PROCEEDINGS OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 136, Number 7, July 2008, Pages 2589–2595
S 0002-9939(08)09248-4
Article electronically published on March 11, 2008

ON THE TRIPLE JUMP OF THE SET OF ATOMS OF A BOOLEAN ALGEBRA

ANTONIO MONTALBÁN

(Communicated by Julia Knight)

Abstract. We prove the following result concerning the degree spectrum of the atom relation on a computable Boolean algebra. Let $\mathcal{C}$ be a computable Boolean algebra with infinitely many atoms and $\mathbf{a}$ be the Turing degree of the atom relation of $\mathcal{C}$. If $\mathbf{d}$ is a c.e. degree such that $\mathbf{a}''' \leq_T \mathbf{d}'''$, then there is a computable copy of $\mathcal{C}$ where the atom relation has degree $\mathbf{d}$. In particular, for every high$_3$ c.e. degree $\mathbf{d}$, any computable Boolean algebra with infinitely many atoms has a computable copy where the atom relation has degree $\mathbf{d}$.

1. Introduction

We study the degree spectrum of the atom relation on computable Boolean algebras. If $\mathcal{C}$ is a computable Boolean algebra, let

$$Dg_{\mathcal{C}}(Atom) = \{\deg(Atom(\mathcal{B})) : \mathcal{B} \cong \mathcal{C}, \mathcal{B} \text{ computable}\},$$

where $\deg(X)$ is the Turing degree of X, and $Atom(\mathcal{B})$ is the set of atoms of $\mathcal{B}$, that is, the set of nonzero $x \in \mathcal{B}$ beneath which there is no element other than 0. This is part of a more general program that studies the information content of relations on computable models, initiated by Ash and Nerode [AN81]. One of the goals of this program is to understand the possible degree spectra that relations can have, or in other words, how the classical isomorphism type of a computable model restricts the computability theoretic properties of a certain relation. In the case of Boolean algebras, the atom relation is the simplest one. It is natural to ask how its degree spectrum looks, or what information can be encoded into it and how hard it is to decode it. A nice survey on this question can be found in [Rem89]. Let us fix a computable Boolean algebra $\mathcal{B}$. If $\mathcal{B}$ has only finitely many atoms, then $Dg_{\mathcal{B}}(Atom) = \{\mathbf{0}\}$. So, let us assume $\mathcal{B}$ has infinitely many atoms. The first observation is that the set of atoms is co-c.e., and hence $Dg_{\mathcal{B}}(Atom)$ is contained in the c.e. degrees. It follows from [Gon75, Theorem 2] that there is a computable Boolean Algebra $\mathcal{B}$ such that $\mathbf{0} \notin Dg_{\mathcal{B}}(Atom)$ (see [Rem89], or [Dow97, Theorem 5.11] adding the word "atomic" in the line where Tarski's result is mentioned). Remmel [Rem81a, 2.12.i] showed that $Dg_{\mathcal{B}}(Atom)$ is closed upwards in

Received by the editors December 8, 2006, and, in revised form, April 12, 2007, April 22, 2007, and May 31, 2007.

2000 *Mathematics Subject Classification.* Primary 03D80.

Key words and phrases. Boolean algebra, atom, relation, degree spectrum.

This research was partially supported by NSF Grant DMS-0600824 and by the Marsden Foundation of New Zealand, via a postdoctoral fellowship.

the c.e. degrees, and, in particular, that it always contains $\mathbf{0}'$. Downey [Dow93] then showed that $Dg_{\mathcal{B}}(Atom)$ also has to contain some incomplete degree. This contrasts a result of Downey and Moses [DM91], who proved that there is a computable linear ordering whose set of successivities is intrinsically complete. We prove the following extension of Downey's result.

