Effective presentability of Boolean algebras of Cantor–Bendixson rank 1

Rod Downey [a,1] Carl G. Jockusch, Jr. [b,2]

[a] *Department of Mathematics, Victoria University of Wellington, P.O. Box 600, Wellington, New Zealand*

[b] *Department of Mathematics, University of Illinois, 1409 West Green St., Urbana, IL 61801-2917, USA*

Abstract

We show that there is a computable Boolean algebra $\mathcal{B}$ and a computably enumerable ideal I of $\mathcal{B}$ such that the quotient algebra $\mathcal{B}/I$ is of Cantor–Bendixson rank 1 and is not isomorphic to any computable Boolean algebra. This extends a result of L. Feiner and is deduced from Feiner's result even though Feiner's construction yields a Boolean algebra of infinite Cantor–Bendixson rank.

1 Introduction and Notation

A Boolean algebra $\mathcal{B}$ is said to be *computable* if its universe is a computable set of natural numbers and its operations ($\cup, \cap$, and $'$) are computable partial functions. $\mathcal{B}$ is said to be *computably enumerable (c.e.)* if it is the quotient of a computable Boolean algebra by a computably enumerable ideal. L. Feiner showed in [3, Theorem 5.1] that there is a c.e. Boolean algebra which is not isomorphic to any computable Boolean algebra. A simplified version of Feiner's proof is given by J. Thurber in [12, Corollary 3.4]. Both Feiner's and Thurber's proofs make strong use of the n–th Cantor–Bendixson derivative of the constructed Boolean algebra for each $n < \omega$ and definitely produce Boolean algebras of infinite Cantor–Bendixson rank. Thus it would appear that some totally new approach would be required to construct a Boolean algebra of Cantor–Bendixson rank 1 which is c.e. but not isomorphic to any computable

[1] Downey's research is partially supported by the Victoria University IGC and by the New Zealand Marsden Fund for Basic Science.

[2] This work was supported by grant DMS 95-03398 from the National Science Foundation and by the Victoria University of Wellington.

Boolean algebra. Nevertheless, in the current paper we deduce the existence of such a Boolean algebra from the relativization of Feiner's theorem to $0^{(3)}$. This is done by "coding" an arbitrary countable Boolean algebra into one of rank at most 1. This coding construction also shows that there are continuum many pairwise nonisomorphic countable Boolean algebras of Cantor–Bendixson rank 1. Rather than working directly with Boolean algebras, we work with their Stone spaces, represented as the set of paths through finitely branching trees.

For background on computability theory, see Soare [10]. We follow its notation and terminology except that the traditional terminology of "recursive" and "recursively enumerable (r.e.)" is replaced by "computable" and "computably enumerable (c.e.)" respectively in order to stress the intensional sense of these concepts. Arguments for this change in terminology are given by Soare in [11]. Other papers on (relative) computability of isomorphic copies of Boolean algebras include [1,2,5,13].

2 Stone spaces and computable trees

The well–known Stone representation theorem gives a convenient method of converting problems about Boolean algebras to problems about Boolean spaces, i.e. compact Hausdorff spaces with a neighborhood basis consisting of clopen sets. (See, for example, [7, Chapter 3] for a discussion of the Stone representation theorem.) Specifically, the *Stone space* of a Boolean algebra $\mathcal{B}$ is defined to be the set X of all ultrafilters of $\mathcal{B}$. If $b \in \mathcal{B}$, the family of all ultrafilters of $\mathcal{B}$ having b as an element is a typical basic open subset of X. Going in the other direction, the clopen subsets of X form a Boolean algebra of sets which is isomorphic to $\mathcal{B}$. Two Boolean algebras are isomorphic iff their respective Stone spaces are homeomorphic, and we use $\cong$ to denote both the isomorphism relation on Boolean algebras and the homeomorphism relation on topological spaces. The Boolean algebra $\mathcal{B}$ is countable iff its Stone space X is separable.

