# Degree Spectra of Intrinsically C.E. Relations

Denis R. Hirschfeldt
Department of Mathematics, Cornell University

January 2, 2001

**Abstract**

We show that for every c.e. degree $\mathbf{a} > \mathbf{0}$ there exists an intrinsically c.e. relation on the domain of a computable structure whose degree spectrum is $\{\mathbf{0}, \mathbf{a}\}$. This result can be extended in two directions. First we show that for every uniformly c.e. collection of sets $S$ there exists an intrinsically c.e. relation on the domain of a computable structure whose degree spectrum is the set of degrees of elements of $S$. Then we show that if $\alpha \in \omega \cup \{\omega\}$ then for any $\alpha$-c.e. degree $\mathbf{a} > \mathbf{0}$ there exists an intrinsically $\alpha$-c.e. relation on the domain of a computable structure whose degree spectrum is $\{\mathbf{0}, \mathbf{a}\}$. All of these results also hold for m-degree spectra of relations.

## 1 Introduction

There has been increasing interest over the last few decades in the study of the effective content of mathematics. One field whose effective content has been the subject of a large body of work, dating back at least to the early 1960's, is model theory. (A valuable reference is the handbook [7]. In particular, the introduction and the articles by Ershov and Goncharov and by Harizanov give useful overviews, while the articles by Ash and by Goncharov cover material related to the topic of this paper.)

Several different notions of effectiveness of model-theoretic structures have been investigated. In this paper, we are concerned with structures whose functions and relations are uniformly computable.

---

The results in this paper are part of the author's doctoral dissertation, written at Cornell University under the supervision of Richard A. Shore. The author thanks Professor Shore for many useful comments and suggestions. The author was partially supported by an Alfred P. Sloan Doctoral Dissertation Fellowship.

Current address: School of Mathematical and Computing Sciences, Victoria University, P.O. Box 600, Wellington, New Zealand.

**1.1 Definition.** A structure $\mathcal{A}$ is *computable* if both its domain $|\mathcal{A}|$ and the atomic diagram of $(\mathcal{A}, a)_{a\in|\mathcal{A}|}$ are computable. An isomorphism from a structure $\mathcal{M}$ to a computable structure is called a *computable presentation* of $\mathcal{M}$. (We often abuse terminology and refer to the image of a computable presentation as a computable presentation.) If $\mathcal{M}$ has a computable presentation then it is *computably presentable.*

In model theory, we identify isomorphic structures. From the point of view of computable model theory, however, two isomorphic structures might be very different. For example, it is easy to give two computable presentations of the same group, only one of which has a computable center. We do not wish to say that these two presentations are the same. Thus, for our purposes, studying structures up to isomorphism is not enough. Instead, we study structures up to *computable* isomorphism. This is reflected in the following definition.

**1.2 Definition.** The *computable dimension* of a computably presentable structure $\mathcal{M}$ is the number of computable presentations of $\mathcal{M}$ up to computable isomorphism.

One way in which we may attempt to understand the differences between noncomputably isomorphic computable presentations of a structure $\mathcal{M}$ is to compare (from a computability-theoretic point of view) the images in these presentations of a particular relation on the domain of $\mathcal{M}$. (Of course, this is only interesting if this relation is not the interpretation in $\mathcal{M}$ of a relation in the language of $\mathcal{M}$.) The study of additional relations on computable structures began with the work of Ash and Nerode [2] and has been continued in a large number of papers. (References can be found in the aforementioned articles in [7].)

Ash and Nerode were concerned with relations that maintain some degree of effectiveness in different computable presentations of a structure.

**1.3 Definition.** Let $U$ be a relation on the domain of a computable structure $\mathcal{A}$ and let $\mathfrak{C}$ be a class of relations. $U$ is *intrinsically* $\mathfrak{C}$ on $\mathcal{A}$ if the image of $U$ in any computable presentation of $\mathcal{A}$ is in $\mathfrak{C}$.

In [2], conditions that guarantee that a relation is intrinsically computable or intrinsically computably enumerable (c.e.) were given. More recent work has led to a number of other conditions guaranteeing that a relation is intrinsically $\mathfrak{C}$ for various classes $\mathfrak{C}$ (see [3], for example).

A different way to approach the study of relations on computable structures is to look at the (Turing) degrees of the images of a relation in different computable presentations of a structure.

**1.4 Definition.** Let $U$ be a relation on the domain of a computable structure $\mathcal{A}$. The *degree spectrum* of $U$ on $\mathcal{A}$, $\mathrm{DgSp}_{\mathcal{A}}(U)$, is the set of degrees of the images of $U$ in all computable presentations of $\mathcal{A}$.

Ash-Nerode type conditions usually imply that the degree spectrum of a relation is either a singleton or infinite. Indeed, for various classes of degrees, conditions have been formulated that guarantee that the degree spectrum of a relation consists of all the degrees in the given class (see [1], for example). Motivated by these considerations, as well as by Goncharov's examples [12] of structures of finite computable dimension, Harizanov and Millar suggested the study of relations with finite degree spectra.

Harizanov [8] was the first to give an example of an intrinsically $\Delta^0_2$ relation with a two-element degree spectrum that includes $\mathbf{0}$.

**1.5 Theorem** (Harizanov)**.** *There exist a $\Delta^0_2$ but not c.e. degree $\mathbf{a}$ and a relation $U$ on the domain of a computable structure $\mathcal{A}$ of computable dimension two such that $DgSp_{\mathcal{A}}(U) = \{\mathbf{0}, \mathbf{a}\}$.*

*Remark.* The existence of a relation that is not intrinsically $\Delta^0_2$ and has a two-element degree spectrum that includes $\mathbf{0}$ is a direct consequence of the existence of a rigid structure of computable dimension two. Indeed, suppose that $\mathcal{A}$ is such a computable structure and let $R$ be the binary relation that holds of $x, y \in |\mathcal{A}|$ if and only if $x < y$ and for all $z \in (x, y)$, $z \notin |\mathcal{A}|$. Clearly, if $\mathcal{B}$ is a computable structure and $f : \mathcal{A} \cong \mathcal{B}$ then $\deg(f(R)) = \deg(f)$. So the fact that $\mathcal{A}$ has computable dimension two implies that $\mathrm{DgSp}_{\mathcal{A}}(R) = \{\mathbf{0}, \mathbf{a}\}$ for some degree $\mathbf{a}$. However, by a result of Goncharov [10], $\mathbf{a}$ cannot be $\Delta^0_2$.

It is also easy to give an example of an intrinsically d.c.e. relation with a two-element degree spectrum that does not include $\mathbf{0}$. Let $\mathbf{d}$ be a maximal incomplete d.c.e. degree, as constructed in [5]. (That is, $\mathbf{d} \neq \mathbf{0}'$ is d.c.e. and there are no d.c.e. degrees in $(\mathbf{d}, \mathbf{0}')$.) It is not hard to build computable structures $\mathcal{A}_0$ and $\mathcal{A}_1$ in the language of directed graphs and unary relations $U_0$ and $U_1$ on the domains of $\mathcal{A}_0$ and $\mathcal{A}_1$, respectively, so that the domains of $\mathcal{A}_0$ and $\mathcal{A}_1$ are disjoint, $\mathrm{DgSp}_{\mathcal{A}_0}(U_0)$ is the set of d.c.e. degrees, and $\mathrm{DgSp}_{\mathcal{A}_1}(U_1) = \{\mathbf{d}\}$. Now let $\mathcal{A}$ be the computable structure in the language of directed graphs plus a unary relation $R$ obtained by taking the union of $\mathcal{A}_0$ and $\mathcal{A}_1$ and letting $R$ hold of $x$ if and only if $x \in |\mathcal{A}_0|$, and let $U = U_0 \cup U_1$. It is easy to check that $\mathrm{DgSp}_{\mathcal{A}}(U) = \{\mathbf{b} \mid \mathbf{d} \leqslant \mathbf{b} \text{ and } \mathbf{b} \text{ is d.c.e.}\} = \{\mathbf{d}, \mathbf{0}'\}$.

Khoussainov and Shore and Goncharov [13],[14] showed the existence of an intrinsically c.e. relation with a two-element degree spectrum.

**1.6 Theorem** (Khoussainov and Shore, Goncharov)**.** *There exist a c.e. degree $\mathbf{a}$ and an intrinsically c.e. relation $U$ on the domain of a computable structure $\mathcal{A}$ of computable dimension two such that $DgSp_{\mathcal{A}}(U) = \{\mathbf{0}, \mathbf{a}\}$.*

This left open the question of which degrees can be the nonzero element of a two-element degree spectrum. In this paper we show that, setting aside the issue of computable dimension, each c.e. degree belongs to some two-element degree spectrum whose other element is $\mathbf{0}$.

**1.7 Theorem.** *Let* $\mathbf{a} > \mathbf{0}$ *be a c.e. degree. There exists an intrinsically c.e. relation* $U$ *on the domain of a computable structure* $\mathcal{A}$ *such that* $DgSp_{\mathcal{A}}(U) = \{\mathbf{0}, \mathbf{a}\}$.

The proof of this theorem, which will be given in Section 2, uses techniques from [14], which in turn builds on work of Goncharov [11],[12] and Cholak, Goncharov, Khoussainov, and Shore [4].

In [14], Khoussainov and Shore also proved the following theorem.

**1.8 Theorem** (Khoussainov and Shore)**.** *For each computable poset* $\mathcal{P}$ *there exists an intrinsically c.e. relation* $U$ *on the domain of a computable structure* $\mathcal{A}$ *such that* $\langle DgSp_{\mathcal{A}}(U), \leqslant_T\rangle \cong \mathcal{P}$. *If* $\mathcal{P}$ *has a least element then we can pick* $U$ *and* $\mathcal{A}$ *so that* $\mathbf{0} \in DgSp_{\mathcal{A}}(U)$.

In Section 3, we show how to modify the proof of Theorem 1.7 to prove the following extension of Theorem 1.8.

**1.9 Theorem.** *Let* $\{A_i\}_{i\in\omega}$ *be a uniformly c.e. (u.c.e.) collection of sets. There exists an intrinsically c.e. relation* $U$ *on the domain of a computable structure* $\mathcal{A}$ *such that* $DgSp_{\mathcal{A}}(U) = \{\deg(A_i) \mid i \in \omega\}$.

Another way in which we can extend Theorem 1.7 is by broadening our focus from the c.e. degrees to larger classes of degrees. In Section 4, we show that each of a large class of $\Delta^0_2$ degrees belongs to some two-element degree spectrum whose other element is $\mathbf{0}$.

**1.10 Definition.** Let $A \subseteq \omega$ be a set. A computable sequence $a_0, a_1, \ldots$ is a $\Delta^0_2$ *approximation* of $A$ if for all $x \in \omega$, $|\{s \mid a_s = x\}|$ is finite and $x \in A \Leftrightarrow |\{s \mid a_s = x\}|$ is odd.

Let $n \in \omega$. $A$ is *n-c.e.* if there exists a $\Delta^0_2$ approximation $a_0, a_1, \ldots$ of $A$ such that $|\{s \mid a_s = x\}| \leqslant n$ for all $x \in \omega$.

$A$ is *ω-c.e.* if there exist a $\Delta^0_2$ approximation $a_0, a_1, \ldots$ of $A$ and a computable function $f$ such that $|\{s \mid a_s = x\}| \leqslant f(x)$ for all $x \in \omega$.

Let $\alpha \in \omega \cup \{\omega\}$. A degree is $\alpha$-c.e. if it contains an $\alpha$-c.e. set. A collection of sets $\{A_i\}_{i\in\omega}$ is *uniformly* $\alpha$*-c.e.* if $\bigoplus_{i\in\omega} A_i = \{\langle i, x\rangle \mid x \in A_i\}$ is $\alpha$-c.e..

*Remark.* The above definition of $\omega$-c.e. is the one that will be useful in Section 4. There is an equivalent definition which can be generalized to define the concepts of $\alpha$-c.e. set and $\alpha$-c.e. degree for any computable ordinal $\alpha$ (see [6]).

**1.11 Theorem.** *Let* $\alpha \in \omega \cup \{\omega\}$ *and let* $\mathbf{b} > \mathbf{0}$ *be an* $\alpha$*-c.e. degree. There exists an intrinsically* $\alpha$*-c.e. relation* $V$ *on the domain of a computable structure* $\mathcal{B}$ *such that* $DgSp_{\mathcal{B}}(V) = \{\mathbf{0}, \mathbf{b}\}$.

The structure $\mathcal{B}$ will be an extension of the structure $\mathcal{A}$ constructed in the proof of Theorem 1.7 for an appropriate c.e. degree $\mathbf{a}$.

*Remark.* One interesting consequence of Theorem 1.11 is that there exists a minimal degree $\mathbf{b}$ such that $\{\mathbf{0}, \mathbf{b}\}$ is realized as the degree spectrum of a relation on the domain of a computable structure.

Theorems 1.9 and 1.11 can be conflated to produce the following theorem, which can be proved by combining the modifications to the proof of Theorem 1.7 presented in Sections 3 and 4.

**1.12 Theorem.** *Let $\alpha \in \omega \cup \{\omega\}$ and let $\{A_i\}_{i \in \omega}$ be a uniformly $\alpha$-c.e. collection of sets. There exists an intrinsically $\alpha$-c.e. relation $V$ on the domain of a computable structure $\mathcal{B}$ such that $DgSp_{\mathcal{B}}(V) = \{\deg(A_i) \mid i \in \omega\}$.*

It is also interesting to consider degree spectra of relations with respect to other reducibilities.

**1.13 Definition.** Let r be a reducibility, such as many-one reducibility (m-reducibility) or truth-table reducibility. Let $U$ be a relation on the domain of a computable structure $\mathcal{A}$. The *r-degree spectrum* of $U$ on $\mathcal{A}$, $\mathrm{DgSp}_{\mathrm{r},\mathcal{A}}(U)$, is the set of r-degrees of the images of $U$ in all computable presentations of $\mathcal{A}$.

It will be clear from their proofs that Theorems 1.7 and 1.11 are both true with "degree" replaced by "m-degree" and "$\mathrm{DgSp}_{\mathcal{A}}(U)$" replaced by "$\mathrm{DgSp}_{\mathrm{m},\mathcal{A}}(U)$". Thus, for any reducibility r weaker than m-reducibility, both theorems remain true with "degree" replaced by "r-degree" and "$\mathrm{DgSp}_{\mathcal{A}}(U)$" replaced by "$\mathrm{DgSp}_{\mathrm{r},\mathcal{A}}(U)$". The same holds of Theorems 1.9 and 1.12 if we require that $A_i \neq \emptyset$ and $A_i \neq \omega$ for all $i \in \omega$.

# 2 Proof of Theorem 1.7

**1.7. Theorem.** *Let $\mathbf{a} > \mathbf{0}$ be a c.e. degree. There exists an intrinsically c.e. relation $U$ on the domain of a computable structure $\mathcal{A}$ such that $DgSp_{\mathcal{A}}(U) = \{\mathbf{0}, \mathbf{a}\}$.*

*Proof.* Let $A$ be a c.e. set that is not computable and let $a_0, a_1, \ldots$ be a computable enumeration of $A$. Let $A[0] = \emptyset$, $A[s+1] = \{a_0, \ldots, a_s\}$. We wish to construct computable structures $\mathcal{A}^0$ and $\mathcal{A}^1$ and unary relations $U^0$ and $U^1$ on the domains of $\mathcal{A}^0$ and $\mathcal{A}^1$, respectively, so that the following properties hold.

(2.1) $\mathcal{A}^0 \cong \mathcal{A}^1$ via an isomorphism that carries $U^0$ to $U^1$.

(2.2) $U^0 \equiv_{\mathrm{m}} A$ and $U^1$ is computable.

(2.3) If $\mathcal{G} \cong \mathcal{A}^0$ is a computable structure then the image of $U^0$ in $\mathcal{G}$ is either computable or $m$-equivalent to $A$.

Our structures will be directed graphs. We begin by defining our basic building blocks.