Theorem 1.1. *Let $\mathcal{C}$ be a computable Boolean algebra with infinitely many atoms. If $\mathbf{a} \in Dg_{\mathcal{C}}(Atom)$ and $\mathbf{d}$ is a c.e. degree such that $\mathbf{a}''' \leq_T \mathbf{d}'''$, then $\mathbf{d} \in Dg_{\mathcal{C}}(Atom)$. In particular, for every high$_3$ c.e. degree $\mathbf{d}$, $\mathbf{d} \in Dg_{\mathcal{C}}(Atom)$.*

It follows that no information can be encoded directly into the atom relation of a Boolean algebra $\mathcal{C}$: If $\mathbf{b}$ is a noncomputable Turing degree, there is a high degree $\mathbf{d}$ which does not compute $\mathbf{b}$ and there is a copy of $\mathcal{C}$ where the atom relation is computable in $\mathbf{d}$ and hence does not compute $\mathbf{b}$. Moreover, via a similar argument, we get that no information can be encoded into the double jump of the atom relation. Another corollary we get is that there is a computable Boolean algebra $\mathcal{B}$ such that $Dg_{\mathcal{B}}(Atom)$ contains no low$_3$ degree: Just consider the Boolean algebra constructed by Goncharov [Gon75] whose atom relation is always noncomputable.

It was proved by Knight and Stob [KS00] that every low$_4$ Boolean algebra has a computable copy. Our theorem has a similar flavor, and it heavily uses results of Knight and Stob [KS00]. It is not known whether every low$_n$ Boolean algebra has a computable copy. A proof answering this question positively would probably yield a generalization of our main theorem to nth-jumps, and in particular that every high$_n$ c.e. degree is in the degree spectrum of the atom relation of any computable Boolean algebra with infinitely many atoms.

Even though the result is a big improvement over Downey's previous result, the proof is not complicated. It is actually an application of a sequence of known lemmas due to Knight and Stob [KS00], Thurber [Thu95], and Remmel [Rem81a], and a variation of another lemma due to Downey and Jockusch [DJ94]. Here is the new lemma.

Lemma 1.2. *Let $\mathbf{d}$ be a c.e. degree and let $\mathcal{A}$ be a $\mathbf{d}$-computable Boolean Algebra whose atom relation is also computable in $\mathbf{d}$. Then, there is a computable copy of $\mathcal{A}$ whose atom relation is computable from $\mathbf{d}$.*

This lemma extends a previous result of Downey and Jockusch [DJ94], where they prove that if $\mathcal{A}$ is a $\mathbf{0}'$-computable Boolean algebra $\mathcal{A}$ whose atom relation is also computable in $\mathbf{0}'$, then $\mathcal{A}$ is isomorphic to a computable Boolean algebra. Their proof has two steps, first a finite injury argument and then an application of the Remmel-Vaught Lemma [Rem81a] (see Lemma 3.1 below). Our proof follows the one found in [KS00, Theorem 2.2], although, since we also need to keep the atom relation below a certain degree, we have to be a bit more careful.

It might be possible to get similar results for relations other than the atom relation, as for example, for the relations "finite" (or equivalently infinite) and "atomless" (see definitions below). Ideas similar to ours might be applicable for studying the following spectrum:

$$Dg_{\mathcal{C}}(Atom, Finite, Atomless) =$$
$$\{\langle \deg(Atom(\mathcal{B})), \deg(Finite(\mathcal{B})), \deg(Atomless(\mathcal{B}))\rangle : \mathcal{B} \cong \mathcal{C}, \mathcal{B} \text{ computable}\},$$

Remmel [Rem81b] has studied the degree spectrum of the relation "finite" on computable Boolean algebras with computable sets of atoms.

2. General outline of the proof

First we describe how Lemma 1.2 is used to prove our main theorem. We need to start by introducing some Boolean algebra unary predicates. Atom, infinite and atomless are the only predicates we will use in the proof of Lemma 1.2. We define the other ones to be able to refer to previous results of Knight and Stob [KS00]. We classify these predicates in terms of how many quantifier alternations, in a computably infinitary formula, are necessary to define them.