If X is any Boolean space, let X' (the Cantor–Bendixson derivative of X) be the set of non–isolated points of X, with the subspace topology. If $\mathcal{B}$ is a Boolean algebra with Stone space X, then X' is homeomorphic to the Stone space of the quotient of $\mathcal{B}$ by the ideal generated by its atoms. As is well–known, this operation can be iterated into the transfinite, but we are concerned only with the first couple of steps. A Boolean algebra has rank 0 if its Stone space X satisfies $X' = X$ (i.e. X is *perfect.*) The countable atomless Boolean algebra is the unique countable Boolean algebra (up to isomorphism) of rank 0, and of course it has a computable presentation. A Boolean algebra with Stone space X has rank 1 if $X \neq X' = X''$.

A *binary* string is a finite sequence of elements of $\{0,1\}$ and a *ternary* string is a finite sequence of elements of $\{0,1,2\}$. Let $2^{<\omega}$ denote the set of binary strings and $3^{<\omega}$ denote the set of ternary strings. If σ and τ are strings, let $\sigma \subseteq \tau$ denote that σ is an initial subsequence of τ, and let $|\sigma|$ denote the length of σ. We call σ and τ *compatible* if $\sigma \subseteq \tau$ or $\tau \subseteq \sigma$, and otherwise *incompatible*. For our purposes, a *tree* is a set T of ternary strings which is closed downwards under $\subseteq$. If T is a tree, a function $f : \omega \to \omega$ is called a *path* through T if every string σ with $\sigma \subseteq f$ is in T. Let $[T]$ denote the set of paths through the tree T. If $\sigma \in T$, let T_σ denote the set of strings in T which are compatible with σ. If T is a tree, then T_σ is also a tree. Topologize $[T]$ by letting the basic open sets be those of the form $[T_\sigma]$ where $\sigma \in T$ and $T_\sigma \neq \emptyset$. Then $[T]$ is a separable Boolean space. (To see that $[T]$ is compact use that T is finite–branching i.e. it contains only finitely many strings of any given length.) Going in the other direction, the Stone space of any countable Boolean algebra is naturally homeomorphic to $[T]$ for some tree $T \subseteq 2^{<\omega}$. To see this, let $\{b_n\}_{n\in\omega}$ be a listing of the universe of a countable Boolean algebra $\mathcal{B}$. For any string $\sigma \in 2^{<\omega}$, let $b_\sigma \in \mathcal{B}$ be defined as $\cap\{b_n : \sigma(n) = 1\} \cap \cap\{b'_n : \sigma(n) = 0\}$, where $b_\lambda = 1_{\mathcal{B}}$ by convention. Let $T = \{\sigma \in 2^{<\omega} : b_\sigma \neq 0_{\mathcal{B}}\}$. If $f \in [T]$ let $U_f = \{b_n : f(n) = 1\}$. It is easily seen that U_f is an ultrafilter of $\mathcal{B}$ and that the mapping $f \mapsto U_f$ is a homeomorphism of $[T]$ onto the Stone space of $\mathcal{B}$. We say that the tree T *represents* the Boolean algebra $\mathcal{B}$ if $[T]$ is homeomorphic to the Stone space of $\mathcal{B}$. The following proposition gives connections between the computability properties of Boolean algebras and their representing trees. A string σ is called a *terminal node* of a tree T if $\sigma \in T$ and no string τ properly extending σ is in T. We write λ for the empty string.

Proposition 1 (Folklore)*(i) Every computable Boolean algebra is represented by some computable tree $T \subseteq 2^{<\omega}$ with no terminal nodes. Conversely, every computable tree $T \subseteq 2^{<\omega}$ with no terminal nodes represents some computable Boolean algebra.*

(ii) Every c.e. Boolean algebra is represented by some computable tree $T \subseteq 2^{<\omega}$. Conversely, every computable tree $T \subseteq 2^{<\omega}$ represents some c.e. Boolean algebra.

