EMBEDDINGS OF COMPUTABLE STRUCTURES

ASHER M. KACH, OSCAR LEVIN, AND REED SOLOMON

ABSTRACT. We study what the existence of a classical embedding between computable structures implies about the existence of computable embeddings. In particular, we consider the effect of fixing and varying the computable presentations of the computable structures.

1. INTRODUCTION

Throughout the field of effective algebra, there are a plethora of instances where classical behavior and effective behavior diverge. A particular example, and the subject of this paper, is within the context of embeddings of algebraic structures. For example, there are computable presentations $\mathcal{S}_1$ and $\mathcal{S}_2$ of computable structures such that $\mathcal{S}_1$ classically embeds into $\mathcal{S}_2$ but for which there is no computable embedding $\alpha : \mathcal{S}_1 \to \mathcal{S}_2$. Indeed, such examples exist within most natural classes of algebraic structures (e.g., linear orders, directed graphs, groups, fields, etc.). However, if the computable presentations of $\mathcal{S}_1$ and $\mathcal{S}_2$ are sufficiently altered, often computable embeddings exist. The following notions capture whether computable embeddings exist after altering the presentations of $\mathcal{S}_1$ and/or $\mathcal{S}_2$.

Definition 1.1. A class $\mathcal{C}$ of computable presentations of computable structures is said to have the *strong embedding property* if for all $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{C}$ such that $\mathcal{S}_1$ classically embeds into $\mathcal{S}_2$, there is a computable embedding $\alpha : \mathcal{S}_1 \to \mathcal{S}_2$.

Definition 1.2. A class $\mathcal{C}$ of computable presentations of computable structures is said to have the *weak domain embedding property* if for all $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{C}$ such that $\mathcal{S}_1$ classically embeds into $\mathcal{S}_2$, there is a computable presentation $\mathcal{S}_1' \cong \mathcal{S}_1$ and a computable embedding $\alpha : \mathcal{S}_1' \to \mathcal{S}_2$.

Definition 1.3. A class $\mathcal{C}$ of computable presentations of computable structures is said to have the *weak range embedding property* if for all $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{C}$ such that $\mathcal{S}_1$ classically embeds into $\mathcal{S}_2$, there is a computable presentation $\mathcal{S}_2' \cong \mathcal{S}_2$ and a computable embedding $\alpha : \mathcal{S}_1 \to \mathcal{S}_2'$.

Definition 1.4. A class $\mathcal{C}$ of computable presentations of computable structures is said to have the *weak embedding property* if for all $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{C}$ such that $\mathcal{S}_1$ classically embeds into $\mathcal{S}_2$, there are computable presentations $\mathcal{S}_1' \cong \mathcal{S}_1$ and $\mathcal{S}_2' \cong \mathcal{S}_2$ and a computable embedding $\alpha : \mathcal{S}_1' \to \mathcal{S}_2'$.

Date: 7 April 2010.

2000 *Mathematics Subject Classification.* 03C57,03D45.

Key words and phrases. computable embedding, strong embedding property, weak domain embedding property, weak range embedding property, weak embedding property.

The first author was partially supported by the Marsden Fund of New Zealand via a Post-Doctoral Fellowship. The authors thank Wesley Calvert, Carl Jockusch, Joe Miller, and Russell Miller for helpful discussions.

Fixing a class $\mathcal{C}$ of computable presentations of computable structures, it is immediate that the pictured implications hold.

$$\begin{tikzcd}
 & \text{weak domain} \arrow[dr] & \\
\text{strong} \arrow[ur] \arrow[dr] & & \text{weak} \\
 & \text{weak range} \arrow[ur] &
\end{tikzcd}$$

A purpose of this paper is to show that no other implications exist. First, we provide examples of structures which fail to have even the weak embedding property: the class of computable ordered abelian groups and the class of computable trees in the language of undirected graphs in Section 2 and Section 3, respectively (other examples of the failure of the weak embedding property are already known). In Section 4, we exhibit a class of algebraic structures that has the weak embedding property but neither the weak domain embedding property nor the weak range embedding property. We then demonstrate, in Section 5 and Section 6 respectively, that the class of computable equivalence structures and the class of computable Boolean algebras have the weak range embedding property but not the weak domain embedding property. In Section 7, we exhibit a class of algebraic structures that has the weak domain embedding property but not the weak range embedding property. For a class of structures which has both the weak domain embedding property and the weak range embedding property but not the strong embedding property, we turn to computable algebraically closed fields in Section 8. Finally, in Section 9, we conclude with some additional examples and open questions.

2. Ordered Abelian Groups

Here we demonstrate that the class of computable ordered abelian groups does not have the weak embedding property. Before doing so, we note that the class of computable trees (viewed as posets), the class of computable linear orders, and any class known to be universal with respect to common computable model theoretic notions (e.g., partial orders, lattices, 2-step nilpotent groups, and integral domains) also fail to have the weak embedding property (see [1], [4], and [4], respectively).

We recall a result about linear orders that will be exploited in our study of ordered abelian groups. We use η to denote the order type of the rationals.

Theorem 2.1 ([4]). *There is a computable nonscattered intrinsically hyperarithmetically scattered linear order, i.e., there is a computable nonscattered linear order $\mathcal{L}_\eta$ such that for all hyperarithmetic presentations $\eta' \cong \eta$ and $\mathcal{L}'_\eta \cong \mathcal{L}_\eta$, there is no hyperarithmetic embedding $\alpha : \eta' \to \mathcal{L}'_\eta$.*

We also recall the following algebraic terminology.

Definition 2.2. Elements x and y in an ordered abelian group $\mathcal{G}$ are *archimedean equivalent*, written $x \approx y$, if there are $m, n \in \mathbb{N}$ such that $m|x| \geq |y|$ and $n|y| \geq |x|$. Here $|x|$ denotes whichever of x and $-x$ is positive if $x \neq 0$ and zero if $x = 0$.

Definition 2.3. A subset U of an ordered abelian group $\mathcal{G}$ is a *set of unique archimedean representatives* for $\mathcal{G}$ provided: $0 \notin U$; for any $u \neq v \in U$, we have $u \not\approx v$; and for all $0 \neq g \in G$, there is some $u \in U$ such that $u \approx g$.

A set of unique archimedean representatives U is a *positive set of unique archimedean representatives* if $x > 0$ for every $x \in U$.

Our choice to exclude 0 from any set of unique archimedean representatives is not necessarily standard, but is chosen to simplify the statements of upcoming results.