- 1-relations
  - atom(x) - x is an atom,
- 2-relations
  - atomless(x) - $x \neq 0$, and there are no atoms below x,
  - infinite(x) - x is not a join of finitely many atoms,
- 3-relations
  - atomic(x) - x has no atomless elements below it,
  - 1-atom(x) - whenever $x = y \vee z$, either y or z is a finite join of atoms,
  - atominf(x) - there are infinitely many atoms below x,
- 4-relations
  - $\sim$-inf(x) - there are infinitely many $\sim$-inequivalent elements below x, where $a \sim b$ if the symmetric difference $a \triangle b = (a-b) \vee (b-a)$ is finite,
  - infatomicless(x) - there is no infinite atomic element below x,
  - 1-atomless(x) - there are no 1-atoms below x,
  - nomaxatomless(x) - x is not a join of atomless and atomic elements,
  - $I(\omega+\eta)(x)$ - atominf(x) and whenever $x = y \vee z$, there are only finitely many atoms below either y or z.

  When we refer to $\leq$4-relations, we mean the set of 1-, 2-, 3- and 4-relations. Similarly for $\leq$3-relations and $\leq$2-relations.

Let $\mathcal{C}_0$ be a computable Boolean algebra with infinitely many atoms whose atom relation is computable in $\mathbf{a}$. The first important observation is that all the ≤ 4-relations of $\mathcal{C}_0$ are computable in $\mathbf{a}''' \leq_T \mathbf{d}'''$: just write down the definitions in terms of the atom relation. We use the following sequence of lemmas.

Lemma 2.1 (Knight and Stob [KS00, 5.2]). *If a Boolean algebra $\mathcal{C}$ is computable in $\mathbf{x}'$ and all its $\leq$4-relations are also computable in $\mathbf{x}'$, then $\mathcal{C}$ has an $\mathbf{x}$-computable copy where all the $\leq$3-relations are computable in $\mathbf{x}$.*

So we have that there is a $\mathbf{d}''$-computable copy $\mathcal{C}_1$ of $\mathcal{C}_0$ whose $\leq$ 3-relations are also computable in $\mathbf{d}''$.

Lemma 2.2 (Knight and Stob [KS00, 3.2]). *If a Boolean algebra $\mathcal{C}$ is computable in $\mathbf{x}'$ and all its $\leq$3-relations are also computable in $\mathbf{x}'$, then $\mathcal{C}$ has an $\mathbf{x}$-computable copy where all the $\leq$2-relations are computable in $\mathbf{x}$.*

So we have that there is a $\mathbf{d}'$-computable copy $\mathcal{C}_2$ of $\mathcal{C}_1$ whose $\leq$2-relations are also computable in $\mathbf{d}'$.

Lemma 2.3 (Thurber [Thu95]; see [KS00, 2.5]). *If a Boolean algebra $\mathcal{C}$ is computable in $\mathbf{x}'$ and all its $\leq$ 2-relations are also computable in $\mathbf{x}'$, then $\mathcal{C}$ has an $\mathbf{x}$-computable copy where all the 1-relations are also computable in $\mathbf{x}$.*

So we have that there is a **d**-computable copy $\mathcal{C}_3$ of $\mathcal{C}_2$ such that the atom relation is also computable in **d**. Now we use Lemma 1.2 and get a computable copy $\mathcal{C}_4$ of $\mathcal{C}_3$ whose atom relation is computable in **d**. To get a copy $\mathcal{C}_5$ of $\mathcal{C}_4$ where the atom relation has degree exactly **d**, we use the following lemma.