Proof. The first statement of the first part follows immediately by effectivizing the proof above that every countable Boolean algebra is represented by some tree $T \subseteq 2^{<\omega}$. The converse is also easy to check. For the second part, assume that $\mathcal{B}$ is a c.e. Boolean algebra. The above argument produces a tree $T \subseteq 2^{<\omega}$ which represent $\mathcal{B}$ and is co–c.e., i.e. $2^{<\omega} - T$ is c.e. But then there is a computable tree $U \subseteq 2^{<\omega}$ such that $[U] = [T]$. (Let U consist of all strings $\sigma \in 2^{<\omega}$ such that no string $\tau \subseteq \sigma$ has been enumerated out of T by stage $|\sigma|$ in a fixed enumeration of $2^{<\omega} - T$.) The converse is easy to check because if

T is a computable tree, $\{\sigma \in T : [T_\sigma] \neq \emptyset\}$ is co–c.e. (by König's Lemma.)

$\square$

Let $A \subseteq \omega$. A Boolean algebra $\mathcal{B}$ is said to be A–computable if its universe is a set of natural numbers which is computable from A and its Boolean operations are partial A–computable functions. $\mathcal{B}$ is said to be A–c.e. if it is the quotient of an A–computable Boolean algebra by an ideal which is c.e. in A. It is clear that the results of Section 2 hold when relativized to A, i.e. computable Boolean algebras are replaced by A–computable Boolean algebras, etc. The following result shows that the $0'$–computable Boolean algebras coincide with the c.e. Boolean algebras, up to isomorphism.

Proposition 2 (Feiner) *If $\mathcal{B}$ is a c.e. Boolean algebra, then $\mathcal{B}$ is isomorphic to some $0'$–computable Boolean algebra. Conversely, each $0'$–computable Boolean algebra is isomorphic to some c.e. Boolean algebra.*

Proof. This result is implicit in [4]. However, we outline a direct proof here for the convenience of the reader. The first part is very easy, as all c.e. sets are $0'$–computable. For the converse, by Proposition 1 it suffices to show that for any $0'$–computable tree $[T] \subseteq 2^{<\omega}$ with no terminal nodes there is a computable tree $[U] \subseteq 2^{<\omega}$ such that $[U] \cong [T]$. We sketch the idea of the construction of U but leave the details to the reader. To each string $\sigma \in T$ we assign a string $f(\sigma) \in U$, where f will be a certain $0'$–computable partial function. We will have that, for $\sigma, \tau \in T$, that $f(\sigma) \subseteq f(\tau)$ iff $\sigma \subseteq \tau$. From this it follows easily (as T has no terminal nodes) that each string in the range of f is extendible to a path in $[U]$. Conversely, each string extendible to a path in $[U]$ will be extendible to a string in the range of f. From this it follows that f induces a homeomorphism φ from $[T]$ to $[U]$, i.e. $\varphi(g) = \cup_{\sigma \subseteq g} f(\sigma)$, for $g \in [T]$.

To construct f and U, we use $\{T_s\}_{s \in \omega}$, a recursive approximation to T. By modifying this approximation if necessary, we may assume that for each s the set T_s is a nonempty tree with no terminal nodes. Let s_0 be the least number s such that for all $t \geq s$ and all strings $\sigma \in 2^{<\omega}$ with $|\sigma| \leq 1$, $\sigma \in T$ iff $\sigma \in T_t$. Then $f(\lambda)$ will be a string in U of length s_0. Note that we may recursively approximate s_0 in such a way that our initial approximation is 0, and if our approximation at $s+1$ differs from that at s, then our approximation to s_0 at stage $s+1$ is simply s. Thus our initial approximation to $f(\lambda)$ is λ, and we start the construction of U by letting it agree with T_0 on strings of length at most 1. (Note that T_0 contains λ and at least one string of length 1 since it is a nonempty tree with no terminal nodes.) In building U at stage $s+1$, we decide membership in U for all binary strings of length $s+1$, and we assume inductively that U contains at least one string of length s. If our approximation to s_0 changes at stage $s+1$, then we effectively choose a string τ of length s in U and let it be our new candidate for $f(\lambda)$. We want the rest of the construction of U to take place above τ, so we omit from U all strings of length $s+1$ which do not extend τ. Further, we act in the belief that the approximation T_{s+1} to T is correct on binary strings of length 1. Thus, for $i \leq 1$ we put $\tau^\frown i$ into U iff $< i > \in T_{s+1}$. (This puts at least one string of length $s+1$ into U because T_{s+1} contains at least one string of length 1.) Obviously, this process must converge because our approximation to s_0 converges. The inductive step for defining $f(\sigma^\frown i)$ for $i \leq 1$ given $f(\sigma)$ is similar. We will have $f(\sigma^\frown i) \supseteq f(\sigma)^\frown i$ and $|f(\sigma^\frown i)|$ will be the least number s such that $s > |f(\sigma)|$ and for all $t \geq s$ and all $j \leq 1$, $\sigma^\frown i^\frown j \in T$ iff $\sigma^\frown i^\frown j \in T_t$. We omit further details. $\Box$