**2.1 Definition.** Let $n \in \omega$. The directed graph $[n]$ consists of nodes $x_0, x_1, \ldots, x_{n+2}$ with an edge from $x_0$ to itself, an edge from $x_{n+2}$ to $x_1$, and an edge from $x_i$ to $x_{i+1}$ for each $i \leqslant n+1$. We call $x_0$ the *top* and $x_{n+2}$ the *coding location* of $[n]$.

Let $S \subset \omega$. The directed graph $[S]$ consists of one copy of $[s]$ for each $s \in S$, with all the tops identified.

Figure 2.1 shows [2] and $[\{2,3\}]$ as examples.

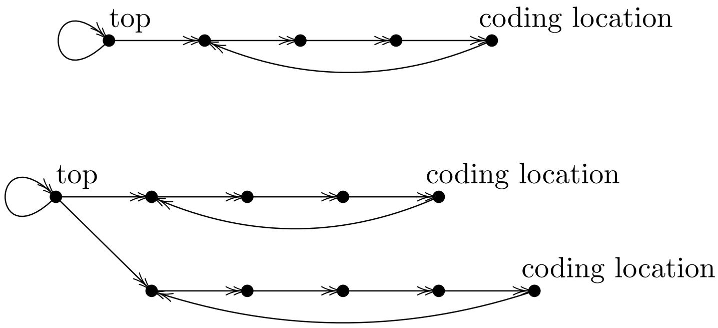


Figure 2.1: [2] and $[\{2,3\}]$

*Remark.* It is convenient to use directed graphs in this proof and in the ones presented in Sections 3 and 4, but in all three cases it would be possible to use undirected graphs (that is, graphs whose edge relations are symmetric) instead. This fact becomes useful in the following situation. Suppose that we wish to prove for a particular theory $T$ that a theorem such as Theorem 1.7 still holds if we require that the structure $\mathcal{A}$ be a model of $T$. One way of doing this is to show that we can code a computable graph that satisfies the given theorem into a computable model $\mathcal{A}$ of $T$ in a way that preserves enough of the properties of the graph to allow us to conclude that the theorem holds of $\mathcal{A}$. However, it will often be easier to code an undirected graph into a model of $T$ than a directed graph. See [9] for details.

Now let us consider how we could go about satisfying (2.1) and (2.2) above. We build $\mathcal{A}^0$ and $\mathcal{A}^1$ in stages. We begin by letting $\mathcal{A}^0_0$ and $\mathcal{A}^1_0$ be computable structures with co-infinite domains, each consisting of one copy of $[k]$ for each $k \in \omega$. If at each stage $s+1$ we enumerate the coding location of the copy of $[3a_s]$ in $\mathcal{A}^0_0$ into $U^0$ then we will have ensured that $U^0 \equiv_{\mathrm{m}} A$. However, we also wish to make $U^1$ computable while guaranteeing that $\mathcal{A}^0 \cong \mathcal{A}^1$ via an isomorphism that carries $U^0$ to $U^1$. To describe how we can do this, we need two more definitions.

**2.2 Definition.** Let $\mathcal{G}$ be a computable structure in the language of directed graphs whose domain is co-infinite. $\mathcal{G}$ consists of the disjoint union of a number of connected components, which from now on we will just call the *components* of $\mathcal{G}$.

Suppose that $\mathcal{G}$ has components $K$ and $L$ isomorphic to $[B]$ and $[C]$, respectively, where $B, C \subset \omega$ are finite. We define the operation $K \cdot L$, which takes $\mathcal{G}$ to a new computable structure extending $\mathcal{G}$, as follows. Extend $K$ to be a copy of $[B \cup C]$ using numbers not in the domain of $\mathcal{G}$. Leave every other component of $\mathcal{G}$ (including $L$) unchanged.

We will also use the notation $K \cdot L$ to denote the graph $[B \cup C]$. It should always be clear which meaning of $K \cdot L$ is being used.

Given a finite sequence of operations, each of which can be applied to $\mathcal{G}$, so that no two operations in the sequence affect the same component of $\mathcal{G}$, we can apply all of the operations in the sequence simultaneously to $\mathcal{G}$ to get a structure extending $\mathcal{G}$. In this case we will say that we have applied the sequence of operations to $\mathcal{G}$.

**2.3 Definition.** Let $\mathcal{G}$ be a computable structure in the language of directed graphs whose domain is co-infinite and let $X_0, \ldots, X_n$ be components of $\mathcal{G}$ such that for each $i \leqslant n$, $X_i$ is isomorphic to $[S_i]$ for some finite $S_i \subset \omega$. We define two operations, each of which takes $\mathcal{G}$ to a new computable structure extending $\mathcal{G}$.

- The **L**-operation $\mathbf{L}(X_0, \ldots, X_n)$ consists of applying the sequence of operations $X_0 \cdot X_1,\ X_1 \cdot X_2,\ \ldots,\ X_n \cdot X_0$ to $\mathcal{G}$.
- The **R**-operation $\mathbf{R}(X_0, \ldots, X_n)$ consists of applying the sequence of operations $X_0 \cdot X_n,\ X_1 \cdot X_0,\ \ldots,\ X_n \cdot X_{n-1}$ to $\mathcal{G}$.

Note that if $\mathcal{H}$ is the structure obtained by applying $\mathbf{L}(X_0, \ldots, X_n)$ to $\mathcal{G}$ and $\mathcal{H}'$ is the structure obtained by applying $\mathbf{R}(X_0, \ldots, X_n)$ to $\mathcal{G}$ then $\mathcal{H} \cong \mathcal{H}'$.

We can now proceed as follows. At stage $s+1$, let $X_s^i$, $Y_s^i$, and $Z_s^i$ be the copies in $\mathcal{A}_s^i$ of $[3a_s]$, $[3a_s+1]$, and $[3a_s+2]$, respectively. Perform $\mathbf{L}(Y_s^0, X_s^0, Z_s^0)$ on $\mathcal{A}_s^0$ to get $\mathcal{A}_{s+1}^0$ and perform $\mathbf{R}(Y_s^1, X_s^1, Z_s^1)$ on $\mathcal{A}_s^1$ to get $\mathcal{A}_{s+1}^1$. (In order to ensure that $\mathcal{A}^0$ and $\mathcal{A}^1$ are computable, the new numbers added to their domains at this stage are assumed to be greater than $s$.) Put the coding location of the old copy of $[3a_s]$ in $\mathcal{A}_{s+1}^0$ (that is, the copy that was already in $\mathcal{A}_0^0$) into $U^0$ and put the coding location of the new copy of $[3a_s]$ in $\mathcal{A}_{s+1}^1$ into $U^1$. (Figure 2.2 pictures what happens on either side of the construction. For each $i = 0, 1$, the copy of $[3a_s]$ whose coding location enters $U^i$ is underlined.)

Now let $\mathcal{A}^0 = \bigcup_{s \in \omega} \mathcal{A}_s^0$ and $\mathcal{A}^1 = \bigcup_{s \in \omega} \mathcal{A}_s^1$. It is easy to show, by induction using the definition of the **L**- and **R**-operations, that for each $s$, $\mathcal{A}_s^0 \cong \mathcal{A}_s^1$ via an isomorphism that carries $U^0[s]$ to $U^1[s]$. (Here $U^i[s]$ is the set of all numbers that have entered $U^i$ by the end of stage $s$.) Furthermore, whenever a component of $\mathcal{A}_s^i$ participates in an operation at stage $s+1$, so does the isomorphic component of $\mathcal{A}_s^{1-i}$. Since $\mathcal{A}^0$ and $\mathcal{A}^1$ have no infinite components, it follows that $\mathcal{A}^0 \cong \mathcal{A}^1$ via an isomorphism that carries $U^0$ to $U^1$.

Furthermore, it is still true that $U^0 \equiv_{\mathrm{m}} A$, since a number is in $U^0$ if and only if it is the coding location of the copy of $[3a]$ in $\mathcal{A}_0^0$ for some $a \in A$. On the other hand, any

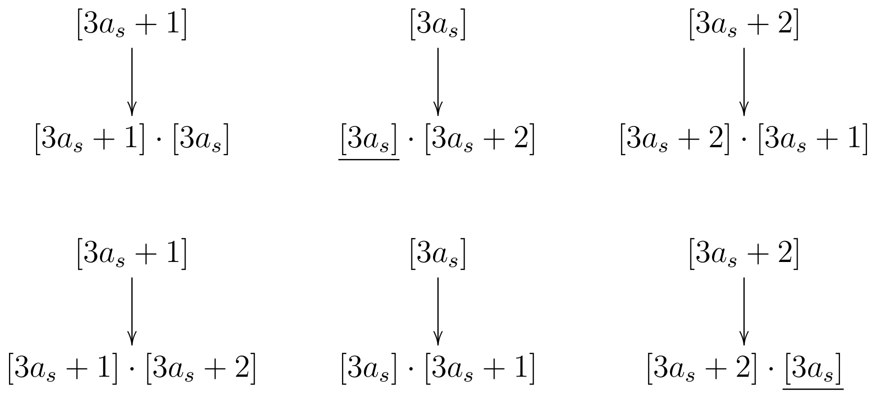


Figure 2.2: The Basic Coding Strategy (top: $\mathcal{A}^0$ / bottom: $\mathcal{A}^1$)

number put into $U^1$ at a stage $s+1$ is a new number, and is therefore greater than $s$. Thus $U^1$ is computable.

So we see that the above construction is enough to satisfy (2.1) and (2.2). We now consider how to satisfy (2.3). Let us begin by attempting to satisfy this property for a particular computable structure $\mathcal{G}$. That is, we want to ensure that if $\mathcal{G} \cong \mathcal{A}^0$ then the image of $U^0$ in $\mathcal{G}$ is either computable or $m$-equivalent to $A$. The way in which we do this is based on the following observation.

Let $U$ be the image of $U^0$ in $\mathcal{G}$ and let $\mathcal{G}[s]$ denote the stage $s$ approximation to $\mathcal{G}$. Assume that for all $s \in \omega$, $\mathcal{A}^0_s$, $\mathcal{A}^1_s$, and $\mathcal{G}[s]$ have no non-trivial automorphisms.

Suppose that at some stage $s$, $\mathcal{A}^0_s$ has components $X^0_s$, $Y^0_s$, $Z^0_s$, and $S^0_s$, $\mathcal{A}^1_s$ has isomorphic components $X^1_s$, $Y^1_s$, $Z^1_s$, and $S^1_s$, respectively, and $\mathcal{G}[s]$ has isomorphic components $X_s$, $Y_s$, $Z_s$, and $S_s$, respectively. Now suppose we perform $\mathbf{L}(Y^0_s, X^0_s, Z^0_s, S^0_s)$ on $\mathcal{A}^0_s$ to get $\mathcal{A}^0_{s+1}$ and perform $\mathbf{R}(Y^1_s, X^1_s, Z^1_s, S^1_s)$ on $\mathcal{A}^1_s$ to get $\mathcal{A}^1_{s+1}$. Then $\mathcal{A}^0_{s+1}$ has components isomorphic to $S^0_s \cdot Y^0_s$, $Y^0_s \cdot X^0_s$, $X^0_s \cdot Z^0_s$, and $Z^0_s \cdot S^0_s$, and these are the only components of $\mathcal{A}^0_{s+1}$ that contain copies of $X^0_s$, $Y^0_s$, $Z^0_s$, or $S^0_s$. So if $X_s$, $Y_s$, $Z_s$, and $S_s$ do not grow into isomorphic copies of the aforementioned components of $\mathcal{A}^0_{s+1}$ then we can win immediately by not involving these components in any further operations, thus guaranteeing that $\mathcal{G} \ncong A^0$.

So if $\mathcal{G} \cong A^0$ then there are only two possibilities. The first is that $S_s$ grows into a copy of $S_s \cdot Y_s$, $Y_s$ grows into a copy of $Y_s \cdot X_s$, $X_s$ grows into a copy of $X_s \cdot Z_s$, and $Z_s$ grows into a copy of $Z_s \cdot S_s$. In this case we will say that $\mathcal{G}$ "goes to the left". The other possibility is that $Y_s$ grows into a copy of $S_s \cdot Y_s$, $S_s$ grows into a copy of $Z_s \cdot S_s$, $Z_s$ grows into a copy of $X_s \cdot Z_s$, and $X_s$ grows into a copy of $Y_s \cdot X_s$. In this case we will say that $\mathcal{G}$ "goes to the right".

Now, if the coding location of $X^0_s$ is put into $U^0$ and the coding location of the new copy of $X^1_s$ is put into $U^1$ then the coding location of the copy of $X_s$ that is part of the component isomorphic to $X_s \cdot Z_s$ is in $U$. In other words, if $\mathcal{G}$ goes to the left then the coding location of $X_s$ in $\mathcal{G}[s]$ is in $U$, while if $\mathcal{G}$ goes to the right then the coding location of the copy of $X_s$ in $\mathcal{G} - \mathcal{G}[s]$ is in $U$. It is easy to conclude from this that if $\mathcal{G}$ goes to the left at all but finitely many stages then $U \equiv_m A$, while if $\mathcal{G}$ goes to the right at all but finitely many stages then $U$ is computable.

So it is enough to ensure that $\mathcal{G}$ either almost always goes to the left or almost always goes to the right. This can be done by always using the same component of $\mathcal{G}$, which we will call the *special component* of $\mathcal{G}$, as $S_s$.

That is, we first pick some component of $\mathcal{G}$ to be its special component. Say we pick the one that extends the first copy of $[0]$ to appear in $\mathcal{G}$. (Let us assume that $0 \notin A$.) At stage 0, we define $\mathcal{A}_0^i$ as above and wait until a copy of $[0]$ is enumerated into $\mathcal{G}$. We also define $r_0$ to be 0. The value of $r_s$ will code whether $\mathcal{G}$ goes to the left or to the right at stage $s$.

At stage $s+1$, we let $X_s^i$, $Y_s^i$, and $Z_s^i$ be the copies in $\mathcal{A}_s^i$ of $[3a_s]$, $[3a_s+1]$, and $[3a_s+2]$, respectively, and let $S_s^i$ be the isomorphic copy in $\mathcal{A}_s^i$ of the special component $S_s$ of $\mathcal{G}[s]$. We wait until copies of $X_s^i$, $Y_s^i$, and $Z_s^i$ are enumerated into $\mathcal{G}[s]$ and then perform the same operations as before. We then wait until copies of $S_s \cdot Y_s$, $Y_s \cdot X_s$, $X_s \cdot Z_s$, and $Z_s \cdot S_s$ are enumerated into $\mathcal{G}$. Either the copy of $S_s \cdot Y_s$ or that of $Z_s \cdot S_s$ will extend $S_s$. Whichever one it is now becomes $S_{s+1}$. If $S_{s+1} \cong S_s \cdot Y_s$ then $r_{s+1} = 0$; otherwise $r_{s+1} = 1$.

The above construction ensures that if $\mathcal{G} \cong A^0$ then the special component of $\mathcal{G}$ is infinite. On the other hand, it is not hard to check that it also guarantees that if $\mathcal{G}$ changes direction infinitely often (that is, if $r_s$ does not have a limit) then no component of $\mathcal{A}^0$ is infinite, so that $\mathcal{G} \ncong A^0$. This is because, for each $s \in \omega$, the copy of the special component of $\mathcal{G}[s+1]$ in $\mathcal{A}_{s+1}^{1-r_{s+1}}$ is a component that participates in an operation for the first time at stage $s+1$.

However, there are two problems with this construction. First of all, it is easy to check that if $\mathcal{G}$ almost always goes to the left then no component of $\mathcal{A}^1$ is infinite, while if $\mathcal{G}$ almost always goes to the right then no component of $\mathcal{A}^0$ is infinite. In either case, (2.1) no longer holds.

We solve this by re-using components in operations. The idea is roughly as follows. Instead of using four components in our operations, we use six. That is, at stage $s+1$, in addition to the components mentioned above, we pick two other components $B_s^0$ and $C_s^0$ of $\mathcal{A}_s^0$ and isomorphic components $B_s^1$ and $C_s^1$ of $\mathcal{A}_s^1$, perform $\mathbf{L}(Y_s^0, X_s^0, Z_s^0, B_s^0, S_s^0, C_s^0)$ on $\mathcal{A}_s^0$ to get $\mathcal{A}_{s+1}^0$, and perform $\mathbf{R}(Y_s^1, X_s^1, Z_s^1, B_s^1, S_s^1, C_s^1)$ on $\mathcal{A}_s^1$ to get $\mathcal{A}_{s+1}^1$. (In order to accommodate the extra components, $X_s^i$ will be the copy of $[6a_s]$ in $\mathcal{A}_s^i$. A similar change will be made for the other components.)