Lemma 2.4 (Remmel [Rem81a, 2.12])**.** *If a computable Boolean algebra $\mathcal{C}$ has infinitely many atoms and its atom relation is computable in a c.e. degree* **d***, then $\mathcal{C}$ has a computable copy whose atom relation has degree* **d***.*

3. Proof of Lemma 1.2

Let $\mathcal{A}$ be a **d**-computable Boolean algebra with infinitely many atoms whose atom relation is computable in **d**. We want to build a computable Boolean algebra $\mathcal{B}$, isomorphic to $\mathcal{A}$, whose atom relation is also computable in **d**. Let D be an infinite c.e. set in **d**. We define the set of true stages for the enumeration of D as usual. Let $\{d_0, d_1, ...\}$ be a computable enumeration of D. We say that t is a *true stage* if $\forall u > t$ $(d_u > d_t)$. Let $t_0 < t_1 < t_2 < ...$ be the sequence of true stages. This sequence is computable in D. A stage t *looks true at* s if $t \leq s$ and $\forall u \leq s$ $(u > t \Rightarrow d_u > d_t)$; this is a computable predicate. Note that if t is a true stage, then t looks true at any stage $s > t$, and if $u < t$ looks true at t, then u is actually true. For every s, we let t^s be the largest $t < s$ which looks true at s. To make t^s defined for every $s > 0$, we might assume that $d_0 = 0$ and hence that 0 is a true stage.

We construct $\mathcal{B}$ by finite approximations. At stage s we computably define a finite Boolean algebra $\mathcal{B}_s$ so that $\mathcal{B}_0 \subseteq \mathcal{B}_1 \subseteq \mathcal{B}_2 \subseteq ...$ and $\mathcal{B} = \bigcup_{s\in\omega} \mathcal{B}_s$. We define $\mathcal{B}_s$ so that $\bigcup B_s = \omega$, where B_s is the domain of $\mathcal{B}_s$. Note that $\mathcal{B}$ is computable. If y is a minimal element of $\mathcal{B}_{s-1}$ and $x \in \mathcal{B}_s$ is strictly below y, $x \neq 0$, we say that x is *enumerated at* s and that y *is split at* s. Let C be the set of atoms of $\mathcal{B}$. So, the elements of C will appear in some $\mathcal{B}_s$ as minimal elements and will never split. To make sure C is computable in D we satisfy the following condition:

(B) If $y \in B_{t^s-1}$ is a minimal element of $\mathcal{B}_{s-1}$, then it does not split at s.

Therefore, if t^s is a true stage, then y will be an atom of $\mathcal{B}$. To compute C from D we do the following. Given $x \in B$ we find the first stage s at which it is enumerated. Then we find some true stage t_k greater than s. So, x is an atom of $\mathcal{B}$ if and only if it is a minimal element in $\mathcal{B}_{t_k}$.

Now we describe our approximation to $\mathcal{A}$. We think of $\mathcal{A}$ as a Boolean algebra with atom relation; that is, a structure $\langle A, \vee, \wedge, \neg, Atom(\cdot)\rangle$. At stage s we will define a finite Boolean algebra $\mathcal{A}_s = \langle A_s, \vee, \wedge, \neg, T_s\rangle$, where $T_s \subseteq A_s$ is a subset of the set of minimal elements of $\mathcal{A}_s$, but since minimal elements of $\mathcal{A}_s$ might not be atoms of $\mathcal{A}$, T_s might not contain all the minimal elements of $\mathcal{A}_s$. There is some Turing functional, that with oracle D, computes $\mathcal{A}$, including the atom relation. We let $\hat{\mathcal{A}}_s = \langle \hat{A}_s, \vee, \wedge, \neg, \hat{T}_s\rangle$ be the largest finite Boolean algebra that this Turing functional computes in less than s steps and using as oracle the finite string $D_s \restriction d_s = \langle D_s(0), D_s(1), ..., D_s(d_s-1)\rangle$. Note that if t looks true at s, then $\hat{\mathcal{A}}_t \subseteq \hat{\mathcal{A}}_s$ where inclusion here also means that $\hat{T}_t = \hat{T}_s \cap A_t$. Therefore $\mathcal{A} = \bigcup_{n\in\omega} \hat{\mathcal{A}}_{t_n}$, where $\{t_n : n \in \omega\}$ is the sequence of true stages. We would like to assume that for each n, either $\mathcal{A}_{t_{n+1}} = \mathcal{A}_{t_n}$ or $\mathcal{A}_{t_{n+1}} = \mathcal{A}_{t_n}[a_{t_{n+1}}]$, that is, the Boolean algebra generated by $\mathcal{A}_{t_n}$ and $a_{t_{n+1}}$, where $a_{t_{n+1}}$ is strictly below some minimal element $b_{t_{n+1}}$ of $\mathcal{A}_{t_n}$. We define $\mathcal{A}_s$ by induction as follows. Let $\mathcal{A}_0$ be the Boolean algebra with two elements. To define $\mathcal{A}_s$, consider $\mathcal{A}_{t^s}$. If $\mathcal{A}_{t^s} = \hat{\mathcal{A}}_s$, let $\mathcal{A}_s = \hat{\mathcal{A}}_s$, and if $\mathcal{A}_{t^s} \subsetneq \hat{\mathcal{A}}_s$, let a_s be some nonzero element of $\hat{\mathcal{A}}_s$ strictly below some minimal element in $\mathcal{A}_{t^s}$, and let $\mathcal{A}_s$ be the subalgebra of $\hat{\mathcal{A}}_s$ generated by $\mathcal{A}_{t^s}$ and a_s. We let $T_s = \hat{T}_s \cap \mathcal{A}_s$. It is not hard to show that one can choose a_s so that $\mathcal{A} = \bigcup_{n\in\omega} \mathcal{A}_{t_n}$.