3 The basic construction

If X is a topological space, let $\mathcal{I}(X)$ denote the set of points in X which are limit points of the isolated points of X, i.e. $\mathcal{I}(X) = X' \cap \overline{(X - X')}$. If X is a Boolean space, then so is $\mathcal{I}(X)$ (in the relative topology), as $\mathcal{I}(X)$ is a closed subspace of X. The following easy construction, which is fundamental for this paper, shows that every separable Boolean space is homeomorphic to $\mathcal{I}(X)$ for some separable Boolean space X of Cantor–Bendixson rank ≤ 1.

Definition 3 *Let $T \subseteq 2^{<\omega}$ be a binary tree. Define a new tree $F(T) \subseteq 3^{<\omega}$ by starting with $2^{<\omega}$ and then attaching an isolated path to each node of T. More precisely, let $F(T)$ consist of all strings which are either in $2^{<\omega}$ or consist of a string in T followed by finitely many 2's.*

The paths through $F(T)$ are those infinite strings which either consist entirely of 0's and 1's or else consist of a string in T followed by an infinite string of 2's. The isolated paths in $[F(T)]$ are clearly those of the latter form, so $\mathcal{I}([F(T)]) = [T]$. It also follows from these remarks that $[F(T)]' = [2^{<\omega}]$ which is a perfect space. Hence the Cantor–Bendixson rank of $[F(T)]$ is at most 1.

The above definition leads easily to the following result.

Proposition 4 *There are continuum many pairwise nonisomorphic countable Boolean algebras of Cantor–Bendixson rank* 1*.*

Proof. There exist continuum many pairwise non–isomorphic countable Boolean algebras (for example by [8, Corollary 2.1.2]), and hence continuum many pairwise non–homeomorphic spaces $[T]$ for $T \subseteq 2^{<\omega}$. As $\mathcal{I}([F(T)]) = [T]$ and the homeomorphism type of $\mathcal{I}(X)$ depends only on the homeomorphism type of X, it follows that there are continuum many homeomorphism types of separable Boolean spaces of Cantor–Bendixson rank ≤ 1. As there is only one of rank 0, the result follows from the Stone representation theorem. $\square$

4 The Main Result

It follows from Proposition 4 that there are countable Boolean algebras of Cantor–Bendixson rank 1 which are not isomorphic to any computable Boolean algebra. This gives a negative answer to Question 5.12 of [2]. Our main result effectivizes this by showing that there are such Boolean algebras which are c.e.

Theorem 5 *There is a Boolean algebra $\mathcal{B}$ of Cantor–Bendixson rank 1 which is isomorphic to a c.e. Boolean algebra but not to any computable Boolean algebra.*

Proof. Let $\mathcal{B}_0$ be a Boolean algebra which is $0^{(3)}$–c.e. but not isomorphic to any $0^{(3)}$–computable Boolean algebra. Such a Boolean algebra may be obtained by relativizing the proof of Feiner's theorem [3, Theorem 5.1] to $0^{(3)}$. Let $T \subseteq 2^{<\omega}$ be a $0^{(3)}$–computable tree which represents $\mathcal{B}_0$. Such a tree T exists by Proposition 1, relativized to $0^{(3)}$. Finally, let $\mathcal{B}$ be a Boolean algebra represented by $F(T)$, where $F(T)$ is as defined in Definition 3. It is clear that $\mathcal{B}$ has Cantor–Bendixson rank at most 1. The following lemmas will show that $\mathcal{B}$ satisfies the rest of the conclusion of the theorem.