As long as $\mathcal{G}$ is going in the same direction, we designate every other stage as an *isomorphism recovery stage*. At such a stage $s+1$, if $r_s = 0$ then we let $C_s^0$ be the component of $\mathcal{A}_s^0$ that extends $B_{s-1}^0$ and let $C_s^1$ be the isomorphic component of $\mathcal{A}_s^1$. On the other hand, if $r_s = 1$ then we let $B_s^1$ be the component of $\mathcal{A}_s^1$ that extends $C_{s-1}^1$ and let $B^0_s$ be the isomorphic component of $\mathcal{A}^0_s$. Whenever $\mathcal{G}$ changes direction, we restart this isomorphism recovery process.

It is straightforward to check that this strategy guarantees that if $r_s$ has a limit then the copies of the special component of $\mathcal{G}$ in $\mathcal{A}^0$ and $\mathcal{A}^1$ are isomorphic, while still ensuring that if $r_s$ does not have a limit then no component of $\mathcal{A}^0$ or $\mathcal{A}^1$ is infinite. We will give an example below to illustrate isomorphism recovery.

Another problem that we must face in the full construction is that, in general, we can not know in advance whether a given computable structure $\mathcal{G}$ is isomorphic to $\mathcal{A}^0$. So it is not possible to wait at each stage until the appropriate components are enumerated into $\mathcal{G}$. To get around this, we introduce the notion of a *recovery stage*.

At stage $s+1$, where we would have waited for $\mathcal{G}$ to provide components $Y_s$, $X_s$, $Z_s$, $B_s$, and $C_s$, we now simply do not involve copies of the special component of $\mathcal{G}$ in our operations unless these components are provided. (That is, if these components are not in $\mathcal{G}[s]$ then we perform $\mathbf{L}(Y^0_s, X^0_s, Z^0_s)$ on $\mathcal{A}^0_s$ to get $\mathcal{A}^0_{s+1}$ and perform $\mathbf{R}(Y^1_s, X^1_s, Z^1_s)$ on $\mathcal{A}^1_s$ to get $\mathcal{A}^1_{s+1}$.) Furthermore, where we would have waited for $Y_s$, $X_s$, $Z_s$, $B_s$, $S_s$, and $C_s$ to grow into copies of $Y_s \cdot X_s$, $X_s \cdot Z_s$, $Z_s \cdot B_s$, $B_s \cdot S_s$, $S_s \cdot C_s$, and $C_s \cdot Y_s$, we now just declare that we are waiting for these copies to appear in $\mathcal{G}$.

A recovery stage is then a stage $s+1$ such that

1. $\mathcal{G}[s]$ contains copies of all the components for which we are currently waiting and

2. for each $j \notin A[s]$ that is less than or equal to the number of recovery stages before stage $s+1$, $\mathcal{G}[s]$ contains components that can be used as $Y_t$, $X_t$, $Z_t$, $B_t$, and $C_t$ if $a_t = j$ for some $t > s$.

(These conditions will be made more precise in the full construction, which will be presented shortly.)

Now suppose that $\mathcal{G} \cong \mathcal{A}^0$. Say that $\mathcal{G}$ is *active* at a given stage if isomorphic copies of its special component participate in the operations performed at that stage. We want there to be infinitely many recovery stages. This will happen as long as there is a bound on how often $\mathcal{G}$ can be active while waiting for recovery.

Let $P$ be the set of all $j \in \omega$ that do not enter $A$ before the $j^{\text{th}}$ recovery stage. Let $M$ be the set of all coding locations of copies of $[6j]$, $j \in P$, in $\mathcal{G}$ and let $N$ be the set of all coding locations of copies of $[6j]$, $j \notin P$, in $\mathcal{G}$. By the definition of recovery stage, $\mathcal{G}$ will be active at each stage $s+1$ such that $a_s \in P$. We make it a rule that $\mathcal{G}$ is not active at any other stage. This clearly provides the desired bound on the number of times $\mathcal{G}$ can be active while waiting for recovery.

Arguing as before, we conclude that if $\mathcal{G}$ almost always goes to the left then $U \cap M \equiv_{\mathrm{m}} A$, while if $\mathcal{G}$ almost always goes to the right then $U \cap M$ is computable. But $P$, $N$, and $U \cap N$ are computable, since if we wait until the $j^{\text{th}}$ recovery stage then we can tell whether $j \in P$, and if $j \notin P$ then $j \in A$. So if $\mathcal{G}$ almost always goes to the left then $U \equiv_{\mathrm{m}} A$, while if $\mathcal{G}$ almost always goes to the right then $U$ is computable. Thus (2.3) is satisfied for this $\mathcal{G}$.

We remark that the modification to the construction that we have just described makes the definition of isomorphism recovery stage a little more complicated, in that we will not want a stage to be an isomorphism recovery stage unless it is a *first stage*, that is, the first stage at which $\mathcal{G}$ is active after a recovery stage. We will discuss this further below.

Before proceeding, let us look at two examples. The first one illustrates what happens when $\mathcal{G}$ recovers. Suppose that $s < t < u < v$ are such that $s+1$ is a first stage, $r_{s+1} = 0$, $v + 1$ is the next recovery stage after stage $s + 1$, and $t + 1$ and $u + 1$ are the only two stages between stages $s + 1$ and $v + 1$ at which $\mathcal{G}$ is active. Figure 2.3 pictures what happens on the $\mathcal{A}^0$ side of the construction. In this figure and in the next one, the notation $R^i_s$ is used in place of $S^i_s$, since this is the notation that we will adopt in the full construction. This change is made because $R^i_w$ might not be isomorphic to the special component of $\mathcal{G}[w]$ if $w + 1$ is not a recovery stage.

Note that, by the definition of recovery stage, the special component of $\mathcal{G}[s]$ is isomorphic to $R^0_s$ and, for each $w = s, t, u$, $\mathcal{G}[s]$ has components $Y_w$, $X_w$, $Z_w$, $B_w$, and $C_w$ isomorphic to $Y^0_w$, $X^0_w$, $Z^0_w$, $B^0_w$, and $C^0_w$, respectively.

Since $\mathcal{G}$ recovers at stage $v + 1$, there are two possibilities. The first one is that the special component of $\mathcal{G}[v]$ is isomorphic to one of $B^0_s \cdot R^0_s$, $B^0_t \cdot R^0_s \cdot C^0_s$, or $B^0_u \cdot R^0_s \cdot C^0_s \cdot C^0_t$. In this case, $r_{v+1} = 1$.

The second possibility is that the special component of $\mathcal{G}[v]$ is isomorphic to $R^0_s \cdot C^0_s \cdot C^0_t \cdot C^0_u$. In this case, the component of $\mathcal{G}[v]$ that extends $C_u$ must be the one isomorphic to $C^0_u \cdot Y^0_u$. From this it follow that the component of $\mathcal{G}[v]$ that extends $Y_u$ must be the one isomorphic to $Y^0_u \cdot X^0_u$. Proceeding in this fashion, we see that for each $w = s, t, u$, the component of $\mathcal{G}[v]$ that extends $X_w$ is the one isomorphic to $X^0_w \cdot Z^0_w$.

Notice that in the previous argument it is crucial that no component of $\mathcal{A}^0$ other than the one that extends $R^0_s$ participates in operations more than once in the interval $(s, v]$. This is the reason for requiring that isomorphism recovery happen only at first stages.

Our second example illustrates isomorphism recovery. Suppose that $s < t < u < v < w$ are such that $s + 1$ and $v + 1$ are first stages, $t + 1$ and $u + 1$ are the only stages between $s + 1$ and $v + 1$ at which $\mathcal{G}$ is active, and $w + 1$ is the first stage after stage $v + 1$ at which $\mathcal{G}$ is active. Suppose further that $r_{s+1} = r_{t+1} = r_{u+1} = r_{v+1} = r_{w+1} = 0$. Figure 2.4 pictures what happens on either side of the construction. The key point to notice here is that if $R^0_t \cong R^1_t$ then $R^0_w$ extends $R^0_t$, $R^1_w$ extends $R^1_t$, and $R^0_w \cong R^1_w$. This pattern would allow us to prove by induction that if $r_s$ has a limit then each $\mathcal{A}^i$ has a unique infinite component $S^i$ and $S^0 \cong S^1$.

In the full construction, we will of course need to satisfy (2.3) for every computable directed graph. Let $\mathcal{G}_0, \mathcal{G}_1, \ldots$ be a standard enumeration of all partial computable directed graphs. In our construction, we will define the concepts of $n$-recovery stage, $n$-isomorphism recovery stage, and so forth.

Let us first clarify what we mean by a partial computable directed graph. (Here we

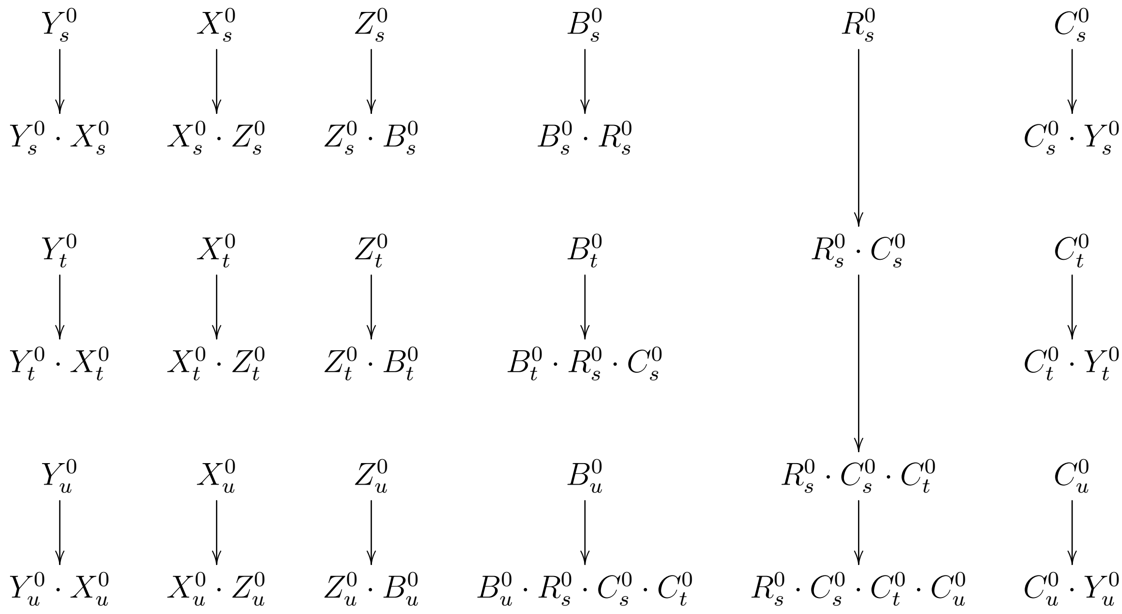


Figure 2.3: Recovery

use standard notation, as in [15].) A partial computable directed graph $\mathcal{G}$ consists of two $0, 1$-valued partial computable functions $\Phi_e$ and $\Phi_i$, the former unary and the latter binary, such that if $\Phi_{e,s}(x)\downarrow = \Phi_{e,s}(y)\downarrow = 1$ then $\Phi_{i,s}(x, y)\downarrow$. The graph $\mathcal{G}$ (resp. $\mathcal{G}[s]$) is the graph whose domain has characteristic function $\Phi_e$ ($\Phi_{e,s}$) and whose edge relation has characteristic function $\Phi_i$ ($\Phi_{i,s}$).

We will be able to satisfy (2.3) for each $\mathcal{G}_n$ independently. We first need some notation to allow us to distinguish the components that are used to satisfy (2.3) for a particular $\mathcal{G}_n$. Fix some one-to-one function from $\omega \times \omega$ onto $\omega$ and let $\langle a, b\rangle$ denote the image under this function of the ordered pair consisting of $a \in \omega$ and $b \in \omega$.

**2.4 Definition.** Let $\mathcal{G}$ be a directed graph. We denote by $(\mathcal{G})_n$ the subgraph of $\mathcal{G}$ consisting of those components $C$ of $\mathcal{G}$ that satisfy both of the following conditions.

1. $C$ is not isomorphic to $[x]$ for any $x \in \omega$.
2. $C$ contains either a copy of $[6n + 3]$ or a copy of $[6\langle n, j\rangle + l]$ for some $j \in \omega$, $l \in \{1, 2, 4, 5\}$.

The idea is that the components of $(\mathcal{A}^i)_n$ are the ones used in the construction to satisfy (2.3) for $\mathcal{G}_n$, and that $(\mathcal{A}^i_s)_n$ is the subgraph of $\mathcal{A}^i_s$ consisting of all such components that have participated in operations before stage $s + 1$.

We also need new **L**- and **R**-operations in order to involve components of $(\mathcal{A}^i)_n$ for different $n$'s in operations at the same stage.

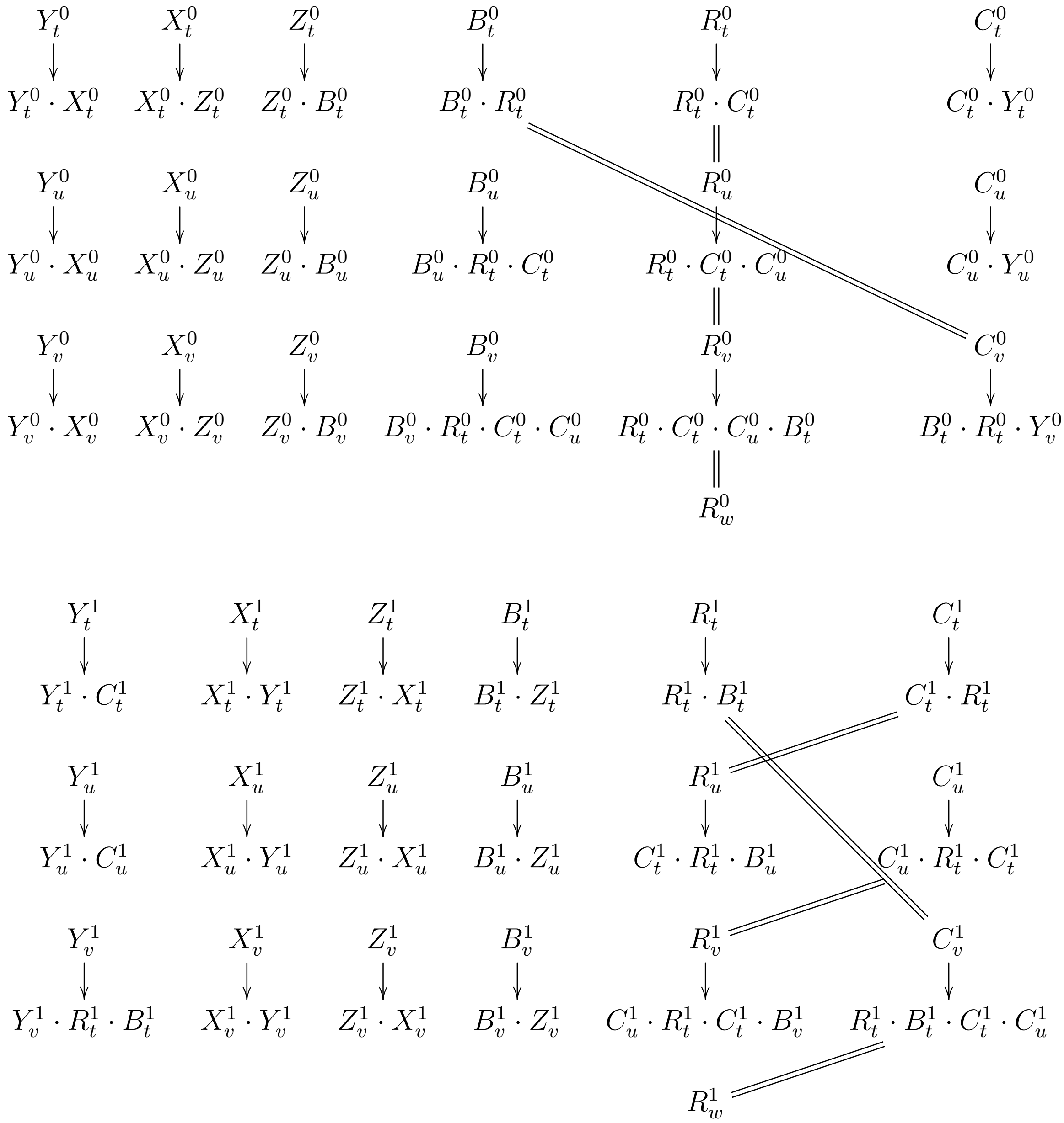


Figure 2.4: Isomorphism Recovery (top: $\mathcal{A}^0$ / bottom: $\mathcal{A}^1$)

**2.5 Definition.** Let $\mathcal{G}$ be a computable structure in the language of directed graphs whose domain is co-infinite. Let $K_0, K_1, \ldots, K_n$ and $L$ be components of $\mathcal{G}$ isomorphic to $[y_0], [y_1], \ldots, [y_n]$ and $[x]$, respectively, where $y_0, y_1, \ldots, y_n, x \in \omega$. We define two operations, each of which takes $\mathcal{G}$ to a new computable structure extending $\mathcal{G}$.