To get $\mathcal{A} \cong \mathcal{B}$, we will construct an embedding $f\colon \mathcal{A} \to \mathcal{B}$ by stages. At each stage s we define

$$f_s\colon \mathcal{A}_s \to \mathcal{B}_s.$$

To be able to define f as some sort of limit of the f_s, we impose the following condition:

(F) $\forall c \in \mathcal{A}_{t^s}$ $(f_s(c) = f_{t^s}(c))$.

So we have that $f_{t_0} \subseteq f_{t_1} \subseteq f_{t_2} \subseteq ....$ We then define $f = \bigcup_{k\in\omega} f_{t_k}$. Note that to compute f we need an oracle for D, because D can enumerate the true stages.

At stage s we have f_s already defined on $\mathcal{A}_{t^s}$ and we need to extend it to $\mathcal{A}_s = \mathcal{A}_{t^s}[a_s]$ where a_s is strictly below some minimal element b_s of $\mathcal{A}_{t^s}$. We only need to decide the value of $f_s(a_s)$, and it has to be some element of $\mathcal{B}_s$ strictly below $f_s(b_s)$. Recall that we are allowed to enlarge $\mathcal{B}_{s-1}$, but we are not allowed to put elements below the minimal elements of $\mathcal{B}_{t^s-1}$ that are still minimal elements in $\mathcal{B}_{s-1}$. To make sure there is always room below $f_s(b)$, we inductively make sure the following conditions are satisfied *at* every stage s and *below* $f_s(d)$, for every minimal element d of $\mathcal{A}_s$.

(B1) If we guess d is an atom (i.e. $d \in T_s$), then no minimal element of $\mathcal{B}_{s-1}$ below $f_s(d)$ is split in $\mathcal{B}_s$.
(B2) Otherwise, every minimal element of $\mathcal{B}_{s-1}$ below $f_s(d)$ is split in $\mathcal{B}_s$.