It is easy to see that there is always a set of positive unique archimedean representatives computable from $\mathbf{0}'$ if $\mathcal{G}$ is computably presented.

Proposition 2.4. *If $\mathcal{G}$ is a computably presented ordered abelian group, then there is a Π^0_1 (and thus computable from $\mathbf{0}'$) set of unique archimedean representatives.*

It is also easy to check that a set U of positive unique archimedean representatives for an ordered abelian group $\mathcal{G}$ forms a linear order (using the ordering inherited from $\mathcal{G}$). With a little more work, it is possible to demonstrate the reverse: given any linear order $\mathcal{L}$, there is an ordered abelian group $\mathcal{G}_{\mathcal{L}}$ such that for any set U of positive unique archimedean representatives, $\langle U, \leq\rangle \cong \mathcal{L}$. The group $\mathcal{G}_{\mathcal{L}}$ can be $\oplus_{x_i \in \mathcal{L}} \mathbb{Z}_{x_i}$, i.e., the group whose elements are formal sums $z_{i_1}x_{i_1} + ... + z_{i_k}x_{i_k}$ for $z_{i_1}, \ldots, z_{i_k} \in \mathbb{Z}$, and whose order is generated by

$$z_{i_1}x_{i_1} + ... + z_{i_k}x_{i_k} > 0 \quad \text{ if and only if } \quad z_{i_j} > 0$$

where x_{i_j} is $\mathcal{L}$-maximal amongst $x_{i_1}, \ldots, x_{i_k}$. Then for any two sums $g, h \in G_{\mathcal{L}}$, we have $g < h$ if and only if $h - g > 0$.

It follows that the set of generators $\{x_i : x_i \in L\}$ is a set of positive unique archimedean representatives with order type $\mathcal{L}$. Also, if $\mathcal{L}$ is computable, then $\mathcal{G}_{\mathcal{L}}$ is computable.

Theorem 2.5. *The class of computable ordered abelian groups fails to have the weak embedding property.*

Proof. It suffices to exhibit computable ordered abelian groups $\mathcal{G}_1$ and $\mathcal{G}_2$ such that $\mathcal{G}_1$ classically embeds into $\mathcal{G}_2$ but for which there is no computable embedding $\alpha : \mathcal{G}_1' \to \mathcal{G}_2'$ for any computable $\mathcal{G}_1' \cong \mathcal{G}_1$ and $\mathcal{G}_2' \cong \mathcal{G}_2$. We take $\mathcal{G}_1$ to be any computable presentation of $\mathcal{G}_\eta$ and $\mathcal{G}_2$ to be any computable presentation of $\mathcal{G}_{\mathcal{L}_\eta}$, where $\mathcal{L}_\eta$ is a computable nonscattered linear order that is intrinsically hyperarithmetically scattered. As $\mathcal{L}_\eta$ is nonscattered, there is a classical embedding from η into $\mathcal{L}_\eta$. This induces an embedding from $\mathcal{G}_1$ into $\mathcal{G}_2$: x_{i_j} in $\mathcal{G}_1$ is mapped to x_{i_k} in $\mathcal{G}_2$ exactly when i_j in η is mapped to i_k in $\mathcal{L}_\eta$. However, there cannot be computable presentations $\mathcal{G}_1' \cong \mathcal{G}_1$ and $\mathcal{G}_2' \cong \mathcal{G}_2$ and a computable embedding $\alpha : \mathcal{G}_1' \to \mathcal{G}_2'$. We reason as follows.

Suppose there were such computable presentations $\mathcal{G}_1'$ and $\mathcal{G}_2'$ and a computable embedding $\alpha : \mathcal{G}_1' \to \mathcal{G}_2'$. Then $\mathbf{0}'$ could compute a set of positive unique archimedean representatives U_1' for $\mathcal{G}_1'$ and U_2' for $\mathcal{G}_2'$. Then $\mathbf{0}''$ could, for each $x \in U_1'$, uniformly find the element $y \in U_2'$ such that $y \approx \alpha(x)$. However, the map $x \mapsto y$ would give a hyperarithmetic embedding (indeed, an embedding computable in $\mathbf{0}''$) of a $\mathbf{0}'$ copy of η into a $\mathbf{0}'$ copy of $\mathcal{L}_\eta$, contradicting Theorem 2.1. $\square$

Remark 2.6. A similar argument can be used to establish the corresponding result for ordered fields. That is, the class of computable ordered fields does not have the weak embedding property.

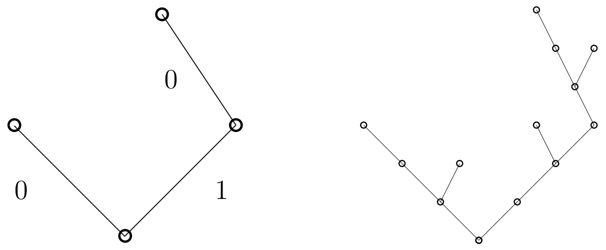
3. Trees in the Language of Undirected Graphs

Here we demonstrate that the class of computable trees, i.e., acyclic connected undirected graphs in the language of undirected graphs, does not have the weak embedding property. The context only necessitates slightly modifying the coding modules used in demonstrating that the class of computable directed graphs does not have the weak embedding property (see [4]).

Definition 3.1. If $\mathcal{T} \subseteq 2^{<\omega}$ is a subtree of $2^{<\omega}$ (in the usual sense), let $\hat{\mathcal{T}}$ be the computable tree (as an undirected graph) whose universe contains $\{\sigma : \sigma \in T\}$ and three additional elements x, y, w for each non-root $\sigma \in T$ with edge relations precisely $E(\sigma^-, x)$, $E(x, y)$, $E(y, \sigma)$, and $E(x, w)$ if $\sigma = \sigma^- \frown 0$ or $E(y, w)$ if $\sigma = \sigma^- \frown 1$.

Here we use σ^- to denote the string of length $|\sigma| - 1$ that is an initial segment of σ.

We illustrate this definition with an example, with $\mathcal{T}$ pictured on the left and $\hat{\mathcal{T}}$ pictured on the right. The tree $\hat{\mathcal{T}}$ is $\mathcal{T}$ with each non-root vertex replaced by a set of four edges, with the location of the *dead end* depending on whether the last digit in σ is zero or one.



With this coding widget, the proof mirrors the proof for computable directed graphs (see [4]).