- The operation $(K_0, K_1, \ldots, K_n) \cdot L$ consists of performing the following steps, and otherwise leaving $\mathcal{G}$ unchanged. Create a new copy of $[x]$ using numbers not in the domain of $\mathcal{G}$. For each $i \leqslant n$, add an edge from the top of this new copy of $[x]$ to the top of $K_i$.

- The operation $L \cdot (K_0, K_1, \ldots, K_n)$ consists of performing the following steps, and otherwise leaving $\mathcal{G}$ unchanged. For each $i \leqslant n$, create a new copy of $[y_i]$ using numbers not in the domain of $\mathcal{G}$. For each $i \leqslant n$, add an edge from the top of $L$ to the top of the new copy of $[y_i]$.

For example, suppose that $L$, $K_0$, and $K_1$ are copies of [2], [3], and [4], respectively. Then the operation $(K_0, K_1) \cdot L$ consists of extending $K_0 \cup K_1$ to a copy of the graph shown in Figure 2.5, while the operation $L \cdot (K_0, K_1)$ consists of extending $L$ to a copy of that same graph.

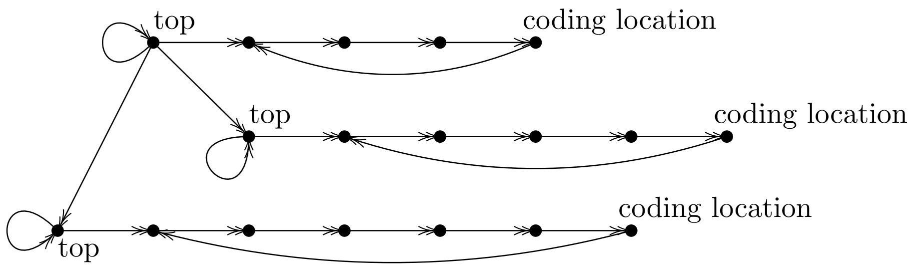


Figure 2.5: The result of either of the operations $([3], [4]) \cdot [2]$ or $[2] \cdot ([3], [4])$

**2.6 Definition.** Let $\mathcal{G}$ be a computable structure in the language of directed graphs whose domain is co-infinite. We say that a component $C$ of $\mathcal{G}$ is a *set component* if it is isomorphic to $[T]$ for some finite $T \subset \omega$. If $T$ is a singleton then we say that $C$ is a *singleton component*.

Let $Y_0, \ldots, Y_n$, $X$, $Z_0, \ldots, Z_n$, $B_0, \ldots, B_n$, $S_0, \ldots, S_n$, and $C_0, \ldots, C_n$ be components of $\mathcal{G}$ such that for each $i \leqslant n$, $X$, $Y_i$, and $Z_i$ are singleton components and $B_i$, $S_i$, and $C_i$ are set components. We define two operations, each of which takes $\mathcal{G}$ to a new computable structure extending $\mathcal{G}$.

- The **L**-*operation*

$$\mathbf{L}(Y_0, \ldots, Y_n; X; Z_0, \ldots, Z_n; B_0, S_0, C_0; \ldots; B_n, S_n, C_n)$$

consists of applying the following sequence of operations to $\mathcal{G}$.

$$(Y_0,\dots,Y_n)\cdot X,\ X\cdot(Z_0,\dots,Z_n),\ Z_0\cdot B_0,\ \dots,\ Z_n\cdot B_n,$$
$$B_0\cdot S_0,\ \dots,\ B_n\cdot S_n,\ S_0\cdot C_0,\ \dots,\ S_n\cdot C_n,\ C_0\cdot Y_0,\ \dots,\ C_n\cdot Y_n$$

- The **R**-*operation*

$$\mathbf{R}(Y_0,\dots,Y_n;X;Z_0,\dots,Z_n;B_0,S_0,C_0;\dots;B_n,S_n,C_n)$$

consists of applying the following sequence of operations to $\mathcal{G}$.

$$Y_0\cdot C_0,\ \dots,\ Y_n\cdot C_n,\ C_0\cdot S_0,\ \dots,\ C_n\cdot S_n,\ S_0\cdot B_0,\ \dots,\ S_n\cdot B_n,$$
$$B_0\cdot Z_0,\ \dots,\ B_n\cdot Z_n,\ (Z_0,\dots,Z_n)\cdot X,\ X\cdot(Y_0,\dots,Y_n)$$

Note that if $\mathcal{H}$ is the structure obtained by applying

$$\mathbf{L}(Y_0,\dots,Y_n;X;Z_0,\dots,Z_n;B_0,S_0,C_0;\dots;B_n,S_n,C_n)$$

to $\mathcal{G}$ and $\mathcal{H}'$ is the structure obtained by applying

$$\mathbf{R}(Y_0,\dots,Y_n;X;Z_0,\dots,Z_n;B_0,S_0,C_0;\dots;B_n,S_n,C_n)$$

to $\mathcal{G}$ then $\mathcal{H}\cong\mathcal{H}'$.

We now proceed with the construction of $\mathcal{A}^0$, $\mathcal{A}^1$, $U^0$, and $U^1$. For each $i=0,1$, we first define a computable structure $\mathcal{A}^i_0$. At each stage $s+1$, we perform an operation on $\mathcal{A}^i_s$ to get $\mathcal{A}^i_{s+1}\supset\mathcal{A}^i_s$ and add an element of the domain of $\mathcal{A}^i_{s+1}$ to $U^i$. We then let $\mathcal{A}^i=\bigcup_{s\in\omega}\mathcal{A}^i_s$. In order to guarantee that $\mathcal{A}^i$ is computable, we make it a convention that all numbers added to the domain of $\mathcal{A}^i_s$ at stage $s+1$ to get $\mathcal{A}^i_{s+1}$ are greater than $s$.

Let $t\geqslant s$. We say that a component $L$ of $\mathcal{A}^i_t$ or $\mathcal{A}^i$ (resp. $\mathcal{G}_n[t]$ or $\mathcal{G}_n$) *extends* a component $K$ of $\mathcal{A}^i_s$ ($\mathcal{G}_n[s]$) if the domain of $K$ is contained in the domain of $L$, and that $L$ *properly extends* $K$ if this containment is proper. (Note that "$L$ extends $K$" means more than just that $K$ can be embedded in $L$, though it of course implies the latter.) If $L$ extends $K$ but not properly then we say that $L$ is a component of $\mathcal{A}^i_s$ ($\mathcal{G}_n[s]$).

It will be the case that if $K$ and $L$ are distinct components of $\mathcal{A}^0_s$ and $K$ is not a copy of $[6k+1]$ or $[6k+2]$ for any $k\in\omega$ then $K$ and $L$ are not extended by the same component of $\mathcal{A}^0$. Thus, since we are not interested in $\mathcal{G}_n$ unless it is isomorphic to $\mathcal{A}^0$, we can assume without loss of generality that, for each $n,s\in\omega$, there is an embedding of $\mathcal{G}_n[s]$ into $\mathcal{A}^0_s$ such that if $K$ and $L$ are distinct components of $\mathcal{G}_n[s]$ and $K$ is not a copy of $[6k+1]$ or $[6k+2]$ for any $k\in\omega$ then $K$ and $L$ are mapped into distinct components of $\mathcal{A}^0_s$.

Suppose there is a least stage $s$ such that $\mathcal{G}_n[s]$ has a component $K$ isomorphic to $[6n+3]$ and let $t\geqslant s$. We call the component of $\mathcal{G}_n[t]$ (resp. $\mathcal{G}_n$) that extends $K$ the *special component* of $\mathcal{G}_n[t]$ ($\mathcal{G}_n$).

It will be easy to check as we go along that the following are properties of the construction.

1. For each $s \in \omega$, $\mathcal{A}^0_s \cong \mathcal{A}^1_s$ and no component of $\mathcal{A}^i_s$ is embeddable in another component of $\mathcal{A}^i_s$.

2. Let $t < s$. No component of $\mathcal{A}^i_t$ isomorphic to one of $[6a_s]$ or $[6\langle n, a_s\rangle + l]$, $l \in \{1, 2, 4, 5\}$, $n \in \omega$, participates in an operation at stage $t + 1$.

*stage* 0. Let $\mathcal{A}^0_0$ and $\mathcal{A}^1_0$ be computable structures with co-infinite domains, each consisting of one copy of $[k]$ for each $k \in \omega$. For each $n \in \omega$, let $r_{n,0} = 0$.

*stage* $s+1$. For each $n < s+1$, say that $s+1$ is an *$n$-recovery stage* if all of the following conditions hold.

1. $\mathcal{G}_n[s]$ has a special component isomorphic to some component of $\mathcal{A}^0_s$.

2. $(\mathcal{G}_n[s])_n \cong (\mathcal{A}^0_s)_n$.

3. Let $j \notin A[s]$ be less than or equal to the number of $n$-recovery stages before stage $s + 1$. There is a component of $\mathcal{G}_n[s]$ isomorphic to $[6j]$ and for each $l \in \{1, 2, 4, 5\}$ there is a component of $\mathcal{G}_n[s]$ isomorphic to $[6\langle n, j\rangle + l]$.

If $s+1$ is an $n$-recovery stage then, for $i = 0, 1$, let $S^i_{n,s}$ be the component of $\mathcal{A}^i_s$ that is isomorphic to the special component of $\mathcal{G}_n[s]$. If $s + 1$ is the first $n$-recovery stage then let $r_{n,s+1} = 0$. Otherwise, proceed as follows. Let $i = r_{n,s}$ and let $t + 1$ be the last $n$-recovery stage before stage $s+1$. If $S^i_{n,s}$ extends $S^i_{n,t}$ then let $r_{n,s+1} = i$, and otherwise let $r_{n,s+1} = 1 - i$.

If $s + 1$ is not an $n$-recovery stage then let $r_{n,s+1} = r_{n,s}$.

Now let $n_0, n_1, \ldots, n_m$ be all the numbers $n_j$ such that $a_s$ is less than the number of $n_j$-recovery stages less than or equal to $s + 1$. We say that each $n_j$, $j \leqslant m$, is *active* at stage $s + 1$. For $i = 0, 1$ and $j \leqslant m$, let $X^i_s$, $Y^i_{n_j,s}$, and $Z^i_{n_j,s}$ be the components of $\mathcal{A}^i_s$ isomorphic to $[6a_s]$, $[6\langle n_j, a_s\rangle + 1]$, and $[6\langle n_j, a_s\rangle + 2]$, respectively.

For each $j \leqslant m$, let $t_j + 1 \leqslant s + 1$ be the last $n_j$-recovery stage. We say that $s + 1$ is an *$n_j$-first stage* if it is the first stage after stage $t_j$ at which $n_j$ is active.

We say that $s+1$ is an *$n_j$-change stage* if it is an $n_j$-first stage and either $t_j+1$ was the first $n_j$-recovery stage or $r_{n_j,t_j+1} \neq r_{n_j,t_j}$.

We say that $s + 1$ is an *$n_j$-isomorphism recovery stage* if it is an $n_j$-first stage but not an $n_j$-change stage and one of the following conditions holds.

1. The last $n_j$-first stage before stage $s + 1$ was an $n_j$-change stage.

2. There has been at least one stage at which $n_j$ was active after the last $n_j$-isomorphism recovery stage and before stage $s + 1$.

For each $j \leqslant m$ we define components $B^i_{n_j,s}$ and $C^i_{n_j,s}$, $i = 0, 1$. There are two cases.

1. $s+1$ is an $n_j$-isomorphism recovery stage. If the first condition in the definition of $n_j$-isomorphism recovery stage holds then let $t+1$ be the last $n_j$-first stage, and otherwise let $t+1$ be the first stage after the last $n_j$-isomorphism recovery stage at which $n_j$ was active. There are two subcases.

   (a) If $r_{n_j,s+1} = 0$ then let $C^0_{n_j,s}$ be the component of $\mathcal{A}^0_s$ that extends $B^0_{n_j,t}$ and let $C^1_{n_j,s}$ be its isomorphic image in $\mathcal{A}^1_s$. For $i = 0, 1$, let $B^i_{n_j,s}$ be the component of $\mathcal{A}^i_s$ isomorphic to $[6\langle n_j, a_s\rangle + 4]$.

   (b) If $r_{n_j,s+1} = 1$ then let $B^1_{n_j,s}$ be the component of $\mathcal{A}^1_s$ that extends $C^1_{n_j,t}$ and let $B^0_{n_j,s}$ be its isomorphic image in $\mathcal{A}^0_s$. For $i = 0, 1$, let $C^i_{n_j,s}$ be the component of $\mathcal{A}^i_s$ isomorphic to $[6\langle n_j, a_s\rangle + 5]$.

2. $s+1$ is not an $n_j$-isomorphism recovery stage. For $i = 0, 1$, let $B^i_{n_j,s}$ be the component of $\mathcal{A}^i_s$ isomorphic to $[6\langle n_j, a_s\rangle + 4]$ and let $C^i_{n_j,s}$ be the component of $\mathcal{A}^i_s$ isomorphic to $[6\langle n_j, a_s\rangle + 5]$.

For each $j \leqslant m$, proceed as follows. Let $i = r_{n_j,s+1}$ and let $t+1 \leqslant s+1$ be the last $n_j$-recovery stage. Let $R^i_{n_j,s}$ be the component of $\mathcal{A}^i_s$ that extends $S^i_{n_j,t}$ and let $R^{1-i}_{n_j,s}$ be its isomorphic image in $\mathcal{A}^{1-i}_s$.

Now perform

$$\mathbf{L}(Y^0_{n_0,s}, \ldots, Y^0_{n_m,s}; X^0_s; Z^0_{n_0,s}, \ldots, Z^0_{n_m,s}; B^0_{n_0,s}, R^0_{n_0,s}, C^0_{n_0,s}; \\ B^0_{n_1,s}, R^0_{n_1,s}, C^0_{n_1,s}; \ldots; B^0_{n_m,s}, R^0_{n_m,s}, C^0_{n_m,s})$$

on $\mathcal{A}^0_s$ to get $\mathcal{A}^0_{s+1}$ and perform

$$\mathbf{R}(Y^1_{n_0,s}, \ldots, Y^1_{n_m,s}; X^1_s; Z^1_{n_0,s}, \ldots, Z^1_{n_m,s}; B^1_{n_0,s}, R^1_{n_0,s}, C^1_{n_0,s}; \\ B^1_{n_1,s}, R^1_{n_1,s}, C^1_{n_1,s}; \ldots; B^1_{n_m,s}, R^1_{n_m,s}, C^1_{n_m,s})$$

on $\mathcal{A}^1_s$ to get $\mathcal{A}^1_{s+1}$. (If no $n$ is active at stage $s+1$ then, for $j = 0, 1$, let $Y^j_s$ and $Z^j_s$ be the components of $\mathcal{A}^j_s$ isomorphic to $[6\langle 0, a_s\rangle + 1]$ and $[6\langle 0, a_s\rangle + 2]$, respectively. Perform $\mathbf{L}(Y^0_s, X^0_s, Z^0_s)$ on $\mathcal{A}^0_s$ to get $\mathcal{A}^0_{s+1}$ and perform $\mathbf{R}(Y^1_s, X^1_s, Z^1_s)$ on $\mathcal{A}^1_s$ to get $\mathcal{A}^1_{s+1}$.)