Notation: For a Boolean algebra $\mathcal{C}$ and $c \in \mathcal{C}$, we let $\mathcal{C}\restriction c = \{x \in \mathcal{C} : x \leq c\}$. Observe that if $c \in \mathcal{A}$ is a finite sum of atoms, the conditions above might imply that $\mathcal{A}\restriction c \ncong \mathcal{B}\restriction f(c)$. However, these two restricted Boolean algebras will not be too different; both will be finite. The reason is the following. On the one hand, for every stage s at which some element $d \leq c$ is a minimal element of $\mathcal{A}_s$ but not an atom, we will have that $f(d)$ splits in $\mathcal{B}_s$ and hence the size of $\mathcal{B}_s\restriction f(c)$ grows. On the other hand, at some true stage t_0, we will have that all the atoms below c are in $\mathcal{A}_{t_0}$. For every stage $t \geq t_0$, since all the minimal elements d of $\mathcal{A}_t$ below c are atoms, by (B1), we have that no minimal element of $\mathcal{B}_{t-1}$ below $f_{t_0}(c)$ is split in $\mathcal{B}_t$. Therefore, $\mathcal{B}\restriction f(c) = \mathcal{B}_{t_0}\restriction f_{t_0}(c)$ is finite.

We would like $f\colon \mathcal{A} \to \mathcal{B}$ to be an isomorphism. However, we only need f to satisfy the conditions in the following lemma.

Lemma 3.1 (Remmel-Vaught [Rem81a, 2.1])**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be countable Boolean algebras, and suppose that $\mathcal{A}$ has infinitely many atoms. Let $f\colon \mathcal{A} \to \mathcal{B}$ be a Boolean algebra embedding such that*

(RV1) *if a is an atom of $\mathcal{A}$, then $f(a)$ is a finite join of atoms in $\mathcal{B}$,*
(RV2) *every atom of $\mathcal{B}$ is below $f(a)$ for some atom a of $\mathcal{A}$,*
(RV3) *$\mathcal{B}$ is generated by $f(\mathcal{A})$ and the atoms of $\mathcal{B}$.*

Then, $\mathcal{A}$ and $\mathcal{B}$ are isomorphic.

Note that condition (RV1) follows from (B1) because if we guess $d \in \mathcal{A}_t$ is an atom at some true stage t, then we guess d is an atom at every stage $s > t$.

We are now ready to do the construction.

Construction: Stage $s = 0$. Let $\mathcal{B}_0$ be the Boolean algebra with two elements, and f_0 be the identity map from $\mathcal{A}_0$ to $\mathcal{B}_0$.

Stage $s > 0$. We have $A_s = A_{t^s}[a_s]$, where $a = a_s$ is strictly below some minimal element $b = b_s$ of $\mathcal{A}_{t^s}$. We need to define f_s satisfying (F) and define $\mathcal{B}_s$ satisfying (B), (B1), (B2), and (RV1)–(RV3). Along with the construction, we verify that conditions (F), (B), (B1) and (B2) are satisfied. It then follows that (RV1) is satisfied. We will verify (RV2) and (RV3) after the construction. For this purpose, during the construction we will make sure that if s is true, then every element of $\mathcal{B}_{s-1} \restriction f_s(b_s)$ is generated by the image of f and the atoms of $\mathcal{B}$, and that every atom of $\mathcal{B}$ that belongs to $\mathcal{B}_{s-1} \restriction f_s(b_s)$ is strictly below $f(d)$ for some atom d of $\mathcal{A}$.

We define $f_s(a)$ and extend $\mathcal{B}_{s-1} \restriction f_s(b)$ to $\mathcal{B}_s \restriction f_s(b)$. We look at different cases depending on our guesses about a and $b-a$. Recall that we already know the value of $f_s(b) = f_{t^s}(b)$, and that at stage t^s we had to split some minimal element below $f_{t^s}(b)$, so $f_s(b)$ is not a minimal element of B_{s-1}.