Theorem 3.2. *The class of computable trees fails to have the weak embedding property.*

Proof. Fix a subtree (in the usual sense) $\mathcal{T} \subseteq 2^{<\omega}$ that is infinite, computable, and has no computable paths. It suffices to take $\mathcal{T}_1$ to be any computable presentation of the tree $\{0^n : n \in \omega\}$ (in the language of undirected graphs) and $\mathcal{T}_2$ to be any computable presentation of $\hat{\mathcal{T}}$. As $\mathcal{T}$ has an infinite path, classically the tree $\mathcal{T}_1$ will embed into the tree $\mathcal{T}_2$. We establish the lack of a computable embedding $\alpha : \mathcal{T}_1' \to \mathcal{T}_2'$ for computable presentations $\mathcal{T}_1' \cong \mathcal{T}_1$ and $\mathcal{T}_2' \cong \mathcal{T}_2$ by showing that both $\mathcal{T}_1$ and $\mathcal{T}_2$ are computably categorical.

Of course, it is immediate that $\mathcal{T}_1$ is computably categorical. We show $\mathcal{T}_2$ is computably categorical by describing a computable isomorphism $\pi : \mathcal{T}_2' \to \mathcal{T}_2$ for any computable presentation $\mathcal{T}_2' \cong \mathcal{T}_2$. Nonuniformly, we may assume knowledge of the vertices in T_2 and T_2' corresponding to the root of $\mathcal{T}$. Note that given any vertex $v \in T_2'$, it is possible to effectively determine to which widget v belongs, which vertex of that widget v is, and to which edge in $\mathcal{T}$ that widget corresponds. This is done by searching through T_2' for the path from v back to the vertex $\lambda_{\mathcal{T}_2'}$ and then counting by threes from $\lambda_{\mathcal{T}_2'}$ back to v. Once this edge in $\mathcal{T}$ is found, it is a simple task to locate the corresponding widget in $\mathcal{T}_2$. We can then effectively define π on all four vertices in v's widget, including v. As this procedure can be carried out effectively for any $v \in T_2'$, we see that π is a computable isomorphism.

Finally, there cannot be a computable embedding $\alpha : \mathcal{T}_1 \to \mathcal{T}_2$. If there were such a computable embedding sending the root vertex of $\mathcal{T}_1$ to the root vertex of $\mathcal{T}_2$, a computable path through $\mathcal{T}$ could be recursively defined as follows. The zeroth bit of the computable path through $\mathcal{T}$ under construction is zero if there is an element $w \in T_2$ with $E(\alpha(0), w)$ and one if there is an element $w \in T_2$ with $E(\alpha(00), w)$. As (exactly) one of these must exist, it suffices to search. Iterating this, searching for w satisfying either $E(\alpha(0^{3k+1}), w)$ or $E(\alpha(0^{3k+2}), w)$, yields the kth bit of the computable path under construction.

More generally, if the root vertex of $\mathcal{T}_1$ was not sent to the root vertex of $\mathcal{T}_2$, only finitely much (nonuniform) information is necessary to recover a computable path in $\mathcal{T}$ from the sequence recursively defined. Of course, this information corresponds to removing some number of digits (if $\alpha(\mathcal{T}_1)$ travels through the root vertex of $\mathcal{T}_2$) or adding some number of digits (if $\alpha(\mathcal{T}_1)$ misses the root vertex of $\mathcal{T}_2$). $\square$

4. A Class of Posets

Here we demonstrate the existence of a class of computable algebraic structures having the weak embedding property but neither the weak domain embedding property nor the weak range embedding property (and so not the strong embedding property). The class will consist of all computable presentations of two isomorphism types of posets.

Definition 4.1. Define $\mathcal{Q}_1$ to be the computable partial order with universe $X \cup Y \cup U$, where $X = \{x_i\}_{i\in\omega}$, $Y = \{y_i\}_{i\in\omega}$, and $U = \{u_i\}_{i\in\omega}$ and order

- the X elements form an ω-chain $x_0 \prec x_1 \prec \dots$;
- the Y elements form an ω^*-chain $y_0 \succ y_1 \succ \dots$ sitting above the X elements (i.e., $x_i \prec y_j$ for all i, j); and
- the U elements form an antichain and the only order relations they satisfy are $x_i \prec u_j$ if $i \leq j$.

So the element u_0 sits above x_0 and is incomparable with everything else; the element u_1 sits above x_1 and x_0 and is incomparable with everything else; etc.

Definition 4.2. Define $\mathcal{Q}_2$ to be the computable partial order with domain $X \cup Y \cup U \cup V \cup W$ where $X = \{x_i\}_{i\in\omega}$, $Y = \{y_i\}_{i\in\omega}$, $U = \{u_i\}_{i\in\omega}$, $V = \{v_i\}_{i\geq 2}$, and $W = \{w_i\}_{i\geq 2}$ and order

- the X elements form an ω-chain $x_0 \prec x_1 \prec \dots$;
- the Y elements form an ω^*-chain $y_0 \succ y_1 \succ \dots$ sitting above the X elements (i.e., $x_i \prec y_j$ for all i, j);
- the U elements form an antichain and the only order relations they satisfy are $x_i \prec u_j$ if and only if $i \leq j$;
- the V elements form an antichain and the order relations v_i satisfies with X and Y elements are $x_j \prec v_i$ for all j, and $y_j \prec v_i$ if $j \geq i$; that is, we make $v_i \succ y_i$ and then only include additional order relations that are forced by transitivity; and
- the W elements form an antichain and the only order relations w_i satisfies are $v_i \prec w_i$, $x_j \prec w_i$ for all j, and $y_j \prec w_i$ if $j \geq i$; that is, we place $w_i \succ v_i$ and only include additional order relations forced by transitivity.

So the poset $\mathcal{Q}_2$ looks like $\mathcal{Q}_1$ except above each element y_i with $i \geq 2$ we have added a new chain $y_i \prec v_i \prec w_i$ of length two.

The isomorphism types of these posets were chosen to have various effectiveness properties, which we proceed to demonstrate.