Put the coding location of the copy of $[6a_s]$ in $\mathcal{A}^0_0$ into $U^0$ and put the coding location of the copy of $[6a_s]$ in $\mathcal{A}^1_{s+1} - \mathcal{A}^1_s$ into $U^1$.

This completes the construction. Let $\mathcal{A}^0 = \bigcup_{s\in\omega} \mathcal{A}^0_s$ and $\mathcal{A}^1 = \bigcup_{s\in\omega} \mathcal{A}^1_s$. Since for each $s \in \omega$ and $i = 0, 1$, all numbers in $\mathcal{A}^i_{s+1} - \mathcal{A}^i_s$ are greater than $s$, $\mathcal{A}^0$ and $\mathcal{A}^1$ are computable. We now wish to argue that properties (2.1)–(2.3) are satisfied. Theorem 1.7 will then follow immediately.

Property (2.2) is easy to establish, so we deal with it first.

**2.7 Lemma.** *$U^0 \equiv_m A$ and $U^1$ is computable.*

*Proof.* The numbers in $U^0$ are all coding locations of components of $\mathcal{A}_0^0$ of the form $[6j]$, $j \in \omega$, and the coding location of the copy of $[6j]$ in $\mathcal{A}_0^0$ is in $U^0$ if and only if $j \in A$. Since given any number we can computably determine whether it is a coding location in $\mathcal{A}_0^0$ and if so, for what $[k]$, this means that $U^0 \equiv_{\mathrm{m}} A$.

Any number put into $U^1$ at a stage $s+1$ is a new number, that is, one not in the domain of $\mathcal{A}_s^1$, and hence is greater than $s$. Thus $U^1$ is computable. □

In showing that (2.1) and (2.3) are satisfied, we will need a few facts about the construction. The more obvious ones are given without proof, while the remaining ones are broken down into easily checked properties of the construction. Figures 2.3 and 2.4 should be helpful here.

We say that a component of $\mathcal{A}^i$ participates in an operation at stage $s+1$ if it extends a component of $\mathcal{A}_s^i$ that participates in an operation at stage $s+1$.

**2.8 Lemma.** *Let $\mathcal{G} \cong A^0$ be computable. Given $x$ in the domain of $\mathcal{G}$, we can computably determine if $x$ is the coding location of a copy of some $[k]$, $k \in \omega$, and if so, for what $k$. In particular, the set of coding locations of copies of $[6j]$, $j \in \omega$, in $\mathcal{G}$ is computable.*

**2.9 Lemma.** *Let $K$ and $L$ be distinct components of $\mathcal{A}_s^i$ such that $K$ is not a copy of $[6k+1]$ or $[6k+2]$ for any $k \in \omega$. $K$ and $L$ are not extended by the same component of $\mathcal{A}^i$.*

Lemma 2.9 will be used without explicit mention several times below.

**2.10 Lemma.** *A component of $\mathcal{A}^i$ is infinite if and only if it participates in operations infinitely often.*

**2.11 Lemma.** *Let $k, n \in \omega$. Any component of $\mathcal{A}^i$ containing a copy of $[6k]$, $[6\langle n,k\rangle+1]$, or $[6\langle n,k\rangle+2]$ can participate in an operation at most once. Any component of $\mathcal{A}^i$ containing a copy of $[6n+3]$, $[6\langle n,k\rangle+4]$, or $[6\langle n,k\rangle+5]$ can participate in operations only at stages at which $n$ is active.*

**2.12 Lemma.** *Suppose that $r_{n,s} = i \neq r_{n,s+1}$. Of all the components of $(\mathcal{A}^i)_n$ that participate in operations before stage $s+1$, the only one that can participate in an operation after stage $s$ is the one that extends $S_{n,s}^i$.*

*Proof.* Suppose that a component of $(\mathcal{A}^i)_n$ participates in operations at stages $t < u$ and does not participate in an operation at any stage in $(t,u)$, and let $v$ be the last $n$-change stage before stage $u$. It is not hard to check that it must then be the case that $t \geqslant v$.

Now let $t$ be the first stage after stage $s$ at which $n$ is active. Then $t$ is an $n$-change stage, and hence not an $n$-isomorphism recovery stage. It follows that, of all the components of $(\mathcal{A}^i)_n$ that participate in operations before stage $s+1$, the only one that participates in an operation at stage $t$ is the one that extends $S_{n,s}^i$. The lemma now follows by induction, using the fact mentioned in the previous paragraph. □

**2.13 Lemma.** *For each $s \in \omega$, $\mathcal{A}_s^0 \cong \mathcal{A}_s^1$ and no component of $\mathcal{A}_s^i$ is embeddable in another component of $\mathcal{A}_s^i$. Furthermore, if a component of $\mathcal{A}_s^i$ participates in an operation at stage $s+1$ then so does the (unique) isomorphic component of $\mathcal{A}_s^{1-i}$.*

**2.14 Lemma.** *Suppose that $r_{n,s} = i$ for all $s > t$ and $n$ is active at stages $s_0 + 1$ and $s_1 + 1$, where $s_1 > s_0 \geqslant t$. Then $R^i_{n,s_1}$ extends $R^i_{n,s_0}$.*

**2.15 Lemma.** *Let $s+1$ be an $n$-recovery stage that is not the first such stage. Let $t+1$ be the last $n$-recovery stage before stage $s+1$. If $r_{n,s} = 0 \neq r_{n,s+1}$ then $S^0_{n,s}$ extends $B^0_{n,u}$ for some $u \in [t,s)$. Similarly, if $r_{n,s} = 1 \neq r_{n,s+1}$ then $S^1_{n,s}$ extends $C^1_{n,u}$ for some $u \in [t,s)$.*

*Proof.* The two cases, $i = 0$ and $i = 1$, are similar. We do the case $i = 0$.

Since $S^0_{n,s}$ contains a copy of $S^0_{n,t}$ and $r_{n,t+1} = r_{n,s} = 0$, either $S^0_{n,s}$ extends $S^0_{n,t}$ or $S^0_{n,s}$ extends $B^0_{n,u}$ for some $u$ such that $t \leqslant u < s$. But it cannot be the case that $S^0_{n,s}$ extends $S^0_{n,t}$, since that would imply that $r_{n,s+1} = 0$. $\square$

**2.16 Lemma.** *Suppose that $r_{n,t} = 0$ (resp. $r_{n,t} = 1$) for all $t \geqslant s_0$. Then no component of $(\mathcal{A}^0)_n$ $((\mathcal{A}^1)_n)$ can participate in an operation more than twice after stage $s_0$ unless it extends $R^0_{n,t}$ $(R^1_{n,t})$ for some $t \geqslant s_0$, while no component of $(\mathcal{A}^1)_n$ $((\mathcal{A}^0)_n)$ can participate in an operation more than twice after stage $s_0$ unless it extends $C^1_{n,t}$ $(B^0_{n,t})$ for some $t \geqslant s_0$ such that $t+1$ is an $n$-isomorphism recovery stage.*

*Proof.* The two cases, $i = 0$ and $i = 1$, are similar. We do the case $i = 0$.

Suppose that component $K$ of $(\mathcal{A}^0)_n$ participates in operations at stages $s+1 < t+1 < u+1$, where $s+1 \geqslant s_0$, but not at any stage in $(t+1, u+1)$. Then either $K$ extends $R^0_{n,u}$ or $u+1$ is an $n$-isomorphism recovery stage and $K$ extends $C^0_{n,u}$. We claim that the latter case cannot hold. Indeed, if $K$ extends $C^0_{n,u}$ then $K$ extends $B^0_{n,v}$ for some $v \in \omega$. Since $K$ does not participate in operations at any stage in $(t+1, u+1)$, $v = t$. But since $r_{n,t+1} = 0$, $B^0_{n,t}$ is a singleton component. Thus $K$ does not participate in an operation at stage $s+1$, contrary to hypothesis.

Now suppose that component $L$ of $(\mathcal{A}^1)_n$ participates in operations at stages $s+1 < t+1 < u+1$, where $s+1 \geqslant s_0$, but not at any stage in $(t+1, u+1)$. Then either $L$ extends $R^1_{n,t}$ or $t+1$ is an $n$-isomorphism recovery stage and $L$ extends $C^1_{n,t}$. But in the former case, $u+1$ is an $n$-isomorphism recovery stage and, since $K$ does not participate in operations at any stage in $(t+1, u+1)$, $L$ extends $C^1_{n,u}$. $\square$

**2.17 Lemma.** *Suppose that $s < t < v$ are such that $s+1$ is an $n$-isomorphism recovery stage, $r_{n,u} = r_{n,s+1}$ for all $u > s$, $t+1$ is the next stage after stage $s+1$ at which $n$ is active, and $v+1$ is the next $n$-isomorphism recovery stage after stage $s+1$. For $i = 0, 1$, let $B^i$, $R^i$, and $C^i$ be the components of $\mathcal{A}^i_{t+1}$ that extend $B^i_{n,t}$, $R^i_{n,t}$, and $C^i_{n,t}$, respectively, and let $\widehat{B}^i$, $\widehat{R}^i$, and $\widehat{C}^i$ be the components of $\mathcal{A}^i_v$ that extend $B^i$, $R^i$, and $C^i$, respectively. If $r_{n,s+1} = 0$ then $\widehat{B}^0 \cong B^0$ and $\widehat{R}^1 \cong R^1$, while if $r_{n,s+1} = 1$ then $\widehat{C}^1 \cong C^1$ and $\widehat{R}^0 \cong R^0$.*

*Proof.* The two cases, $i = 0$ and $i = 1$, are similar. We do the case $i = 0$. It is enough to show that the components of $(\mathcal{A}^0)_n$ and $(\mathcal{A}^1)_n$ that extend $B^0$ and $R^1$, respectively, do not participate in operations at any stage in $(t + 1, v + 1)$.

Suppose that component $K$ of $(\mathcal{A}^0)_n$ participates in operations at stages $t + 1$ and $u + 1$, where $t < u < v$. Since no stage in $(t + 1, v + 1)$ is an $n$-isomorphism recovery stage, $K$ extends $R^0_{n,u}$, which in turn extends $R^0_{n,t}$. Thus $K$ does not extend $B^0$.

Now suppose that component $L$ of $(\mathcal{A}^1)_n$ participates in operations at stages $t + 1$ and $u + 1$, where $t < u < v$. Again, no stage in $(t + 1, v + 1)$ is an $n$-isomorphism recovery stage, so $L$ extends $R^1_{n,u}$, which in turn extends $C^1_{n,t}$. Thus $L$ does not extend $R^1$. □

**2.18 Lemma.** *Let $x$ be the coding location of a copy of $[6a_s]$ in component $K$ of $\mathcal{A}^i$. Either $K$ contains a copy of $[6\langle n, a_s\rangle + 1]$ for some $n \in \omega$, in which case $x \notin U^i$, or $K$ contains a copy of $[6\langle n, a_s\rangle + 2]$ for some $n \in \omega$, in which case $x \in U^i$.*

We now wish to show that (2.1) holds. It follows from Lemmas 2.10, 2.13, and 2.18 that it is enough to show that for each infinite component of $\mathcal{A}^i$ there is a corresponding isomorphic component of $\mathcal{A}^{1-i}$. The first step in establishing this result is characterizing the infinite components of $\mathcal{A}^i$.

**2.19 Lemma.** *If $r_{n,s}$ does not have a limit then no component of $(\mathcal{A}^i)_n$ is infinite.*

*Proof.* Suppose that $r_{n,s} = 0 \neq r_{n,s+1}$ and let $t + 1$ be the last $n$-recovery stage before stage $s + 1$. By Lemma 2.12, of all the components of $(\mathcal{A}^0)_n$ that have participated in operations before stage $s + 1$, the only one that can participate in an operation after stage $s$ is the component $L$ that extends $S^0_{n,s}$. By Lemma 2.15, $L$ extends $B^0_{n,u}$ for some $u \in [t, s)$. But the fact that $r_{n,t+1} = 0$ means that for all $u \in [t, s)$, $B^0_{n,u}$ is a singleton component, and hence did not participate in an operation at any stage before stage $t+1$.

Thus no component of $(\mathcal{A}^0)_n$ that participates in an operation before stage $t + 1$ can do so again after stage $s$. A similar argument shows that if $r_{n,s} = 1 \neq r_{n,s+1}$ and $t + 1$ is the last $n$-recovery stage before stage $s + 1$ then no component of $(\mathcal{A}^1)_n$ that participates in an operation before stage $t+1$ can do so again after stage $s$. The lemma now follows from Lemma 2.10. □

Thus the only components of $\mathcal{A}^i$ that can be infinite are those components that are in $(\mathcal{A}^i)_n$ for some $n$ such that $r_{n,s}$ has a limit and $n$ is active infinitely often. So, by the comments preceding Lemma 2.19, to establish that (2.1) holds, it is enough to show that if $r_{n,s}$ has a limit and $n$ is active infinitely often then, for each $i = 0, 1$, there is exactly one infinite component $S^i_n$ of $(\mathcal{A}^i)_n$ and $S^0_n \cong S^1_n$. This is what we do in the next few lemmas.

**2.20 Lemma.** *There are infinitely many $n$-recovery stages if and only if $n$ is active infinitely often.*

*Proof.* By definition, $n$ is active at a stage $s+1$ if and only if $a_s$ is less than the number of $n$-recovery stages less than or equal to $s+1$. Thus, if there are finitely many $n$-recovery stages then $n$ cannot be active infinitely often.

For the other direction, suppose that there are infinitely many $n$-recovery stages but only finitely many stages at which $n$ is active. Let $s$ be the last stage at which $n$ is active. Now given $x \in \omega$, let $t+1$ be the first stage after stage $s$ by which there have been $x+1$ many $n$-recovery stages. Then $x \in A \Leftrightarrow x \in A[t]$, since if $x$ were equal to $a_u$ for some $u \geqslant t$ then $n$ would be active at stage $u+1$. But this means that $A$ is computable, contrary to hypothesis. □

**2.21 Lemma.** *If $n$ is active infinitely often and $r_{n,s}$ has a limit then there are infinitely many $n$-isomorphism recovery stages.*

*Proof.* If $n$ is active infinitely often then, by Lemma 2.20, there are infinitely many $n$-recovery stages, and thus infinitely many $n$-first stages. The fact that $r_{n,s}$ has a limit implies that only finitely many of these can be $n$-change stages. The lemma now follows directly from the definition of $n$-isomorphism recovery stage. □

**2.22 Lemma.** *Suppose that $n$ is active infinitely often and $s$ and $i$ are such that $r_{n,t} = r_{n,s} = i$ for all $t \geqslant s$. By Lemma 2.21, there are infinitely many $n$-isomorphism recovery stages. Let $s_0 + 1 < s_1 + 1 < \cdots$ be the $n$-isomorphism recovery stages after stage $s$. For each $j \in \omega$, let $t_j + 1$ be the next stage after stage $s_j + 1$ at which $n$ is active. (Note that $t_j < s_{j+1}$ for all $j \in \omega$.) For $t \geqslant t_0$, let $K^l_t$ be the component of $\mathcal{A}^l_t$ that extends $R^l_{n,t_0}$. Then $K^l_{t_j} = R^l_{n,t_j}$ for all $j \in \omega$.*

*Proof.* The two cases, $i = 0$ and $i = 1$, are similar. We do the case $i = 0$.

That $K^0_{t_j} = R^0_{n,t_j}$ for all $j \in \omega$ follows from Lemma 2.14.