Lemma 6 *$\mathcal{B}$ is not isomorphic to any computable Boolean algebra.*

Proof. Suppose for a contradiction that $\mathcal{B}$ is isomorphic to $\mathcal{B}_1$, a computable Boolean algebra. Let T_1 be a computable tree without terminal nodes which represents $\mathcal{B}_1$. Such a tree T_1 exists by Proposition 1, and $[T_1] \cong [F(T)]$ where T is as chosen above. It follows that $\mathcal{I}([T_1]) \cong \mathcal{I}([F(T)]) = [T]$. We now define a Σ^0_2 tree $T_2 \subseteq T_1$ such that $[T_2] \cong \mathcal{I}([T_1])$. First, let I be the set of strings σ on T_1 such that any two extensions of σ on T_1 are compatible. It is easily seen (using the fact that T_1 has no terminal nodes) that I is the set of nodes of T which lie on a unique (necessarily isolated) branch of T. Let T_2 be the set of nodes $\sigma \in T_1$ such that there exist incompatible strings τ_1, τ_2 which are each in I and extend σ. It is easy to see that I is Π^0_1, and so T_2 is Σ^0_2, and that $[T_2] \cong \mathcal{I}([T_1])$. Finally, let T_3 be the set of strings $\sigma \in T_2$ such that there exists $f \in T_2$ with $f \supseteq \sigma$, i.e. T_3 is the set of extendible nodes of T_2. By König's Lemma, T_3 is also the set of nodes σ of T_2 which have extensions in T_2 of each length $\geq |\sigma|$, so $T_3 \in \Pi^0_3$. Thus T_3 is a $0^{(3)}$–computable tree, and it clearly has no terminal nodes. But $[T_3] = [T_2] \cong \mathcal{I}([T_1]) \cong [T]$, so T_3 represents $\mathcal{B}_0$. Thus by Proposition 1 , $\mathcal{B}_0$ is isomorphic to some $0^{(3)}$–computable Boolean algebra, in contradiction to our choice of $\mathcal{B}_0$. $\square$

Since $\mathcal{B}$ is not isomorphic to any computable Boolean algebra, it does not have rank 0, so its rank is exactly 1. It remains to show that $\mathcal{B}$ is isomorphic to some c.e. Boolean algebra. The following lemma (relativized to $0'$) will be used to replace the $0^{(3)}$–computable tree T by a Σ^0_3 tree.

Lemma 7 *If V is any $0'$–computable tree, there is a c.e tree W such that $[W] = [V]$.*

Proof. Let $\{V_s\}_{s\in\omega}$ be a uniformly recursive sequence of sets with $V = \lim_s V_s$ pointwise. Define

$$W = \{\sigma : (\exists s \geq |\sigma|)(\forall \tau \subseteq \sigma)[\tau \in V_s]\}$$

It is easy to check that this works. $\square$

The following lemma is the main step in showing that $\mathcal{B}$ is isomorphic to some c.e. Boolean algebra.

Lemma 8 *Let $U \subseteq 2^{<\omega}$ be a $0^{(2)}$–computable tree. There is a computable tree $V \subseteq 3^{<\omega}$ with no terminal nodes such that $[V] \cong [F(U)]$.*

Before giving the proof of this lemma we indicate how it is used to finish the proof of the theorem. Since T is $0^{(3)}$–computable, the proof of Lemma 8 (relativized to $0'$) shows that there is a $0'$–computable tree $V \subseteq 2^{<\omega}$ with no terminal nodes such that $[V] \cong [F(T)]$. Then by relativizing Proposition 1 to $0'$ it follows that V represents some $0'$–computable Boolean algebra $\mathcal{B}_2$, and $\mathcal{B}_2 \cong \mathcal{B}$ since they are represented respectively by V and $F(T)$. By Proposition 2, $\mathcal{B}_1$, being $0'$–computable, is isomorphic to some c.e. Boolean algebra, and thus so is $\mathcal{B}$. It remains to prove Lemma 8.