- **Case 1:** We are guessing that at least one of a or $b - a$ is an atom (i.e. $a \in T_s$ or $b - a \in T_s$). Suppose first that a is an atom in $\mathcal{A}_s$. Let $x_0, ..., x_k$ be the minimal elements of $\mathcal{B}_{s-1} \restriction f_s(b)$. Define $f_s(a) = \bigvee_{i=1,...,k} x_i$ and $f_s(b-a) = x_0$. Notice that if s is true, the minimal elements of $\mathcal{B}_s \restriction f_s(a)$ will be atoms in $\mathcal{B}$, because we will always guess a is an atom. Moreover, every nonzero element of $\mathcal{B}_{s-1} \restriction f_s(b)$ is generated by the minimal elements of $\mathcal{B}_s \restriction f_s(a)$ and $f_s(b-a)$. So, all the elements of $\mathcal{B}_{s-1} \restriction f_s(b)$ satisfy (RV2) and (RV3). If a is not an atom in $\mathcal{A}_s$ but $b-a$ is, replace a by $b-a$ in the definition of f above.
- **Case 2:** Neither a nor $b - a$ is an atom. Since there might not be any atoms in $\mathcal{A}$ below b, we have to try to get that every element of $\mathcal{B} \restriction f(b)$ is in the image of f. Here is one way of not leaving anybody in $\mathcal{B} \restriction f(b)$ out of the image of f. Let $u < s$ be the least stage at which there is a nonzero element strictly below $f_s(b)$ in $\mathcal{B}_u$. Define $f_s(a)$ to be one of those elements in $\mathcal{B}_u \restriction f_s(b)$. If s is true and $\mathcal{B} \restriction f_s(b)$ is atomless, then eventually every element of $\mathcal{B}_u \restriction f_s(b)$ will be contained in the image of f, and only after this do we start letting $f \restriction b$ have values that have been enumerated into $\mathcal{B}$ after u.

Let us extend $\mathcal{B}_{s-1} \restriction f_s(d)$ to $\mathcal{B}_s \restriction f_s(d)$ for every minimal element d of $\mathcal{A}_s$, in order to satisfy (B1) and (B2) at s below $f_s(d)$. If we are currently guessing d is an atom, let $\mathcal{B}_s \restriction f_s(d) = \mathcal{B}_{s-1} \restriction f_s(d)$. Otherwise, since (B2) was satisfied at t^s, every minimal element in $B_{s-1} \restriction f_s(d)$ was enumerated after $t^s - 1$, so we can split them now. Of course, every time we split a minimal element $y \in \mathcal{B}_{s-1}$ into $x, y - x \in B_s$, we are also adding to $\mathcal{B}_s$ all the elements generated by $\mathcal{B}_{s-1}$ and x, namely the elements of the form $z \vee x$ or $z \vee (y - x)$ for $z \in \mathcal{B}_{s-1}$.

This finishes the construction of f and $\mathcal{B}$. Now we have to verify that (RV2) and (RV3) are satisfied: Note that every element of $\mathcal{B}$ is generated by the minimal elements of $\mathcal{B}_t$ for true stages t, and hence, for (RV3), it is enough to show that the minimal elements of the $\mathcal{B}_t$'s are generated by the image of f and the atoms of $\mathcal{B}$. For (RV2) we have to show that if a minimal element of $\mathcal{B}_t$ is an atom in $\mathcal{B}$, it is below the image of some atom in $\mathcal{A}$. Consider a true stage t and a minimal element e of $\mathcal{B}_t$. It is below $f_t(d)$ for some minimal element d of $\mathcal{A}_t$. So, at stage t, the only elements of $\mathcal{B}_t \restriction f_t(d)$ in the image of f_t are 0 and $f_t(d)$. Note that since t is true, $f_t(d) = f(d)$. Let $c \geq e$ be the minimal element of $\mathcal{B}_t$ that is in the image of f (at the end of time). Say $c = f_s(b)$ for some true stage $s \geq t$ and some $b \in \mathcal{A}$. Note that $c \leq f(d)$ and $b \leq d$. If $c = e$, then it is not hard to see that (RV2) and (RV3) are satisfied, so suppose not. (Because $e = f(b)$ is clearly in the image of f, and if e is an atom, b would have to be an atom of $\mathcal{A}$ too.) If b is an atom of $\mathcal{A}$, then e will be an atom of $\mathcal{B}$, and again, (RV2) and (RV3) would be satisfied, so suppose not. So we have that $c = f_s(b)$, where b is not an atom of $\mathcal{A}$ and s is a true stage, and we have that c is the minimal element in $\mathcal{B}_t$ above e. At some later true stage s_1, since b is not an atom, we consider a split of $b = b_s$ into a and $b-a$. If either of a or $b-a$ is an atom of $\mathcal{A}$, then by the argument in case 1 we have that e is generated by the image of f and the atoms of $\mathcal{B}$. Also, if e is an atom of $\mathcal{B}$, it has to be below some atom of $f(\mathcal{A})$. If neither a nor $b-a$ is an atom of $\mathcal{A}$, then we are in case 2. Let $u < s_1$ be the least stage at which there is an element strictly below $c = f_s(b)$ in $\mathcal{B}_u$. We know such elements exist in $\mathcal{B}_t$, namely e, so $u \leq t$. Therefore, at s_1 we define $f_{s_1}(a)$ and $f_{s_1}(b-a)$ to be elements of $\mathcal{B}_u \restriction c \subseteq \mathcal{B}_t \restriction c$. Since e in minimal in $\mathcal{B}_t$, e is below either $f_{s_1}(a)$ or $f_{s_1}(b-a)$. This contradicts the minimality of c.