Now assume by induction that $K^1_{t_j} = R^1_{n,t_j}$. Let $B$ be the component of $\mathcal{A}^0_{t_j+1}$ that extends $B^0_{n,t_j}$. By construction, $B \cong K^1_{t_j+1}$. Since $s_{j+1}+1$ is an $n$-isomorphism recovery stage, $C^0_{n,s_{j+1}}$ extends $B$. Thus, by Lemma 2.17, $C^0_{n,s_{j+1}} \cong B$. By the same lemma, $K^1_{s_{j+1}} \cong K^1_{t_j+1}$, so $C^0_{n,s_{j+1}} \cong K^1_{s_{j+1}}$, and thus $C^1_{n,s_{j+1}} = K^1_{s_{j+1}}$. Let $R$ be the component of $\mathcal{A}^0_{s_{j+1}+1}$ that extends $R^0_{n,s_{j+1}}$. Then $R \cong K^1_{s_{j+1}+1}$. But, by Lemma 2.11, $R^0_{n,t_{j+1}} \cong R$ and $K^1_{t_{j+1}} \cong K^1_{s_{j+1}+1}$, so $K^1_{t_{j+1}} \cong R^0_{n,t_{j+1}}$, and hence $K^1_{t_{j+1}} = R^1_{n,t_{j+1}}$. □

For the next two lemmas, we assume the hypotheses of Lemma 2.22 and adopt its notation. Let $S^l_n$ be the component of $\mathcal{A}^l$ that extends $R^l_{n,s_0}$.

**2.23 Lemma.** *$S^l_n$ is the only infinite component of $(\mathcal{A}^l)_n$.*

*Proof.* This follows immediately from Lemmas 2.10, 2.16, and 2.22 and the observation that, for all $j \in \omega$, if $i = 0$ in the hypotheses of Lemma 2.22 then $R^1_{n,t_j}$ extends $C^1_{n,s_j}$, while if $i = 1$ then $R^0_{n,t_j}$ extends $B^0_{n,s_j}$. □

**2.24 Lemma.** $S^0_n \cong S^1_n$.

*Proof.* This follows immediately from Lemma 2.22, since, by definition, $R^0_{n,t_j} \cong R^1_{n,t_j}$ for all $j \in \omega$, and $S^i_n = \bigcup_{j \in \omega} R^i_{n,t_j}$ for $i = 0, 1$. □

As we have argued above, Lemmas 2.23 and 2.24 suffice to establish that (2.1) holds.

**2.25 Lemma.** *$\mathcal{A}^0 \cong \mathcal{A}^1$ via an isomorphism that carries $U^0$ to $U^1$.*

We are left with showing that (2.3) holds. This will break down into three steps. Suppose that $\mathcal{G}_n \cong \mathcal{A}^0$ and let $U$ be the image of $U^0$ in $\mathcal{G}_n$.

1. We show that $r_{n,s}$ reaches a limit $r_n$.
2. Let $t$ be such that for all $u \geqslant t$, $r_{n,u} = r_n$. Let $A'$ be the set of all $a_s$ such that either $s < t$ or the number of $n$-recovery stages less than or equal to $s+1$ is less than or equal to $a_s$. Let $N$ be the set of all $x \in \mathcal{G}_n$ such that $x$ is the coding location of a copy of $[6a]$, $a \in A'$. We show that $A'$, $N$, and $U \cap N$ are computable.
3. Let $C$ be the set of coding locations of copies of graphs of the form $[6j]$, $j \in \omega$, in $\mathcal{G}_n$ and let $M = C - N$. Note that $M$ is computable. We show that
   (a) if $r_n = 0$ then an element $x$ of $M$ is in $U$ if and only if, for some $j \in A$, $x$ is the coding location of the first copy of $[6j]$ to appear in $\mathcal{G}_n$, so that $U \cap M \equiv_{\mathrm{m}} A$, while
   (b) if $r_n = 1$ then an element $x$ of $M$ is in $U$ if and only if, for some $j \in \omega$, $x$ is the coding location of the second copy of $[6j]$ to appear in $\mathcal{G}_n$, so that $U \cap M$ is computable.

Since $U = (U \cap N) \cup (U \cap M)$, this is enough to establish that (2.3) holds.

**2.26 Lemma.** *If $\mathcal{G}_n \cong \mathcal{A}^0$ then there are infinitely many $n$-recovery stages, and hence the special component of $\mathcal{G}_n$ is infinite.*

*Proof.* If $\mathcal{G}_n \cong A^0$ then $\mathcal{G}_n$ has a special component. Now suppose that there are only $m$ many $n$-recovery stages. Let $s_0$ be the last $n$-recovery stage. (If there are no $n$-recovery stages then let $s_0$ be the first stage at which $\mathcal{G}_n$ has a special component.) By Lemma 2.20, there is a stage $s_1 > s_0$ such that $n$ is not active at any stage $t \geqslant s_1$. If $m = a_u$ for some $u > s_1$ then let $s = u + 1$; otherwise let $s = s_1$.

Consider the components of $\mathcal{A}^0$ that contain a copy of the special component of $\mathcal{G}_n$. By Lemma 2.11, each such component is finite. Thus, if the first condition in the definition of $n$-recovery stage is not eventually satisfied after stage $s$ then the special component of $\mathcal{G}_n$ is not isomorphic to any component of $\mathcal{A}^0$.

Now consider $(\mathcal{A}^0)_n$. Again by Lemma 2.11, $(\mathcal{A}^0)_n$ is finite. So if the second condition in the definition of $n$-recovery stage is not eventually satisfied after stage $s$ then $(\mathcal{G}_n)_n \not\cong (\mathcal{A}^0)_n$.

Finally, let $j \notin A[s]$, $j \leqslant m$, and $l \in \{1, 2, 4, 5\}$ and consider the components of $\mathcal{A}^0$ that contain a copy of $[6\langle n, j\rangle + l]$. By the choice of $s$, $j \notin A[s] \Rightarrow j \notin A$, so there is only one such component and it is isomorphic to $[6\langle n, j\rangle + l]$. Similarly, there is only one component that contains a copy of $[6j]$ and it is isomorphic to $[6j]$.

Thus, if the third condition in the definition of $n$-recovery stage is not eventually satisfied after stage $s$ then there is a component of $\mathcal{A}^0$ that is not isomorphic to any component of $\mathcal{G}_n$.

In any case, $\mathcal{G}_n$ cannot be isomorphic to $\mathcal{A}^0$, contradicting the hypothesis of the lemma. So there are infinitely many $n$-recovery stages.

Now, given any two $n$-recovery stages $t + 1 < u + 1$ such that there is a stage in $(t, u]$ at which $n$ is active, the special component of $\mathcal{G}_n[u]$ properly extends the special component of $\mathcal{G}_n[t]$. But, by Lemma 2.20, $n$ is active at infinitely many stages. This establishes the second part of the lemma. □

**2.27 Lemma.** *If $\mathcal{G}_n \cong \mathcal{A}^0$ then $r_n = \lim_s r_{n,s}$ is well-defined.*

*Proof.* This follows immediately from Lemmas 2.19 and 2.26. □

**2.28 Lemma.** *Suppose that $\mathcal{G}_n \cong \mathcal{A}^0$. Let $U$ be the image of $U^0$ under this isomorphism. By Lemma 2.27, $r_n = \lim_s r_{n,s}$ is well-defined. Let $t$ be such that for all $u \geqslant t$, $r_{n,u} = r_n$. Let $A'$ be the set of all $a_s$ such that either $s < t$ or the number of $n$-recovery stages less than or equal to $s + 1$ is less than or equal to $a_s$. Let $N$ be the set of all $x \in \mathcal{G}_n$ such that $x$ is the coding location of a copy of $[6a]$, $a \in A'$. Then $A'$, $N$, and $U \cap N$ are computable.*

*Proof.* By Lemma 2.8, given $x$ in the domain of $\mathcal{G}_n$, we can computably determine if $x$ is the coding location of a copy of some $[k]$, $k \in \omega$, and if so, for what $k$.

By Lemma 2.26, there are infinitely many $n$-recovery stages, so the set of all $a_s$ such that the number of $n$-recovery stages less than or equal to $s + 1$ is less than or equal to $a_s$ is computable. Thus $A'$ and $N$ are computable.

Now, if $x \in N$ then $x$ is the coding location of a copy of $[6a_s]$ for some $s \in \omega$. Let $K$ be the component of $\mathcal{G}_n$ that contains $x$. By Lemma 2.18, $K$ contains either a copy of $[6\langle m, a_s\rangle + 1]$ for some $m \in \omega$ or a copy of $[6\langle m, a_s\rangle + 2]$ for some $m \in \omega$, but not both, and $x \in U \cap N$ if and only if $K$ contains a copy of $[6\langle m, a_s\rangle + 2]$ for some $m \in \omega$. Thus $U \cap N$ is computable. □

**2.29 Lemma.** *Suppose that $s + 1$ is an $n$-recovery stage, but not the first such stage, and that $r_{n,s+1} = r_{n,s} = i$. Let $t + 1$ be the last $n$-recovery stage before stage $s + 1$ and let $s_0 + 1 < s_1 + 1 < \cdots < s_m + 1$ be the stages in the interval $(t, s]$ at which $n$ is active. For each $k \leqslant m$, let $Y_k$, $X_k$, $Z_k$, $B_k$, $R_k$ and $C_k$ be $Y^i_{n,s_k}$, $X^i_{s_k}$, $Z^i_{n,s_k}$, $B^i_{n,s_k}$, $R^i_{n,s_k}$, and $C^i_{n,s_k}$, respectively, and let $Y'_k$, $X'_k$, $Z'_k$, $B'_k$, $R'_k$ and $C'_k$ be the components of $\mathcal{A}^i_s$ that extend $Y_k$, $X_k$, $Z_k$, $B_k$, $R_k$ and $C_k$, respectively. Then the following hold.*

1. *For every $k \leqslant m$, $Y_k$, $X_k$, $Z_k$, $B_k$, and $C_k$ are components of $\mathcal{A}^i_t$, and so is $R_0$. For every $k, l \leqslant m$, $R'_k = R'_l$.*

2. *There exists a component $\widehat{R}_0$ of $\mathcal{G}_n[t]$ such that $\widehat{R}_0 \cong R_0$ and, for each $k \leqslant m$, there exist components $\widehat{Y}_k$, $\widehat{X}_k$, $\widehat{Z}_k$, $\widehat{B}_k$, and $\widehat{C}_k$ of $\mathcal{G}_n[t]$ such that $\widehat{Y}_k \cong Y_k$, $\widehat{X}_k \cong X_k$, $\widehat{Z}_k \cong Z_k$, $\widehat{B}_k \cong B_k$, and $\widehat{C}_k \cong C_k$.*

3. *Let $\widehat{R}'_0$ be the component of $\mathcal{G}_n[s]$ that extends $\widehat{R}_0$ and, for each $k \leqslant m$, let $\widehat{Y}'_k$, $\widehat{X}'_k$, $\widehat{Z}'_k$, $\widehat{B}'_k$, and $\widehat{C}'_k$ be the components of $\mathcal{G}_n[s]$ that extend $\widehat{Y}_k$, $\widehat{X}_k$, $\widehat{Z}_k$, $\widehat{B}_k$, and $\widehat{C}_k$, respectively. $\widehat{R}'_0 \cong R'_0$ and, for each $k \leqslant m$, $\widehat{Y}'_k \cong Y'_k$, $\widehat{X}'_k \cong X'_k$, $\widehat{Z}'_k \cong Z'_k$, $\widehat{B}'_k \cong B'_k$, and $\widehat{C}'_k \cong C'_k$.*

*Proof.* The first part of the lemma follows from the way $Y^i_{n,s_k}$, $X^i_{s_k}$, $Z^i_{n,s_k}$, $B^i_{n,s_k}$, $R^i_{n,s_k}$, and $C^i_{n,s_k}$ are defined and Lemma 2.14. The second part of the lemma follows from the definition of $n$-recovery stage. We prove the third part of the lemma.

The two cases, $i = 0$ and $i = 1$, are similar. We do the case $i = 0$. Figure 2.3 might be helpful here.

By definition, $\widehat{R}_0$ and $\widehat{R}'_0$ are the special components of $\mathcal{G}_n[t]$ and $\mathcal{G}_n[s]$, respectively. Thus, since $r_{n,s+1} = r_{n,s} = 0$ and $s+1$ is an $n$-recovery stage, $\widehat{R}'_0 \cong R'_0$. We now proceed by reverse induction, beginning with $m$.

It follows from the construction and the first part of the lemma that if $K$ is taken from among $\widehat{R}'_0$, $\widehat{Y}'_k$, $\widehat{X}'_k$, $\widehat{Z}'_k$, $\widehat{B}'_k$, and $\widehat{C}'_k$, $k \leqslant m$, and $L \neq K$ is taken from among $\widehat{R}'_0$, $\widehat{Y}'_l$, $\widehat{X}'_l$, $\widehat{Z}'_l$, $\widehat{B}'_l$, and $\widehat{C}'_l$, $l \leqslant m$, then $K \ncong L$. Furthermore, if $K$ is one of $\widehat{Y}'_k$, $\widehat{X}'_k$, $\widehat{Z}'_k$, $\widehat{B}'_k$, or $\widehat{C}'_k$, and $L$ is a component of $\mathcal{A}^0_s$ such that $K \cong L$ then $L$ is one of $R'_0$, $Y'_l$, $X'_l$, $Z'_l$, $B'_l$, or $C'_l$, $l \geqslant k$.

Thus, since we assume by induction that for all $j > k$, $\widehat{Y}'_j \cong Y'_j$, $\widehat{X}'_j \cong X'_j$, $\widehat{Z}'_j \cong Z'_j$, $\widehat{B}'_j \cong B'_j$, and $\widehat{C}'_j \cong C'_j$, we may assume that if $K$ is one of $\widehat{Y}'_k$, $\widehat{X}'_k$, $\widehat{Z}'_k$, $\widehat{B}'_k$, or $\widehat{C}'_k$ and $L$ is a component of $\mathcal{A}^0_s$ such that $K \cong L$ then $L$ is one of $R'_0$, $Y'_k$, $X'_k$, $Z'_k$, $B'_k$, or $C'_k$.

The only components among $R'_0$, $Y'_k$, $X'_k$, $Z'_k$, $B'_k$, or $C'_k$ that contain copies of $\widehat{C}_k$ are $R'_0$ and $C'_k$. Since $\widehat{R}'_0 \cong R'_0$, it must be the case that $\widehat{C}'_k \cong C'_k$.

The only components among $R'_0$, $Y'_k$, $X'_k$, $Z'_k$, $B'_k$, or $C'_k$ that contain copies of $\widehat{Y}_k$ are $C'_k$ and $Y'_k$. Since $\widehat{C}'_k \cong C'_k$, it must be the case that $\widehat{Y}'_k \cong Y'_k$.

The only components among $R'_0$, $Y'_k$, $X'_k$, $Z'_k$, $B'_k$, or $C'_k$ that contain copies of $\widehat{X}_k$ are $Y'_k$ and $X'_k$. Since $\widehat{Y}'_k \cong Y'_k$, it must be the case that $\widehat{X}'_k \cong X'_k$.

The only components among $R'_0$, $Y'_k$, $X'_k$, $Z'_k$, $B'_k$, or $C'_k$ that contain copies of $\widehat{Z}_k$ are $X'_k$ and $Z'_k$. Since $\widehat{X}'_k \cong X'_k$, it must be the case that $\widehat{Z}'_k \cong Z'_k$.