Lemma 4.3. *In every computable presentation $\mathcal{P}_1'$ of $\mathcal{Q}_1$, the set of X elements in $\mathcal{P}_1'$ is computably enumerable.*

Proof. An element $a \in P_1'$ is an X element if and only if there are two incomparable elements above it. □

Lemma 4.4. *There is a computable presentation $\mathcal{P}_1$ of $\mathcal{Q}_1$ such that the set of Y elements in $\mathcal{P}_1$ is immune.*

Proof. The idea is essentially the same as when making a copy of $(\omega + \omega^*, \leq)$ in which the ω^* part is immune. The requirements are

$$R_e : W_e \text{ infinite } \rightarrow W_e \cap (X \cup U) \neq \emptyset$$

where $X \cup U$ refers to the X and U elements in the copy we are building. We build $\mathcal{P}_1$ by starting to build the X and U elements, placing x_s and u_s at stage s if they are not already built by that stage. We also place potential Y elements and use markers $y_{i,s}$ to denote the current y_i element at stage s. We need to make sure that each $y_{i,s}$ has a limit for each s, so we need a requirement to protect them.

$$N_i : y_{i,s} \text{ has a limit}$$

The action of N_i is just to restrain lower priority requirements from changing $y_{i,s}$. The action of R_e is to wait for a stage s at which some $y_{i,s} \in W_e$ for $i > e$ (to respect the higher priority N_i requirements). Then we move the current $y_{j,s}$ element for $j \geq i$ down to the X part (so each becomes some x_k element) and add appropriate U elements to them. We add new $y_{j,s+1}$ elements to P_1 and declare R_e satisfied. The interaction involves only finite injury to the N_i requirements and no injury to the R_e requirements. □

Lemma 4.5. *In every computable presentation $\mathcal{P}_2'$ of $\mathcal{Q}_2$, the set of Y elements in $\mathcal{P}_2'$ is computably enumerable.*

Proof. Let $\mathcal{T}$ be the five element partial order with elements e_i for $1 \leq i \leq 5$ and order relations $e_1 \prec e_2 \prec e_3$ and $e_1 \prec e_4 \prec e_5$. Thus, $\mathcal{T}$ looks like a "V". Notice that if $\mathcal{T}$ is embedded into $\mathcal{P}_2'$, then the e_1 element must lie on the $\omega + \omega^*$ chain formed by the X and Y elements. Furthermore, the e_2, e_3, e_4 and e_5 elements must lie in $Y \cup V \cup W$.

Then $a \in P_2'$ is a Y element if and only if either

- $a = y_0$ (determined nonuniformly); or
- $a = y_1$ (determined nonuniformly); or
- there are at least two elements of P_2' above a, and there is an embedding of $\mathcal{T}$ into $\mathcal{P}_2'$ such that at least two elements in the image of the embedding lie below a.

The first half of the third condition guarantees that a is not a V or W element and the second half of third condition guarantees that a is not an X or U element. □

Lemma 4.6. *There is a computable copy $\mathcal{P}_2$ of $\mathcal{Q}_2$ such that the set of X elements in $\mathcal{P}_2$ is immune.*

Proof. The idea is just like the proof of Lemma 4.4 except we move potential X elements (with their corresponding potential U elements) up to the Y part (and the U elements become V elements) and we add new W elements. Thus, the idea is just like making a copy of $(\omega + \omega^*, \leq)$ in which the ω part is immune. □

Notice that $\mathcal{Q}_1$ classically embeds into $\mathcal{Q}_2$, but $\mathcal{Q}_2$ does not classically embed into $\mathcal{Q}_1$. Furthermore, any embedding of $\mathcal{Q}_1$ into $\mathcal{Q}_2$ has to send X elements to X elements, Y elements to Y elements, and U elements to U elements.

Theorem 4.7. *The class of computable presentations of $\mathcal{Q}_1$ and $\mathcal{Q}_2$ has the weak embedding property but neither the weak domain embedding property nor the weak range embedding property.*

Proof. For the weak embedding property, the only case where anything needs to be shown is when $\mathcal{S}_1$ is a computable copy of $\mathcal{Q}_1$ and $\mathcal{S}_2$ is a computable copy of $\mathcal{Q}_2$. Taking $\mathcal{S}_1'$ to be $\mathcal{Q}_1$ and $\mathcal{S}_2'$ to be $\mathcal{Q}_2$ suffices.

For the weak domain embedding property, let $\mathcal{P}_2$ be the copy from Lemma 4.6. Suppose for a contradiction that there is a computable copy $\mathcal{P}_1'$ of $\mathcal{Q}_1$ and a computable embedding $\alpha : \mathcal{P}_1' \to \mathcal{P}_2$. This embedding must send the X elements of $\mathcal{P}_1'$ into the X elements of $\mathcal{P}_2$. Therefore, $a \in P_2$ is an X element of $\mathcal{P}_2$ if and only if there is a $b \in \mathcal{P}_1'$ such that $a \prec \alpha(b)$ and b is an X element of $\mathcal{P}_1'$. Therefore, the set of X elements in $\mathcal{P}_2$ is computably enumerable, giving the desired contradiction.

For the weak range embedding property, let $\mathcal{P}_1$ be the copy from Lemma 4.4. Suppose for a contradiction there is a copy $\mathcal{P}_2'$ of $\mathcal{Q}_2$ and a computable embedding $\alpha : \mathcal{P}_1 \to \mathcal{P}_2'$. Such an embedding has to map the Y elements of $\mathcal{P}_1$ into the Y elements of $\mathcal{P}_2'$, and the $X \cup U$ elements of $\mathcal{P}_1$ into the $X \cup U$ elements of $\mathcal{P}_2'$. Since the set of Y elements in $\mathcal{P}_2'$ is computably enumerable, the embedding α gives a computably enumerable description of the Y elements in $\mathcal{P}_1$. That is, $a \in P_1$ is a Y element of $\mathcal{P}_1$ if and only if $\alpha(a)$ is a Y element in $\mathcal{P}_2'$. This is the desired contradiction. □

5. Equivalence Structures

Here we demonstrate that the class of computable equivalence structures has the weak range embedding property (and so the weak embedding property) but not the weak domain embedding property (and so not the strong embedding property). We refer the reader to [2] for background on computable equivalence structures.

Lemma 5.1 (implicit in [2])**.** *If $\mathcal{E}$ is a computable equivalence structure with unbounded character (i.e., finite classes of arbitrarily large size) and (at most) finitely many infinite equivalence classes, then there is a decomposition $\mathcal{E} \cong \mathcal{S}_1 \oplus \mathcal{S}_2$ (where $\oplus$ denotes disjoint union) and a computable function $g : \omega \times \omega \to \omega$ such that for all x:*

- *$g(x, s) \leq g(x, s + 1)$,*
- *$G(x) := \lim_s g(x, s)$ is finite,*
- *$G(x) > x$,*
- *$\mathcal{S}_1$ and $\mathcal{S}_2$ are computable,*
- *$\mathcal{S}_1$ has a class of size k if and only if k is in the range of G, and*
- *$\mathcal{S}_1$ has no infinite classes and at most one class of any fixed finite size.*

Proposition 5.2. *The class of computable equivalence structures has the weak range embedding property.*

Proof. Fixing computable presentations $\mathcal{E}_1$ and $\mathcal{E}_2$ of computable equivalence structures such that $\mathcal{E}_1$ classically embeds into $\mathcal{E}_2$, we exhibit a computable presentation $\mathcal{E}_2' \cong \mathcal{E}_2$ and a computable embedding $\alpha : \mathcal{E}_1 \to \mathcal{E}_2'$. The manner in which we construct the computable presentation $\mathcal{E}_2'$ depends primarily on whether $\mathcal{E}_2$

has bounded or unbounded character and finitely many or infinitely many infinite classes.