Proof. By Lemma 7, relativized to $0'$, we may assume without loss of generality that U is a $0'$–c.e. tree, and hence is Σ^0_2. Hence there is a uniformly recursive sequence of sets $\{U_s\}_{s\in\omega}$ with $U_s \subseteq 2^\omega$ such that, for all $\sigma \in 2^{<\omega}$, $\sigma \in V$ iff $(\exists s)(\forall t \geq s)[\sigma \in V_t]$ (see [10, Theorem IV.3.2]). Recall that $F(U)$ contains all strings in $2^{<\omega}$ and all strings obtained by concatenating any string $\sigma \in U$ with a finite number of 2's. Of course, we cannot expect $F(U)$ to be computable as U is only Σ^0_2. However, we can define a computable tree V which "approximates" $F(U)$ in the following way. We put all strings in $2^{<\omega}$ into V. Recall that, for $\sigma \in 3^{<\omega}$, V_σ denotes the set of strings in V which are compatible with σ. We will ensure that if $\sigma \in U$ then $[V_{\sigma\frown 2}]$ is a nonempty finite set. Further, for $\sigma \in 2^{<\omega} - U$ we ensure that $V_{\sigma\frown 2}$ is a perfect tree, i.e. it is nonempty and each string in it has incompatible extensions in it. The computable tree V is defined as follows. To start, V contains the empty string. Now let $\gamma \in 3^{<\omega}$ and $i < 3$ be given. We decide whether $\gamma\frown i \in V$, assuming inductively that we have decided whether $\gamma \in V$. If $\gamma \in 2^{<\omega}$, put $\gamma\frown i$ into V. If $\gamma \notin V$, don't put $\gamma\frown i$ into V. Suppose now that $\gamma \in V - 2^{<\omega}$, and let σ be the longest string $\mu \subseteq \gamma$ such that $\mu \in 2^{<\omega}$. If $\sigma \notin U_{|\gamma|}$ (in which case we currently think that $V_{\sigma\frown 2}$ should be a perfect tree), put $\gamma\frown i$ into V iff $i = 0$ or 1. If $\sigma \in U_{|\gamma|}$ (in which case we currently think that $[V_{\sigma\frown 2}]$ should be finite), put $\gamma\frown i$ into V iff $i = 2$. It is immediate from this definition that V is a computable tree with no terminal nodes. Further, it is easy to see that the sets $V_{\sigma\frown 2}$ for $\sigma \in 2^{<\omega}$ have the desired properties. For example, suppose that $\sigma \in U$, and fix $s > |\sigma|$ with $\sigma \in U_t$ for all $t \geq s$. Let D the set of strings in V which extend $\sigma\frown 2$ and have length s. D is clearly a finite non–empty set of strings. Furthermore, $[V_\sigma]$ consists precisely of the paths by concatenating strings in D with an infinite sequence of 2's. The proof that $V_{\sigma\frown 2}$ is a perfect tree for $\sigma \in 2^{<\omega} - U$ is left to the reader.

It does not seem apparent that $[V] \cong [F(U)]$. Isolated paths through $F(U)$ are replaced by finite nonempty sets of isolated paths in $[V]$. Furthermore, perfect subtrees are added to V which do not correspond to anything in $[F(U)]$. However, we are rescued by a result of Ketonen [6] which is a consequence of a theorem of Vaught [7, Theorem 5.15].

Lemma 9 (Ketonen) *[6, Lemma 1.13]. Let X and Y be separable uncountable Boolean spaces of Cantor–Bendixson rank ≤ 1 each having infinitely many isolated points. Then $X \cong Y$ iff there is a homeomorphism $\varphi : X' \to Y'$ such that $\varphi(\mathcal{I}(X)) = \mathcal{I}(Y)$.*