The only components among $R'_0$, $Y'_k$, $X'_k$, $Z'_k$, $B'_k$, or $C'_k$ that contain copies of $\widehat{B}_k$ are $Z'_k$ and $B'_k$. Since $\widehat{Z}'_k \cong Z'_k$, it must be the case that $\widehat{B}'_k \cong B'_k$. □

**2.30 Lemma.** *Suppose that $s+1$ is an $n$-recovery stage such that $r_{n,s+1} = r_{n,s}$. Let $t+1$ be the last $n$-recovery stage before stage $s+1$ and let $j \in A[s] - A[t]$ be less than the* *number of $n$-recovery stages less than or equal to $t+1$. By the definition of $n$-recovery stage, there is a unique component $K$ of $\mathcal{G}_n[t]$ isomorphic to $[6j]$. Let $L$ be the component of $\mathcal{G}_n$ that extends $K$. Then $L$ contains a copy of $[6\langle n,j\rangle+2]$ if and only if $r_{n,s+1}=0$.*

*Proof.* Let $i = r_{n,s+1}$. Let $u$ be such that $j = a_u$. Since $t+1 \leqslant u < s$ and $j$ is less than the number of $n$-recovery stages less than or equal to $t+1$, $n$ is active at stage $u+1$. So, adopting the notation of Lemma 2.29, $K = \widehat{X}_k$ for some $k$. By Lemma 2.11, $L \cong \widehat{X}'_k$. Thus, by Lemma 2.29, $L \cong X'_k$. But $X'_k$ is the component of $\mathcal{A}^i_s$ that extends $X^i_u$, so, by construction, $X'_k$ contains a copy of $[6\langle n,j\rangle+2]$ if and only if $i=0$. □

**2.31 Lemma.** *Suppose that $\mathcal{G}_n \cong \mathcal{A}^0$. Let $U$ be the image of $U^0$ under this isomorphism. Then either $U$ is computable or $U \equiv_m A$.*

*Proof.* Let $N$ be as in Lemma 2.28. Let $C$ be the set of coding locations of copies of graphs of the form $[6j]$, $j \in \omega$, in $\mathcal{G}_n$ and let $M = C - N$. By Lemmas 2.8 and 2.28, $C$ and $N$ are computable, and hence so is $M$. By Lemma 2.28 and the fact that $U = (U \cap N) \cup (U \cap M)$, it is enough to show that either $U \cap M \equiv_{\mathrm{m}} A$ or $U \cap M$ is computable.

But, combining Lemmas 2.18 and 2.30, we conclude that

1. if $r_n = 0$ then an element $x$ of $M$ is in $U$ if and only if, for some $j \in A$, $x$ is the coding location of the first copy of $[6j]$ to appear in $\mathcal{G}_n$, so that $U \cap M \equiv_{\mathrm{m}} A$, while
2. if $r_n = 1$ then an element $x$ of $M$ is in $U$ if and only if, for some $j \in \omega$, $x$ is the coding location of the second copy of $[6j]$ to appear in $\mathcal{G}_n$, so that $U \cap M$ is computable.

□

Theorem 1.7 follows from Lemmas 2.7, 2.25, and 2.31. ■

# 3 Proof of Theorem 1.9

**1.9. Theorem.** *Let $\{A_i\}_{i\in\omega}$ be a uniformly c.e. (u.c.e.) collection of sets. There exists an intrinsically c.e. relation $U$ on the domain of a computable structure $\mathcal{A}$ such that $DgSp_{\mathcal{A}}(U) = \{\deg(A_i) \mid i \in \omega\}$.*

*Proof.* Let $\{A_i\}_{i\in\omega}$ be a u.c.e. collection of sets. Let $A = \bigoplus_{i\in\omega} A_i = \{\langle i,x\rangle \mid x \in A_i\}$ and let $a_0, a_1, \ldots$ be a computable enumeration of $A$. Let $A[0] = \emptyset$, $A[s+1] = \{a_0, \ldots, a_s\}$. For $a \in \omega$, $\pi(a)$ will denote the first coordinate of the ordered pair coded by $a$. That is, if $a = \langle i,x\rangle$ then $\pi(a) = i$.

We wish to construct computable structures $\mathcal{A}^i$, $i \in \omega$, and for each such structure a corresponding unary relation $U^i$ on the domain of $\mathcal{A}^i$, so that for all $i,j \in \omega$, the following properties hold.

(3.1) $\mathcal{A}^i \cong \mathcal{A}^j$ via an isomorphism that carries $U^i$ to $U^j$.

(3.2) $U^i \equiv_{\mathrm{m}} A_i$.

(3.3) If $\mathcal{G} \cong \mathcal{A}^0$ is a computable structure then the image of $U^0$ in $\mathcal{G}$ is $m$-equivalent to $A_k$ for some $k \in \omega$.

The construction will be similar to the one in Section 2, as will the proof that the above properties hold. In this section, we restrict ourselves to pointing out the necessary changes.

We assume without loss of generality that, for all $i \in \omega$, $A_i \neq \emptyset$ and $A_i \neq \omega$. We also assume that $A$ is not computable. (If $A$ is computable then $\{\deg(A_i) \mid i \in \omega\} = \{\mathbf{0}\}$, and it is obvious that there exists a relation on the domain of a computable structure with degree spectrum $\{\mathbf{0}\}$.)

The basic idea is the following. Suppose that at stage $s+1$ we perform an **L**-operation involving copies of $[6a_s]$ and appropriate special components on $\mathcal{A}_s^{\pi(a_s)}$ and perform the corresponding **R**-operation on each $\mathcal{A}_s^j$, $j \neq \pi(a_s)$, and that we then put the coding location of the old copy of $[6a_s]$ in $\mathcal{A}_s^{\pi(a_s)}$ into $U^{\pi(a_s)}$ and, for each $j \neq \pi(a_s)$, we put the coding location of the new copy of $[6a_s]$ in $\mathcal{A}_s^j$ into $U^j$. Then the coding location $x$ of a copy of $[6\langle i,k\rangle]$, $k \in \omega$, in $\mathcal{A}^i$ is in $U^i$ if and only if $x \in \mathcal{A}_0^i$ and $k \in A_i$. On the other hand, the coding location of a copy of $[6\langle j,k\rangle]$, $k \in \omega$, $j \neq i$, is in $U^i$ if and only if it is not in $\mathcal{A}_0^i$. Thus (3.2) is satisfied.

However, there is a problem in defining the isomorphism recovery mechanism used to satisfy (3.1), due to the fact that both **L**- and **R**-operations are applied to a given $\mathcal{A}^i$ during the construction. We deal with this by separating the stages at which elements enter the $U^i$ from the stages at which isomorphism recovery can happen, reserving the even stages for the former purpose and the odd ones for the latter. (As before, we will say that $n$ is active at a stage if copies of the special component of $\mathcal{G}_n$ participate in operations at that stage, but the conditions that must be satisfied for this to happen will depend on whether the stage is even or odd.)

We now give the full description of the construction of the $\mathcal{A}^i$ and $U^i$.

*stage* 0. Let each $\mathcal{A}_0^i$, $i \in \omega$, be a computable structure with co-infinite domain, consisting of one copy of $[k]$ for each $k \in \omega$. For each $n \in \omega$, let $r_{n,0} = 0$.

*stage* $2s+1$. For each $n < s+1$, say that $2s+1$ is an $n$-recovery stage if all of the following conditions hold.

1. $\mathcal{G}_n[2s]$ has a special component isomorphic to some component of $\mathcal{A}_{2s}^0$. (Here "special component" has the same meaning as in the previous section.)

2. $(\mathcal{G}_n[2s])_n \cong (\mathcal{A}_{2s}^0)_n$.

3. Let $j \notin A[s]$ be less than or equal to the number of $n$-recovery stages before stage $2s+1$. There is a component of $\mathcal{G}_n[2s]$ isomorphic to $[6j]$, for each $l \in \{1,2\}$ there is a component of $\mathcal{G}_n[2s]$ isomorphic to $[6\langle n,j\rangle + l]$, and for each $l \in \{10,11\}$ there is a component of $\mathcal{G}_n[2s]$ isomorphic to $[12\langle n,j\rangle + l]$.

4. Let $c$ be the number of $n$-recovery stages before stage $2s+1$. For each $l \in \{4,5\}$ there is a component of $\mathcal{G}_n[2s]$ isomorphic to $[12\langle n,c\rangle + l]$.

If $2s+1$ is an $n$-recovery stage then, for $i \in \omega$, let $S^i_{n,2s}$ be the component of $\mathcal{A}^i_{2s}$ that is isomorphic to the special component of $\mathcal{G}_n[2s]$. If $2s+1$ is the first $n$-recovery stage then let $r_{n,2s+1} = 0$. Otherwise, proceed as follows. Let $i = r_{n,2s}$ and let $2t+1$ be the last $n$-recovery stage before stage $2s+1$. If $S^i_{n,2s}$ extends $S^i_{n,2t}$ then let $r_{n,2s+1} = i$. Otherwise, let $c$ be the number of $n$-change stages (defined below) before stage $2s+1$ and let $r_{n,2s+1} = \pi(c)$.

If $2s+1$ is not an $n$-recovery stage then let $r_{n,2s+1} = r_{n,2s}$.

We say that $2s+1$ is an $n$-change stage if it is the first $n$-recovery stage or $r_{n,2s+1} \neq r_{n,2s}$. We say that $2s+1$ is an $n$-isomorphism recovery stage if it is an $n$-recovery stage but not an $n$-change stage and one of the following conditions holds.

1. The last $n$-recovery stage before stage $2s+1$ was an $n$-change stage.

2. There has been at least one stage at which $n$ was active after the last $n$-isomorphism recovery stage and before stage $2s+1$.

Let $n_0, n_1, \ldots, n_m$ be all the numbers $n_k$ such that $2s+1$ is an $n_k$-recovery stage. We say that each $n_k$, $k \leqslant m$, is active at stage $2s+1$. For each $k \leqslant m$, proceed as follows. Let $i = r_{n_k,2s+1}$ and let $2t+1 \leqslant 2s+1$ be the last $n_k$-recovery stage. Let $R^i_{n_k,2s}$ be the component of $\mathcal{A}^i_{2s}$ that extends $S^i_{n_k,2t}$ and, for each $j \neq i$, let $R^j_{n_k,2s}$ be the isomorphic image of $R^i_{n_k,2s}$ in $\mathcal{A}^j_{2s}$. Let $c_k$ be the number of $n_k$-recovery stages before stage $2s+1$.

For each $k \leqslant m$, we define components $B^j_{n_k,2s}$ and $C^j_{n_k,2s}$, $j \in \omega$. There are two cases.

1. $2s+1$ is an $n_k$-isomorphism recovery stage. If the first condition in the definition of $n_k$-isomorphism recovery stage holds then let $t+1$ be the last $n_k$-recovery stage, and otherwise let $t+1$ be the first stage after the last $n_k$-isomorphism recovery stage at which $n_k$ was active. Let $C^i_{n_k,2s}$ be the component of $\mathcal{A}^i_{2s}$ that extends $B^i_{n_k,t}$ and, for $j \neq i$, let $C^j_{n_k,2s}$ be the isomorphic image of $C^i_{n_k,2s}$ in $\mathcal{A}^j_{2s}$. For $j \in \omega$, let $B^j_{n_k,2s}$ be the component of $\mathcal{A}^j_{2s}$ isomorphic to $[12\langle n_k, c_k\rangle + 4]$.

2. $2s+1$ is not an $n_k$-isomorphism recovery stage. For $j \in \omega$, let $B^j_{n_k,2s}$ and $C^j_{n_k,2s}$ be the components of $\mathcal{A}^j_{2s}$ isomorphic to $[12\langle n_k, c_k\rangle + 4]$ and $[12\langle n_k, c_k\rangle + 5]$, respectively.

For each $i \in \omega$, we define operations $\mathbf{O}^i_0, \ldots, \mathbf{O}^i_m$ as follows. If $i = r_{n_k,2s+1}$ then let $\mathbf{O}^i_k = \mathbf{L}(B^i_{n_k,2s}, R^i_{n_k,2s}, C^i_{n_k,2s})$. Otherwise, let $\mathbf{O}^i_k = \mathbf{R}(B^i_{n_k,2s}, R^i_{n_k,2s}, C^i_{n_k,2s})$.

For each $i \in \omega$, perform the sequence of operations $\mathbf{O}^i_0, \ldots, \mathbf{O}^i_m$ on $\mathcal{A}^i_{2s}$ to get $\mathcal{A}^i_{2s+1}$.

*stage* $2s+2$. For each $n \in \omega$, let $r_{n,2s+2} = r_{n,2s+1}$.

Let $l = \pi(a_s)$. Let $n_0, n_1, \ldots, n_m$ be all the numbers $n_j$ such that $a_s$ is less than the number of $n_j$-recovery stages before stage $2s+2$. We say that each $n_j$, $j \leqslant m$, is active at stage $2s+2$. For $i \in \omega$ and $j \leqslant m$, let $X^i_{2s+1}$, $Y^i_{n_j,2s+1}$, $Z^i_{n_j,2s+1}$, $B^i_{n_j,2s+1}$, and $C^i_{n_j,2s+1}$ be the components of $\mathcal{A}^i_{2s+1}$ isomorphic to $[6a_s]$, $[6\langle n_j, a_s\rangle + 1]$, $[6\langle n_j, a_s\rangle + 2]$, $[12\langle n_j, a_s\rangle + 10]$, and $[12\langle n_j, a_s\rangle + 11]$, respectively.

For each $k \leqslant m$, proceed as follows. Let $i = r_{n_k,2s+2}$ and let $2t+1$ be the last $n_k$-recovery stage before stage $2s+2$. Let $R^i_{n_k,2s+1}$ be the component of $\mathcal{A}^i_{2s+1}$ that extends $S^i_{n_k,2t}$ and, for each $j \neq i$, let $R^j_{n_k,2s+1}$ be the isomorphic image of $R^i_{n_k,2s+1}$ in $\mathcal{A}^j_{2s+1}$.

Now perform

$$\mathbf{L}(Y^l_{n_0,2s+1}, \ldots, Y^l_{n_m,2s+1}; X^l_{2s+1}; Z^l_{n_0,2s+1}, \ldots, Z^l_{n_m,2s+1}; B^l_{n_0,2s+1}, R^l_{n_0,2s+1}, C^l_{n_0,2s+1}; \\ B^l_{n_1,2s+1}, R^l_{n_1,2s+1}, C^l_{n_1,2s+1}; \ldots; B^l_{n_m,2s+1}, R^l_{n_m,2s+1}, C^l_{n_m,2s+1})$$

on $\mathcal{A}^l_{2s+1}$ to get $\mathcal{A}^l_{2s+2}$ and, for each $j \neq l$, perform

$$\mathbf{R}(Y^j_{n_0,2s+1}, \ldots, Y^j_{n_m,2s+1}; X^j_{2s+1}; Z^j_{n_0,2s+1}, \ldots, Z^j_{n_m,2s+1}; B^j_{n_0,2s+1}, R^j_{n_0,2s+1}, C^j_{n_0,2s+1}; \\ B^j_{n_1,2s+1}, R^j_{n_1,2s+1}, C^j_{n_1,2s+1}; \ldots; B^j_{n_m,2s+1}, R^j_{n_m,2s+1}, C^j_{n_m,2s+1})$$

on $\mathcal{A}^j_{2s+1}$ to get $\mathcal{A}^j_{2s+2}$. (If no $n$ is active at stage $2s+2$ then, for each $i \in \omega$, let $Y^i_{2s+1}$ and $Z^i_{2s+1}$ be the components of $\mathcal{A}^i_{2s+1}$ isomorphic to $[6\langle 0, a_s\rangle + 1]$ and $[6\langle 0, a_s\rangle + 2]$, respectively. Perform $\mathbf{L}(Y^l_{2s+1}, X^l_{2s+1}, Z^l_{2s+1})$ on $\mathcal{A}^l_{2s+1}$ to get $\mathcal{A}^l_{2s+2}$ and, for each $j \neq l$, perform $\mathbf{R}(Y^j_{2s+1}, X^j_{2s+1}, Z^j_{2s+1})$ on $\mathcal{A}^j_{2s+1}$ to get $\mathcal{A}^j_{2s+2}$.)

Put the coding location of the copy of $[6a_s]$ in $\mathcal{A}^l_0$ into $U^l$ and, for each $j \neq l$, put the coding location of the copy of $[6a_s]$ in $\mathcal{A}^j_{2s+2} - \mathcal{A}^j_{2s+1}$ into $U^j$.

This completes the construction. For each $i \in \omega$, let $\mathcal{A}^i = \bigcup_{s\in\omega} \mathcal{A}^i_s$. As previously remarked, the proof that (3.1)–(3.3) are satisfied is similar to what we did in Section 2. We begin by showing that (3.2) is satisfied.