If $\mathcal{E}_2$ has infinitely many infinite classes, it suffices to take $\mathcal{E}_2' = \mathcal{E}_2 \oplus \mathcal{E}_\infty$ where $\mathcal{E}_\infty$ is a computable equivalence structure with infinitely many infinite classes and no finite classes. Then $\mathcal{E}_1$ computably embeds into $\mathcal{E}_2'$ by embedding each equivalence class in $\mathcal{E}_1$ into one of the infinitely many additional infinite equivalence classes.

If $\mathcal{E}_2$ does not have infinitely many infinite classes, we must break into two cases: when $\mathcal{E}_2$ has bounded character and when $\mathcal{E}_2$ has unbounded character. In considering these cases, we will without loss of generality assume that $\mathcal{E}_2$ has *no* infinite classes. For if $\mathcal{E}_2$ has a positive but finite number of infinite classes, we can (nonuniformly) select representatives of the infinite classes in $\mathcal{E}_2$ and representatives of the appropriate corresponding classes in $\mathcal{E}_1$. This allows us to compute the embedding (as described below) on all the other classes independent of these finitely many classes.

If $\mathcal{E}_2$ has bounded character and no infinite classes, let k be the largest size such that $\mathcal{E}_2$ has infinitely many classes of size k (such a k must exist). Then $\mathcal{E}_1$ also has only finitely many classes of size larger than k as it classically embeds into $\mathcal{E}_2$ by hypothesis. It is then possible to (nonuniformly) map the classes of size larger than k in $\mathcal{E}_1$ to appropriate classes in $\mathcal{E}_2$. For the remaining classes in $\mathcal{E}_1$, it is possible to map them to classes of size k in $\mathcal{E}_2$.

If $\mathcal{E}_2$ has unbounded character and no infinite classes, it is easy when $\mathcal{E}_1$ has bounded character. It suffices to map each class in $\mathcal{E}_1$ to a class of size at least k, where k is an integer witnessing that $\mathcal{E}_1$ has bounded chracter. When $\mathcal{E}_1$ has unbounded character, it is slightly more difficult. Fix a (classical) decomposition $\mathcal{E}_2 \cong \mathcal{S}_1 \oplus \mathcal{S}_2$ and a computable function g as in Lemma 5.1. We build $\mathcal{E}_2' = \mathcal{S}_1' \oplus \mathcal{S}_2'$ where $\mathcal{S}_1 \cong \mathcal{S}_1'$ and $\mathcal{S}_2 = \mathcal{S}_2'$; the structure $\mathcal{E}_1$ will computably embed into $\mathcal{S}_1'$.

The construction of $\mathcal{S}_1'$ is dynamic and depends on both the presentation $\mathcal{E}_1$ and the function g. For each new class in $\mathcal{E}_1$ (say, of size n at stage s) that appears, we choose a column x (with x large) and start a new class in $\mathcal{S}_1'$ of size $g(x, s)$. The computable embedding $\alpha : \mathcal{E}_1 \to \mathcal{S}_1'$ maps this new class in $\mathcal{E}_1$ to this new class in $\mathcal{S}_1'$. As the stage s increases, we grow (as necessary) this class in $\mathcal{S}_1'$ to have size $g(x, s)$. If the size of the class in $\mathcal{E}_1$ grows beyond the current value of $g(x, s)$, we choose a new large column y and grow the class of $\mathcal{S}_1'$ to have size $g(y, s)$. In this fashion, we have this class of $\mathcal{S}_1'$ always is larger than or equal to the class in $\mathcal{E}_1$ and has size $g(z, s)$ for some column z.

Additionally, at each stage s, we assure that there is a class of size $g(z, s)$ for all $z < s$. This is done by adding additional elements to the class built for $g(z, s-1)$ as necessary if it exists and building a fresh class otherwise.

As each class in $\mathcal{E}_1$ is finite, it is not difficult to see that each class in $\mathcal{S}_1'$ will eventually choose a permanent column x to follow. Indeed, the column x can change only when the class in $\mathcal{E}_1$ grows. Also, it will be the case that $\mathcal{S}_1 \cong \mathcal{S}_1'$ as each class in $\mathcal{S}_1'$ is associated to a value $g(z, s-1)$ and each column z is associated to a class (either explicitly or implicitly at the end of each stage). Finally, the construction explicitly gives the computable embedding $\alpha : \mathcal{E}_1 \to \mathcal{S}_1'$.

Having exhausted all possible cases, we conclude that the class of computable equivalence structures has the weak range embedding property. $\square$

Proposition 5.3. *The class of computable equivalence structures fails to have the weak domain embedding property.*

Proof. It suffices to construct computable equivalence structures $\mathcal{E}_1$ and $\mathcal{E}_2$ with unbounded character and no infinite classes (so $\mathcal{E}_1$ classically embeds into $\mathcal{E}_2$) such that for all $\mathcal{E}_1' \cong \mathcal{E}_1$, there is no computable embedding $\alpha : \mathcal{E}_1' \to \mathcal{E}_2$.

Towards stating the requirements, we view each function $\varphi_e(x, y)$ as computing the characteristic function of a binary relation on $\omega \times \omega$ and let $\mathcal{A}_e$ denote the resulting computable structure. Of course, the function φ_e may not be total, in which case $\mathcal{A}_e$ is not really a computable structure and we can safely disregard it. We view each function $\varphi_i(x)$ as a candidate embedding of $\mathcal{A}_e$ into $\mathcal{E}_2$. Again, the function φ_i may not be total, in which case φ_i is not really a computable embedding and we can safely disregard it.

Throughout, if $\mathcal{E}$ is an equivalence structure and $a \in E$, we denote the equivalence class of a by $[a]_{\mathcal{E}}$ and the size of this class by $|[a]_{\mathcal{E}}|$.

