharpoonright \varphi_i(x_{e,i})}(x_{e,i})$ either diverges or equals one, then we are done as $x_{e,i} \notin A$. Otherwise, we preemptively injure $\mathbf{R}_{\langle e,i\rangle}$ at some stage $s > s_0$ and diagonalize on behalf of $\mathbf{R}_{\langle e,i\rangle}$ at some stage $t > s$. At stage s, the construction ensures that $\varphi_{h(n,k_n)}(z) > \varphi_i(x_{e,i})$ for all $\mathbf{Q}_n$ of lower priority than $\mathbf{R}_{\langle e,i\rangle}$. As a result, for all $s_1 > s$ we have $B_{s_1} \upharpoonright \varphi_i(x_{e,i}) = B \upharpoonright \varphi_i(x_{e,i})$. Thus,

$$\Phi_e^{B \upharpoonright \varphi_i(x_{e,i})}(x_{e,i}) = \Phi_e^{B_t \upharpoonright \varphi_i(x_{e,i})}(x_{e,i}) = 0 \neq 1 = A(x_{e,i})$$

so $\mathbf{R}_{\langle e,i\rangle}$ is satisfied.

Suppose that s_0 is the last stage at which any requirement of higher priority than $\mathbf{Q}_n$ acts or is injured, so k_n and $h(n,k_n)$ are fixed for all $s > s_0$. By construction, $\Phi_{g(n)}^{B \upharpoonright s_0}\uparrow$ so $g(n) \notin B^b[s_0]$. We show $n \in A^b \Leftrightarrow g(n) \in B^b$. We consider two cases. First, suppose that $\mathbf{Q}_n$ never acts at any stage $s > s_0$. Then, $n \notin A^b[s]$ for any $s > s_0$, so $n \notin A^b$. Also, no new convergence declarations are made about $\Phi_{g(n)}^{\sigma}$ for any string $\sigma \in 2^{<\omega}$ in B^b after stage s_0 (and all strings σ for which there are existing convergence declarations are incomparable with B_{s_0}). All lower priority $\mathbf{Q}_m$ requirements satisfy $\varphi_{h(m,k_m)}(z) > s_0$, so $B \upharpoonright s_0[s_0] = B \upharpoonright s_0$. Thus, $g(n) \notin B^b$.

Second, suppose that $\mathbf{Q}_n$ acts at some stage $s > s_0$. Then, $n \in A^b[s]$ and $g(n) \in B^b[s]$. At stage s, the injury procedure ensures that $x_{e,i} > s$ for all lower priority $\mathbf{R}_{\langle e,i\rangle}$ requirements and $\varphi_{h(m,k_m)}(z) > s$ for all lower priority $\mathbf{Q}_m$ requirements. Hence, $A \upharpoonright s[s] = A \upharpoonright s$ and $B \upharpoonright s[s] = B \upharpoonright s$. Thus, $n \in A^b$ and $g(n) \in B^b$.

We conclude $n \in A^b \Leftrightarrow g(n) \in B^b$ so $\mathbf{Q}_n$ is satisfied, completing our proof. □

4 Questions

Here we provided examples of extreme kinds of bounded low and bounded high sets. It is natural to ask what other kinds of examples exist. We constructed a c.e. high bounded low set in Theorem 2.1, but we only constructed an ω-c.e. low bounded high set in Theorem 2.5. We conjecture that there is no c.e. example of a low set that is bounded high.

Question 4.1. *Does there exist a c.e. set that is both low and bounded high?*

A set $A \leq_T \emptyset'$ is called *superlow* if $A' \leq_{tt} \emptyset'$ and *superhigh* if $A' \geq_{tt} \emptyset''$. Mohrherr [11] constructed an example of an incomplete superhigh set. We have a low set that is bounded high, which is clearly bounded high and not high. Similarly, we have an example of a bounded low set that is not low.

Question 4.2. *Does there exist a bounded low set that is low but not superlow? Does there exists a high and bounded high set that is not superhigh?*

Observe that any superlow set is bounded low since $A^b \leq_1 A'$. However, we can ask:

Question 4.3. *Does there exist a superhigh set that is not bounded high?*

Recall that A is bounded low iff A^b is ω-c.e. Could we provide a better characterization?

Question 4.4. *Provide characterizations of bounded low and bounded high sets.*

It is also natural to consider these questions for other restricted computation jump operators, such as Gerla's operator [8], which is discussed in [1].

Acknowledgements

The second author was partially supported by Canadian NSERC Discovery Grant 312501. The third author was partially supported by National Science Foundation grants DMS-0802961 and DMS-1100604 and a faculty award from Wellesley College.

References

[1] B. Anderson and B. Csima. A bounded jump for the bounded Turing degrees. *Notre Dame J. Formal Logic*, 55(2):245–264, 2014.

[2] B. A. Anderson. Automorphisms of the truth-table degrees are fixed on a cone. *J. Symbolic Logic*, 74(2):679–688, 2009.

[3] C. J. Ash and J. Knight. *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 2000.

[4] B. Cooper. *Computability Theory*. Chapman and Hall/CRC, 2004.

[5] B. Csima, R. Downey, and K. M. Ng. Limits on jump inversion for strong reducibilities. *J. Symbolic Logic*, 76(4):1287–1296, 2011.

[6] R. Downey and N. Greenberg. A transfinite hierarchy of notions of lowness in the computably enumerable degrees, unifying classes, and natural definability. Preprint.

[7] R. L. Epstein, R. Haas, and R. L. Kramer. Hierarchies of sets and degrees below $\mathbf{0}'$. In *Logic Year 1979–80 (Proc. Seminars and Conf. Math. Logic, Univ. Connecticut, Storrs, Conn., 1979/80)*, volume 859 of *Lecture Notes in Math.*, pages 32–48. Springer, Berlin, 1981.

[8] G. Gerla. Una generalizzazione della gerarchia di Ershov. *Bolletino U.M.I.*, 16-B(5):765–778, 1979.

[9] C. G. Jockusch, Jr. and R. A. Shore. Pseudo-jump operators II: Transfinite iterations, hierarchies, and minimal covers. *J. Symbolic Logic*, 49:1205–1236, 1984.

[10] J. Mohrherr. Density of a final segment of the truth-table degrees. *Pacific J. Math.*, 115(2):409–419, 1984.

[11] J. Mohrherr. A refinement of low *n* and high *n* for the r.e. degrees. *Z. Math. Logik Grundlag. Math.*, 32(1):5–12, 1986.

[12] R. I. Soare. *Recursively Enumerable Sets and Degrees*. Springer-Verlag, 1987.

[13] R. I. Soare. *Computability Theory and Applications*. Springer-Verlag, preprint.