Note that the conclusion of Lemma 8 is immediate if $[U]$ is empty. Thus we assume without loss of generality that $[U]$ is nonempty, from which it follows that $[F(U)]$ and $[V]$ each have infinitely many isolated points. We apply the above lemma with $X = [F(U)]$ and $Y = [V]$, which are clearly uncountable and of Cantor–Bendixson rank at most 1. To construct φ as required in the lemma, recall the well–known fact that any two separable perfect Boolean spaces are homeomorphic. For each $\sigma \in 2^{<\omega} - U$, let φ_σ be a homeomorphism from $\{f \in 2^\omega : f \supseteq \sigma\}$ onto $[V_\sigma]$. (Recall that V_σ is the set of strings in V compatible with σ and is a perfect tree for $\sigma \in 2^{<\omega} - U$.) Now, to define φ, suppose that $f \in [F(U)]' = 2^\omega$ is given. If $f \in [U]$, let $\varphi(f) = f$. If $f \notin [U]$, let σ be the shortest string extended by f and not in U, and set $\varphi(f) = \varphi_\sigma(f)$. To conclude that $[F(U)] \cong [V]$ we must show that φ is a homeomorphism of $[F(U)]'$ onto $[V]'$ such that $\varphi(\mathcal{I}([F(U)]) = \mathcal{I}(V)$. We illustrate this routine verification by checking that f is 1-1 and leave the rest to the reader. Suppose that $\varphi(f) = \varphi(g)$. We must show that $f = g$. This is immediate if f and g

are both in $[U]$. Suppose now that $f \in [U]$ and $g \notin [U]$. Let σ be the shortest string extended by g which is not in U, so that $\varphi(g) = \varphi_\sigma(g) \supseteq \sigma \notin U$. It follows that $\varphi(g) \notin [U]$, but $\varphi(f) = f \in [U]$, a contradiction. Suppose finally that $f \notin [U]$ and $g \notin [U]$. Let σ, τ be the shortest strings not in U extended by f, g respectively. Since $\varphi(f) = \varphi(g)$ extends both σ and τ, it follows that σ and τ are compatible. Hence, by the minimality of σ and τ, $\sigma = \tau$. Hence $f = g$ since $\varphi_\sigma = \varphi_\tau$ is 1-1. We thus conclude from Lemma 9 that $[V] \cong [F(U)]$. This completes the proof of Lemma 8 and, as previously explained, our main result follows. $\square$

References

[1] R. Downey and C. Jockusch, Every low Boolean algebra is isomorphic to a recursive one, *Proc. Amer. Math. Soc* **122** (1994), 871–880.

[2] R. Downey, On presentations of algebraic structures, to appear in *Proceedings of COLORET II*, Marcel Dekker.

[3] L. Feiner, Hierarchies of Boolean algebras, *J. Symbolic Logic* **35** (1970), 365–374.

[4] L. Feiner, Degrees of nonrecursive presentability, *Proc. Amer. Math. Soc.* **38** (1973), 621–624.

[5] C. Jockusch and R. Soare, Boolean algebras, Stone spaces, and the iterated Turing jump, *J. Symbolic Logic* **59** (1994), 1121–1138.

[6] J. Ketonen, The structure of countable Boolean algebras, *Annals of Mathematics* **108** (1978), 41-89.

[7] Sabine Koppelberg, General theory of Boolean algebras, Volume 1 of *Handbook of Boolean Algebras*, ed. J.D. Monk and R. Bonnet, North Holland, Amsterdam, New York, Oxford, Tokyo, 1989.

[8] R.S. Pierce, Countable Boolean algebras, pp. 775–876 in *Handbook of Boolean Algebras*, Volume 3, ed. J.D. Monk and R. Bonnet, North Holland, Amsterdam, New York, Oxford, Tokyo, 1989.

[9] J. Remmel, Recursive Boolean algebras, pp. 1097–1166 in *Handbook of Boolean Algebras*, Volume 3, ed. J.D. Monk and R. Bonnet, North Holland, Amsterdam, New York, Oxford, Tokyo, 1989.

[10] R. I. Soare, *Recursively enumerable sets and degrees: A study of computable functions and computably generated sets*, Springer–Verlag, Heidelberg, 1987.

[11] R. I. Soare, Computability and recursion, *Bulletin of Symbolic Logic* **2** (1996), 284–321.

[12] J. Thurber, Recursive and r.e. quotient Boolean algebras, *Arch. Math. Logic* **33** (1994), 121–129.

[13] J. Thurber, Every low_2 Boolean algebra has a recursive copy, *Proc. Amer. Math. Soc* **123** (1995), 3859–3966.