We meet the requirements:

$$R_{\langle e,i\rangle} : \text{If } \mathcal{A}_e \cong \mathcal{E}_1, \text{ then } \varphi_i \text{ is not an embedding of } \mathcal{A}_e \text{ into } \mathcal{E}_2.$$

The strategy to meet a single requirement $R_{\langle e,i\rangle}$ is as follows. Fix an element $a \in A_e$ (i.e., fix a number) and wait for $\varphi_i(a)$ to converge. Once it converges, we want to force either $|[a]_{\mathcal{A}_e}| > |[\varphi_i(a)]_{\mathcal{E}_2}|$ (assuring φ_i is not an embedding) or $|[a]_{\mathcal{A}_e}| \neq |[k]_{\mathcal{E}_1}|$ for all $k \in E_1$ (assuring $\mathcal{A}_e \not\cong \mathcal{E}_1$).

To make progress on the first option, we freeze the size of $[\varphi_i(a)]$ in $\mathcal{E}_2$. That is, we refrain from ever putting another element into the class of $\varphi_i(a)$. However, this might not be enough as it might currently be the case that $|[a]_{\mathcal{A}_e}| \leq |[\varphi_i(a)]_{\mathcal{E}_2}|$. So, we need to force the size of $[a]_{\mathcal{A}_e}$ to grow. To do this, we add elements to the equivalence classes in $\mathcal{E}_1$ to guarantee that $\mathcal{E}_1$ has no classes of any size n with $|[a]_{\mathcal{A}_e}| \leq n \leq |[\varphi_i(a)]_{\mathcal{E}_2}|$ (using the current values of these classes). That is, for any class in $\mathcal{E}_1$ with a size currently in this range, we add elements to the class so that it becomes strictly larger than $[\varphi_i(a)]_{\mathcal{E}_2}$.

Having taken this action, the requirement $R_{\langle e,i\rangle}$ is satisfied (provided it is not later injured) as the only way to have $\mathcal{A}_e \cong \mathcal{E}_1$ is if $[a]_{\mathcal{A}_e}$ grows to a size larger than the size of $[\varphi_i(a)]_{\mathcal{E}_2}$. However, if it does this, then φ_i cannot be extended to an embedding. Notice that our action to meet $R_{\langle e,i\rangle}$ imposes two restraints on the construction:

- the class $[\varphi_i(a)]_{\mathcal{E}_2}$ cannot grow – which causes no problem since we want the classes in $\mathcal{E}_2$ to be finite and we can add other large classes to make $\mathcal{E}_2$ have unbounded character; and
- all future classes created in $\mathcal{E}_1$ must have size greater than $|[\varphi_i(a)]_{\mathcal{E}_2}|$ – which also causes no problem since we want $\mathcal{E}_1$ to have unbounded character.

To combine multiple requirements, we use a standard finite injury construction. In terms of building $\mathcal{E}_1$ and $\mathcal{E}_2$, at each stage we add a new class of large size to each structure. We will never change the size of an $\mathcal{E}_2$ class, so although we mention freezing the size of such classes, they are really always frozen in the sense that they never grow.

Each requirement $R_{\langle e,i\rangle}$ has a parameter $b_{\langle e,i\rangle}$. When the requirement first acts (or first acts after being initialized), it sets the value of this parameter large. It then waits for an element $a \in A_e$ such that $|[a]_{\mathcal{A}_e}| \geq b_{\langle e,i\rangle}$ and $\varphi_i(a)$ converges. (Notice that once $b_{\langle e,i\rangle}$ settles down, if $\mathcal{A}_e \cong \mathcal{E}_1$, then there must be $\mathcal{A}_e$ classes satisfying this size restriction since $\mathcal{E}_1$ has unbounded character.) If the current values satisfy $|[a]_{\mathcal{A}_e}| > |[\varphi_i(a)]_{\mathcal{E}_2}|$, it freezes $[\varphi_i(a)]_{\mathcal{E}_2}$ and wins. Otherwise, it does the following:

- increases the sizes of the appropriate $\mathcal{E}_1$ classes so that there are no classes of size between $|[a]_{\mathcal{A}_e}|$ and $|[\varphi_i(a)]_{\mathcal{E}_2}|$;
- freezes $[\varphi_i(a)]_{\mathcal{E}_2}$;
- let m be large (in particular, larger than the size of any of the current classes in $\mathcal{E}_1$ including the ones that were just expanded) and declares that all new $\mathcal{E}_1$ classes must have size at least m; and
- initializes all lower priority requirements.

The initialization will force all lower priority requirements to do their diagonalization with classes of size greater than m, so the classes that were just expanded will not be expanded again by a lower priority requirement. Therefore, each $\mathcal{E}_1$ class can only grow finitely often and hence will be finite in the limit. □

6. Boolean Algebras

Here we demonstrate that the class of computable Boolean algebras has the weak range embedding property (and so the weak embedding property) but not the weak domain embedding property (and so not the strong embedding property). We refer the reader to [5] or [7] for background on countable Boolean algebras.

Proposition 6.1. *The class of computable Boolean algebras has the weak range embedding property.*

Proof. Fix computable presentations $\mathcal{B}_1$ and $\mathcal{B}_2$ of computable Boolean algebras such that $\mathcal{B}_1$ classically embeds into $\mathcal{B}_2$. The manner in which we construct the computable presentation $\mathcal{B}_2'$ depends on whether $\mathcal{B}_2$ is superatomic or not.

If $\mathcal{B}_2$ is superatomic, then $\mathcal{B}_1$ is superatomic as it classically embeds into $\mathcal{B}_2$. There are therefore computable ordinals α_1 and α_2 and positive integers m_1 and m_2 such that $\mathcal{B}_1 \cong \mathrm{IntAlg}(\alpha_1 \cdot m_1)$ and $\mathcal{B}_2 \cong \mathrm{IntAlg}(\alpha_2 \cdot m_2)$. If $\alpha_1 = \alpha_2$, it suffices to take $\mathcal{B}_2'$ to be the join of $\mathcal{B}_1$ with $(m_2 - m_1)$-many additional α_1-atoms. If $\alpha_1 < \alpha_2$, it suffices to take $\mathcal{B}_2'$ to be $\mathcal{B}_1 \oplus \mathcal{B}_2$. Of course, as $\mathcal{B}_1$ classically embeds into $\mathcal{B}_2$, it is impossible to have $\alpha_1 > \alpha_2$ or $\alpha_1 = \alpha_2$ and $m_1 > m_2$.