Acknowledgements

The author would like to thank Rod Downey for motivating him to work on this problem, Noam Greenberg for proofreading part of this paper, Kenneth Harris for proofreading and pointing out a significant simplification of our main lemma, and the referee for detailed suggestions.

References

[AN81] C. J. Ash and A. Nerode. Intrinsically recursive relations. In *Aspects of effective algebra (Clayton, 1979)*, pages 26–41. Upside Down A Book Co. Yarra Glen, Vic., 1981. MR629248 (83a:03039)

[DJ94] Rod Downey and Carl G. Jockusch. Every low Boolean algebra is isomorphic to a recursive one. *Proc. Amer. Math. Soc.*, 122(3):871–880, 1994. MR1203984 (95a:03044)

[DM91] Rodney G. Downey and Michael F. Moses. Recursive linear orders with incomplete successivities. *Trans. Amer. Math. Soc.*, 326(2):653–668, 1991. MR1005933 (91k:03115)

[Dow93] Rod Downey. Every recursive Boolean algebra is isomorphic to one with incomplete atoms. *Ann. Pure Appl. Logic*, 60(3):193–206, 1993. MR1216669 (94c:03059)

[Dow97] Rodney G. Downey. On presentations of algebraic structures. In *Complexity, logic, and recursion theory*, volume 187 of *Lecture Notes in Pure and Appl. Math.*, pages 157–205. Dekker, New York, 1997. MR1455136 (98k:03099)

[Gon75] S. S. Goncharov. Certain properties of the constructivization of Boolean algebras. *Sibirskii Matematicheskii Zhurnal*, 16(2):264–278, 1975. MR0381957 (52:2846)

[KS00] Julia F. Knight and Michael Stob. Computable Boolean algebras. *J. Symbolic Logic*, 65(4):1605–1623, 2000. MR1812171 (2001m:03086)

[Rem81a] J. B. Remmel. Recursive isomorphism types of recursive Boolean algebras. *J. Symbolic Logic*, 46(3):572–594, 1981. MR627907 (83a:03042)

[Rem81b] Jeffrey B. Remmel. Recursive Boolean algebras with recursive atoms. *J. Symbolic Logic*, 46(3):595–616, 1981. MR627908 (82j:03055)

[Rem89] J. B. Remmel. Recursive Boolean algebras. In *Handbook of Boolean algebras, Vol. 3*, pages 1097–1165. North-Holland, Amsterdam, 1989. MR991614

[Thu95] John J. Thurber. Every low_2 Boolean algebra has a recursive copy. *Proc. Amer. Math. Soc.*, 123(12):3859–3866, 1995. MR1283564 (96b:03047)

Department of Mathematics, University of Chicago, Chicago, Illinois 60637
E-mail address: antonio@mcs.vuw.ac.nz
URL: www.math.uchicago.edu/∼antonio