If $\mathcal{B}_2$ is not superatomic, then either $\mathcal{B}_2 \cong \mathrm{IntAlg}(\omega_1^{CK} \cdot (1 + \eta))$ or there is a computable ordinal μ such that $\mathcal{B}_2 \cong \mathcal{B}_2 \oplus \mathrm{IntAlg}(\omega^\mu \cdot (1 + \eta))$. This follows from aspects of Ketonen invariants for countable Boolean algebras. Specifically, it is a direct consequence of the fact that the measure σ for $\mathcal{B}_2$ has a minimal ordinal μ in its range (see [5] or [7] for significant background on measures and why this is the case) and that $\mu \leq \omega_1^{CK}$ as $\mathcal{B}_2$ is computable. In the former case, it suffices to take $\mathcal{B}_2'$ to be a presentation of $\mathrm{IntAlg}((\omega_1^{CK} \cdot (1+\eta)) \cdot (1+\eta))$ where the outer $(1+\eta)$ is *nicely* presented. In the latter case, it suffices to take $\mathcal{B}_2'$ to be a presentation where $\mathrm{IntAlg}(\omega^\mu \cdot (1 + \eta))$ is *nicely* presented. Here, by *nicely* we mean a copy into which the computable atomless algebra computably embeds. For both cases this suffices as $\mathcal{B}_1$ computably embeds into the countable atomless algebra and the countable atomless algebra computably embeds into $\mathcal{B}_2'$. □

Proposition 6.2. *The class of computable Boolean algebras fails to have the weak domain embedding property.*

Proof. It suffices to take the countable atomless algebra for $\mathcal{B}_1$ and the *standard* computable presentation of the Harrison algebra for $\mathcal{B}_2$. More specifically, with $\mathcal{L}$ a computably copy of the Harrison ordering $\omega_1^{CK} \cdot (1+\eta)$ having no infinite computable descending sequences, take $\mathcal{B}_2$ to be the computable presentation obtained from

$\mathrm{IntAlg}(\mathcal{L})$. If there were a computable embedding $\alpha : \mathcal{B}_1 \to \mathcal{B}_2$ (note that $\mathcal{B}_1$ is computably categorical so the choice of presentation does not matter), then there would necessarily be a computable descending chain in $\mathcal{L}$. We reason as follows.

We argue that a computable descending chain $\{y_i\}_{i\geq 1} \subseteq L$ can be constructed from a computable embedding α. With some abuse of notation, we use y_0 to denote ∞ within $\mathcal{L}$. Fix a nonzero element $a_0 \in B_1$. Then, fix a nonzero element $b_0 \in B_1$ with $b_0 <_{\mathcal{B}_1} a_0$. Because $\mathcal{B}_2$ was constructed as the interval algebra of $\mathcal{L}$, it is possible to effectively pass from $\alpha(b_0)$ and $\alpha(a_0 - b_0)$ to the associated finite union of clopen intervals. For y_1, take the rightmost element $y \in L$ such that $y <_{\mathcal{L}} y_0$ and $[x, y)$ appears as a clopen interval in the decomposition of $\alpha(b_0)$ or $\alpha(a_0 - b_0)$. Note that both $\alpha(b_0)$ and $\alpha(a_0 - b_0)$ must be considered as it might be the case, for example, that $\alpha(b_0)$ is representing the interval $[x_0, y_0)$ for some $x_0 \in L$. Note also that the rightmost element $y \in L$ must be taken as it might be the case, for example, that $[x, y) \subseteq \alpha(b)$ for all $b < b_0$. If so, the induction would be prevented from continuing.

For the next step, let a_1 be b_0 if $[x, y)$ was a clopen interval in $\alpha(b_0)$ and let a_1 be $a_0 - b_0$ if $[x, y)$ was a clopen interval in $\alpha(a_0 - b_0)$. Then, fix a nonzero element $b_1 \in B_1$ with $b_1 <_{\mathcal{B}_1} a_1$. For y_2, taken the rightmost element $y \in L$ such that $y <_{\mathcal{L}} y_1$ and $[x, y)$ appears as a clopen interval in the decomposition of $\alpha(b_1)$ or $\alpha(a_1 - b_1)$.

Continuing in this fashion, we can inductively define the sequence $\{y_i\}_{i\geq 1}$. □

7. A Class of Linear Orders

Here we demonstrate the existence of a class of computable algebraic structures having the weak domain embedding property (and so the weak embedding property) but not the weak range embedding property (and so not the strong embedding property). The class will consist of all computable presentations of the linear order isomorphism types $\omega + \omega^*$ and $\omega + 1 + \omega^*$.

Theorem 7.1. *There is a class of computable algebraic structures having the weak domain embedding property but not the weak range embedding property.*

Proof. It suffices to consider the class of all computable presentations of the order types $\omega + \omega^*$ and $\omega + 1 + \omega^*$. We verify it has the weak domain embedding property but not the weak range embedding property.

For the weak domain embedding property, the only nontrivial case to verify is when $\mathcal{L}_1$ is a computable copy of $\omega + \omega^*$ and $\mathcal{L}_2$ is a computable copy of $\omega + 1 + \omega^*$. Take $\mathcal{L}_1'$ to be a decidable copy of $\omega + \omega^*$. By (nonuniformly) fixing the limit point x in $\mathcal{L}_2$, it is possible to computably embed the ω portion of $\mathcal{L}_1$ to the left of x and the ω^* portion of $\mathcal{L}_1$ to the right of x.

For the failure of the weak range embedding property, it suffices to take for $\mathcal{L}_1$ any computable copy of $\omega + \omega^*$ in which neither the ω nor the ω^* is a computable subset of the universe and to take for $\mathcal{L}_2$ any computable copy of $\omega + 1 + \omega^*$. If there were a computable copy $\mathcal{L}_2'$ of $\omega + 1 + \omega^*$ and a computable embedding $\alpha : \mathcal{L}_1 \to \mathcal{L}_2'$, then an element $a \in L_1$ would belong to the ω part if $\alpha(a) <_{\mathcal{L}_2} x$ and belong to the ω^* part if $x <_{\mathcal{L}_2} \alpha(a)$ (again x is the nonuniformly fixed limit point in $\mathcal{L}_2'$). This would contradict that neither ω nor ω^* was computable in $\mathcal{L}_1$. □

8. Algebraically Closed Fields

Here we demonstrate that the class of algebraically closed fields has the weak range embedding property and the weak domain embedding property (and so the weak embedding property) but not the strong embedding property.

Proposition 8.1. *The class of computable algebraically closed fields has the weak range embedding property and the weak domain embedding property.*

Proof. Fix computable presentations $\mathcal{F}_1$ and $\mathcal{F}_2$ of computable algebraically closed fields such that $\mathcal{F}_1$ classically embeds into $\mathcal{F}_2$. Let $\mathcal{F}$ be the common prime subfield of $\mathcal{F}_1$ and $\mathcal{F}_2$ (so $\mathcal{F}$ is either $\mathbb{Q}$ or $\mathbb{Z}_p$ for a prime p). The choice of $\mathcal{F}_1'$ (for the weak domain embedding property) and $\mathcal{F}_2'$ (for the weak range embedding property) depends on the transcendence degree of $\mathcal{F}_1$.

If the transcendence degree of $\mathcal{F}_1$ is infinite, then $\mathcal{F}_1$ and $\mathcal{F}_2$ are isomorphic. Thus if we let $\mathcal{F}_1' = \mathcal{F}_2$, the identity map $\alpha : \mathcal{F}_1' \to \mathcal{F}_2$ witnesses that the weak domain embedding property holds. Alternatively, if we let $\mathcal{F}_2' = \mathcal{F}_1$, the identity map $\alpha : \mathcal{F}_1 \to \mathcal{F}_2'$ witnesses that the weak range embedding property holds.

Now suppose the transcendence degree of $\mathcal{F}_1$ is $n < \infty$. To establish the weak range embedding property, it suffices to take $\mathcal{F}_2'$ to be the computable algebraic closure of the field generated by adjoining an appropriate number of transcendental elements to $\mathcal{F}_1$. For the weak domain embedding property, we take for $\mathcal{F}_1'$ a presentation of $\mathcal{F}_1$ in which $\mathcal{E} = \mathcal{F}(x_0, \ldots, x_{n-1})$ is a computable subset. We define a computable embedding $\alpha : \mathcal{F}_1' \to \mathcal{F}_2$. Nonuniformly pick n algebraically independent elements $y_0, \ldots, y_{n-1}$ of $\mathcal{F}_2$ and define $\alpha(x_i) = y_i$ for $0 \leq i \leq n-1$. Note that this choice determines α on $\mathcal{E}$. What is more, for any element $x \in E$, we can compute $\alpha(x)$ by expressing x in terms of $x_0, \ldots, x_{n-1}$ and elements of $\mathcal{F}$ and then finding the corresponding expression in $\mathcal{F}_2$. In general, the process for computing $\alpha(u)$ for $u \in F_1'$ proceeds as follows. First, find the minimal polynomial $p(t)$ for u over $\mathcal{E}$. This is possible as $\mathcal{E}$ has an effective splitting algorithm. Next, find all the roots of $p(t)$ in $\mathcal{F}_1'$ and all the roots of the corresponding polynomial $\hat{p}(t)$ in $\mathcal{F}_2$. Define $\alpha(u)$ to be any root of $\hat{p}(t)$ not already in the range of α. As $\mathcal{F}_2$ is algebraically closed, there will always be a suitable root for $\alpha(u)$, so in this fashion, we can define α on all of $\mathcal{F}_1'$. □

The failure of the strong embedding property within the class of computable algebraically closed fields is a special case of a result of Metakides and Nerode (see Theorem 4.1 of [6]).

Proposition 8.2. *The class of computable algebraically closed fields fails to have the strong embedding property.*

Proof. It suffices to take for $\mathcal{F}_1$ the natural copy of the algebraic closure of $\mathbb{Q}(x_i)_{i\in\omega}$ and for $\mathcal{F}_2$ a copy of the same having no infinite computably enumerable set of algebraically independent elements (such exist by Theorem 3.1 of [6]). Then there cannot be a computable embedding $\alpha : \mathcal{F}_1 \to \mathcal{F}_2$ as else the hypotheses on $\mathcal{F}_2$ would be violated. □

9. Other Examples and Open Questions

The classes of computable algebraic structures already discussed served to demonstrate that no nontrivial implications hold between the strong embedding property, the weak embedding property, the weak domain embedding property, and the weak range embedding property. There are many other examples that could have been used to make some of these separations. We state additional examples here, none of which are deep.

Proposition 9.1. *If $\mathcal{S}$ is a computably categorical structure, then the class of computable presentations of $\mathcal{S}$ has the strong embedding property.*

The class of computable presentations of the order type ω has the weak domain embedding property and the weak range embedding property (and so the weak embedding property) but not the strong embedding property.

Unfortunately, the classes in Section 4 and Section 7 are not as natural as one might hope. We therefore ask:

Question 9.2. Is there a natural class of computable structures $\mathcal{C}$ having the weak domain embedding property (and so the weak embedding property) but not the weak range embedding property (and so not the strong embedding property)?

Is there a natural class of computable structures $\mathcal{C}$ having the weak embedding property but neither the weak domain embedding property nor the weak range embedding property (and so not the strong embedding property)?

References

[1] Stephen Binns, Bjorn Kjos-Hanssen, Manuel Lerman, James H. Schmerl, and Reed Solomon. Self embeddings of computable trees. *Trans. Amer. Math. Soc.*, 49:1–37, 2008.
[2] Wesley Calvert, Douglas Cenzer, Valentina Harizanov, and Andrei Morozov. Effective categoricity of equivalence structures. *Ann. Pure Appl. Logic*, 141(1-2):61–78, 2006.
[3] Denis R. Hirschfeldt, Bakhadyr Khoussainov, Richard A. Shore, and Arkadii M. Slinko. Degree spectra and computable dimensions in algebraic structures. *Ann. Pure Appl. Logic*, 115(1-3):71–113, 2002.
[4] Asher M. Kach and Joseph S. Miller. Embeddings of computable linear orders. In preparation.
[5] Jussi Ketonen. The structure of countable Boolean algebras. *Ann. of Math. (2)*, 108(1):41–89, 1978.
[6] G. Metakides and A. Nerode. Effective content of field theory. *Ann. Math. Logic*, 17(3):289–320, 1979.
[7] R. S. Pierce. Countable Boolean algebras. In *Handbook of Boolean algebras, Vol. 3*, pages 775–876. North-Holland, Amsterdam, 1989.
[8] Reed Solomon. Π^0_1 classes and orderable groups. *Ann. Pure Appl. Logic*, 115(1-3):279–302, 2002.

School of Mathematics, Statistics, and Operations Research, Victoria University of Wellington, Wellington 6140, NEW ZEALAND
E-mail address: `asher.kach@msor.vuw.ac.nz`

Department of Mathematics and Statistics, Coastal Carolina University, Conway, SC 29526, USA
E-mail address: `olevin@coastal.edu`

Department of Mathematics, University of Connecticut, Storrs, CT 06269, USA
E-mail address: `solomon@math.uconn.edu`