**3.1 Lemma.** *For each* $i \in \omega$, $U^i \equiv_m A_i$.

*Proof.* If $k$ is the coding location of a copy of $[6\langle i, x\rangle]$ in $\mathcal{A}^i$ then $k \in U^i$ if and only if $k \in \mathcal{A}^i_0$ and $x \in A_i$. On the other hand, if $k$ is the coding location of a copy of $[6\langle j, x\rangle]$ in $\mathcal{A}^i$ for some $x \in \omega$, $j \neq i$, and $k$ enters $U^i$ at stage $s+1$ then $k$ is a new number at that stage, and hence is greater than $s$. □

Lemmas 2.8, 2.9, 2.10, 2.11, 2.12, 2.14, and 2.18 still hold, as do the following versions of Lemmas 2.13, 2.15, 2.16, and 2.17. In all cases, the reasoning is basically the same as in Section 2.

**3.2 Lemma.** *Let $i, j, s \in \omega$. $\mathcal{A}^i_s \cong \mathcal{A}^j_s$ and no component of $\mathcal{A}^i_s$ is embeddable in another component of $\mathcal{A}^i_s$. Furthermore, if a component of $\mathcal{A}^i_s$ participates in an operation at stage $s+1$ then so does the (unique) isomorphic component of $\mathcal{A}^j_s$.*

**3.3 Lemma.** *Let $2s+1$ be an $n$-recovery stage that is not the first such stage. Let $2t+1$ be the last $n$-recovery stage before stage $2s+1$ and suppose that $r_{n,2t+1} = i \neq r_{n,2s+1}$. Then for some $u \in [t, s)$, $S^i_{n,2s}$ extends one of $B^i_{n,2u}$, $B^i_{n,2u+1}$, or $C^i_{n,2u+1}$.*

**3.4 Lemma.** *Suppose that $r_{n,t} = i$ for all $t \geqslant s$. Then no component of $(\mathcal{A}^i)_n$ can participate in an operation more than twice after stage $s$ unless it extends $R^i_{n,t}$ for some $t \geqslant s$, while for $j \neq i$, no component of $(\mathcal{A}^j)_n$ can participate in an operation more than twice after stage $s$ unless it extends $C^i_{n,t}$ for some $t \geqslant s$ such that $t+1$ is an $n$-isomorphism recovery stage.*

**3.5 Lemma.** *Suppose that $s < t < v$ are such that $s+1$ is an $n$-isomorphism recovery stage, $r_{n,u} = r_{n,s+1}$ for all $u > s$, $t+1$ is the next stage after stage $s+1$ at which $n$ is active, and $v+1$ is the next $n$-isomorphism recovery stage after stage $s+1$. For $j \in \omega$, let $B^j$ and $R^j$ be the components of $\mathcal{A}^j_{t+1}$ that extend $B^j_{n,t}$ and $R^j_{n,t}$, respectively, and let $\widehat{B}^j$ and $\widehat{R}^j$ be the components of $\mathcal{A}^j_v$ that extend $B^j$ and $R^j$, respectively. Then $\widehat{B}^i \cong B^i$ and, for $j \neq i$, $\widehat{R}^j \cong R^j$.*

We now wish to show that (3.3) is satisfied. Lemma 2.20 still holds, and hence so does Lemma 2.26. In both cases the proofs are essentially the same as in Section 2. Using Lemma 3.3 in place of Lemma 2.15, we can prove Lemma 2.19 in much the same way as before. (Notice that the way we define $r_{n,2s+1}$ guarantees that if $r_{n,s}$ does not have a limit then for each $i \in \omega$ there are infinitely many stages $s$ such that $r_{n,s} = i$.) Now Lemma 2.27 follows, as before, from Lemmas 2.19 and 2.26.

Lemmas 2.28 and 2.29 still hold, with essentially the same proofs as in Section 2, provided that, in the latter lemma, we make the obvious changes arising from the fact that if $n$ is active at stage $2s+1$ then the components $B^i_{n,2s}$, $R^i_{n,2s}$, and $C^i_{n,2s}$ are defined but the components $Y^i_{n,2s}$, $X^i_{2s}$, and $Z^i_{n,2s}$ are not.

Now the following lemma can be proved in basically the same way as Lemma 2.30.

**3.6 Lemma.** *Suppose that $2s+1$ is an $n$-recovery stage such that $r_{n,2s+1} = r_{n,2s}$. Let $2t+1$ be the last $n$-recovery stage before stage $2s+1$ and let $j \in A[s] - A[t]$ be less than the number of $n$-recovery stages less than or equal to $2t+1$. By the definition of $n$-recovery stage, there is a unique component $K$ of $\mathcal{G}_n[2t]$ isomorphic to $[6j]$. Let $L$ be the component of $\mathcal{G}_n$ that extends $K$. Then $L$ contains a copy of $[6\langle n, j\rangle + 2]$ if and only if $r_{n,2s+1} = \pi(j)$.*

The previous lemma allows us to establish that (3.3) is satisfied.

**3.7 Lemma.** *Suppose that $\mathcal{G}_n \cong \mathcal{A}^0$. Let $U$ be the image of $U^0$ under this isomorphism. Then $U \equiv_m A_i$ for some $i \in \omega$.*

*Proof.* Let $N$ and $M$ be as in the proof of Lemma 2.31. By Lemma 2.28, it is enough to show that $U \cap M \equiv_{\mathrm{m}} A_i$ for some $i \in \omega$. By Lemma 2.27, $r_{n,s}$ has a limit $i$. Let $M_0$ be the set of elements of $M$ that are coding locations of copies of graphs of the form $[6n]$, $\pi(n) = i$, and let $M_1 = M - M_0$. Note that $M_0$ and $M_1$ are computable.

Now, combining Lemmas 2.18 and 3.6, we see that

1. an element $x$ of $M_0$ is in $U$ if and only if, for some $j \in A_i$, $x$ is the coding location of the first copy of $[6\langle i, j\rangle]$ to appear in $\mathcal{G}_n$, while

2. an element $x$ of $M_1$ is in $U$ if and only if, for some $k \in \omega$, $x$ is the coding location of the second copy of $[6k]$ to appear in $\mathcal{G}_n$.

So $U \cap M_0 \equiv_{\mathrm{m}} A_i$ and $U \cap M_1$ is computable, and thus $U \cap M \equiv_{\mathrm{m}} A_i$. □

We are left with showing that (3.1) is satisfied. Lemma 2.21 still holds, with basically the same proof as before. Lemma 2.22 still holds, but the proof needs to be slightly modified, so we restate the lemma and give the new proof.

**3.8 Lemma.** *Suppose that $n$ is active infinitely often and $s$ and $i$ are such that $r_{n,t} = r_{n,s} = i$ for all $t \geqslant s$. By Lemma 2.21, there are infinitely many $n$-isomorphism recovery stages. Let $s_0 + 1 < s_1 + 1 < \cdots$ be the $n$-isomorphism recovery stages after stage $s$. For each $j \in \omega$, let $t_j + 1$ be the next stage after stage $s_j + 1$ at which $n$ is active. (Note that $t_j < s_{j+1}$ for all $j \in \omega$.) For $t \geqslant t_0$, let $K^l_t$ be the component of $\mathcal{A}^l_t$ that extends $R^l_{n,t_0}$. Then $K^l_{t_j} = R^l_{n,t_j}$ for all $j \in \omega$.*

*Proof.* That $K^i_{s_j} = R^i_{n,s_j}$ for all $j \in \omega$ follows from Lemma 2.14.

Now let $l \neq i$ and assume by induction that $K^l_{t_j} = R^l_{n,t_j}$. Let $B$ be the component of $\mathcal{A}^i_{t_j+1}$ that extends $B^i_{n,t_j}$. By construction, $B \cong K^l_{t_j+1}$. Since $s_{j+1} + 1$ is an $n$-isomorphism recovery stage, $C^i_{n,s_{j+1}}$ extends $B$. Thus, by Lemma 3.5, $C^i_{n,s_{j+1}} \cong B$. By the same lemma, $K^l_{s_{j+1}} \cong K^l_{t_j+1}$, so $C^i_{n,s_{j+1}} \cong K^l_{s_{j+1}}$, and hence $C^l_{n,s_{j+1}} = K^l_{s_{j+1}}$. Let $R$ be the component of $\mathcal{A}^i_{s_{j+1}+1}$ that extends $R^i_{n,s_{j+1}}$. Then $R \cong K^l_{s_{j+1}+1}$. But, by Lemma 2.11, $R^i_{n,t_{j+1}} \cong R$ and $K^l_{t_{j+1}} \cong K^l_{s_{j+1}+1}$, so $K^l_{t_{j+1}} \cong R^i_{n,t_{j+1}}$, and hence $K^l_{t_{j+1}} = R^l_{n,t_{j+1}}$. □

If we assume the hypotheses of Lemma 3.8 and let $S^l_n$ be the component of $\mathcal{A}^l$ that extends $R^l_{n,s_0}$ then we can prove Lemma 2.23 in the same way as before, using Lemma 3.4 in place of Lemma 2.16. Furthermore, the following version of Lemma 2.24 follows directly from Lemma 3.8.

**3.9 Lemma.** *Assume the hypotheses of Lemma 3.8 and let $S^l_n$ be the component of $\mathcal{A}^l$ that extends $R^l_{n,s_0}$. Then $S^k_n \cong S^l_n$ for all $k, l \in \omega$.*

By the same reasoning as in Section 2 (using Lemma 3.2 in place of Lemma 2.13), Lemmas 2.23 and 3.9 suffice to establish that (3.1) is satisfied.

**3.10 Lemma.** *For each $i, j \in \omega$, $\mathcal{A}^i \cong \mathcal{A}^j$ via an isomorphism that carries $U^i$ to $U^j$.*

Theorem 1.9 follows from Lemmas 3.1, 3.7, and 3.10. ■

# 4 Proof of Theorem 1.11

**1.11. Theorem.** *Let $\alpha \in \omega \cup \{\omega\}$ and let $\mathbf{b} > \mathbf{0}$ be an $\alpha$-c.e. degree. There exists an intrinsically $\alpha$-c.e. relation $V$ on the domain of a computable structure $\mathcal{B}$ such that $DgSp_{\mathcal{B}}(V) = \{\mathbf{0}, \mathbf{b}\}$.*

*Proof.* Let $\alpha \in \omega \cup \{\omega\}$ and let $B$ be an $\alpha$-c.e. set that is not computable. It follows immediately from Definition 1.10 that there exist a computable sequence $b_0, b_1, \ldots \in \omega$ and a function $f$ such that

1. either $\alpha < \omega$ and $f(x) = \alpha$ for all $x \in \omega$ or $\alpha = \omega$ and $f$ is computable,
2. $|\{s \mid b_s = x\}| \leqslant f(x)$ for all $x \in \omega$, and
3. $x \in B \Leftrightarrow |\{s \mid b_s = x\}| \equiv 1 \bmod 2$.

Since the $\alpha = 0$ case is trivial, we may assume without loss of generality that $f(x) > 0$ for all $x \in \omega$.

We wish to construct computable structures $\mathcal{B}^0$ and $\mathcal{B}^1$ and unary relations $V^0$ and $V^1$ on the domains of $\mathcal{B}^0$ and $\mathcal{B}^1$, respectively, so that the following properties are satisfied.

(4.1) $\mathcal{B}^0 \cong \mathcal{B}^1$ via an isomorphism that carries $V^0$ to $V^1$.

(4.2) $V^0 \equiv_{\mathrm{m}} B$ and $V^1$ is computable.

(4.3) If $\mathcal{G} \cong \mathcal{B}^0$ is a computable structure then the image of $V^0$ in $\mathcal{G}$ is either computable or $m$-equivalent to $B$.

For each $s \in \omega$, let $c_s = |\{t < s \mid b_t = b_s\}|$ and let $a_s = \langle b_s, c_s \rangle$. Let $A = \{a_0, a_1, \ldots\}$. $A$ is clearly c.e. but not computable, so we can follow the construction in Section 2 to obtain computable structures $\mathcal{A}^0$ and $\mathcal{A}^1$ and relations $U^0$ and $U^1$ on the domains of $\mathcal{A}^0$ and $\mathcal{A}^1$, respectively, satisfying properties (2.1)–(2.3). (We assume that the construction has been carried out in such a way that the domains of $\mathcal{A}^0$ and $\mathcal{A}^1$ are co-infinite.)

Now, for $i = 0, 1$, proceed as follows. Add a node to $\mathcal{A}^i$ and add an edge from this node to each node of $\mathcal{A}^i$. For each $j \in \omega$ and each sequence of components $L_0, L_1, \ldots, L_{f(j)-1}$ such that each $L_k$ contains a copy of $[6\langle j, k \rangle]$, add an element $x$ (which will be said to be a *$j$-coding node*) to the domain of $\mathcal{A}^i$ and, for each $k < f(j)$, add an edge from $x$ to the coding location of the copy of $[6\langle j, k \rangle]$ in $L_k$. The resulting graph is $\mathcal{B}^i$.

Clearly, we can build each $\mathcal{B}^i$ so that it is a computable graph. We now define a relation $V^i$ on the domain of $\mathcal{B}^i$. Let $K^i$ be the set of coding nodes in $\mathcal{B}^i$.

Let $j \in \omega$ and let $x$ be a $j$-coding node in $\mathcal{B}^i$. By construction, there exist components $L_0, \ldots, L_{f(j)-1}$ of $\mathcal{A}^i$ such that, for each $k < f(j)$, $L_k$ contains a copy of $[6\langle j, k \rangle]$ whose coding location $y_k$ is attached to $x$. Let $c^i(x)$ be the least $k < f(j)$ such that $y_k \notin U^i$, if such a $k$ exists, and let $c^i(x) = f(j)$ otherwise. Now let $V^i = \{x \in K^i \mid c^i(x) \text{ is odd}\}$.

**4.1 Lemma.** *$\mathcal{B}^0 \cong \mathcal{B}^1$ via an isomorphism that carries $V^0$ to $V^1$.*

*Proof.* By (2.1), $\mathcal{A}^0 \cong \mathcal{A}^1$ via an isomorphism that carries $U^0$ to $U^1$. It is straightforward to extend this isomorphism to an isomorphism $h : \mathcal{B}^0 \cong \mathcal{B}^1$. The fact that $h(U^0) = (U^1)$ implies that if $x \in K^0$ then $c^0(x) = c^1(h(x))$. Thus $h(V^0) = V^1$. □

**4.2 Lemma.** *Suppose that $\mathcal{G}$ is computable and $h : \mathcal{B}^0 \cong \mathcal{G}$. Let $U = h(U^0)$ and $V = h(V^0)$. If $U$ is computable then so is $V$, while if $U \equiv_m A$ then $V \equiv_m B$.*

*Proof.* Let $\mathcal{G}' = h(\mathcal{A}^0)$ and $K = h(K^0)$. Note that both $\mathcal{G}'$ and $K$ are computable. Let $j \in \omega$ and let $x$ be a $j$-coding node in $\mathcal{G}$, by which we mean that $x = h(z)$ for some $j$-coding node $z$ in $\mathcal{A}^0$. By construction, there exist components $L_0, \ldots, L_{f(j)-1}$ of $\mathcal{G}'$ such that, for each $k < f(j)$, $L_k$ contains a copy of $[6\langle j,k\rangle]$ whose coding location $y_k$ is attached to $x$. Let $c(x)$ be the least $k < f(j)$ such that $y_k \notin U$, if such a $k$ exists, and let $c(x) = f(j)$ otherwise. Note that, by the definition of $V^0$, $x \in V$ if and only if $c(x)$ is odd.

By (2.3), either $U$ is computable or $U \equiv_m A$. First suppose that $U$ is computable. Then there is a computable procedure for determining $c(x)$ given $x \in K$, and thus $V$ is computable.

Now suppose that $U \equiv_m A$. Let $M$ be as defined in the proof of Lemma 2.31. Let $x \in K$ and let $y_0, \ldots, y_{f(j)-1}$ be as above. Let $d(x)$ be the least $k$ such that, for all $m \geqslant k$, $y_m \in M$ and $y_m$ is the coding location of the first copy of $[6\langle j,m\rangle]$ to appear in $\mathcal{G}$, if such a $k$ exists, and let $d(x) = f(j)$ otherwise. Note that there is a computable procedure for determining $d(x)$ given $x \in K$.

If $d(x) > 0$ then clearly $\langle j, d(x)-1\rangle \in A$. But this means that, in fact, $\langle j,k\rangle \in A$ for all $k < d(x)$. It follows that we can computably determine whether $y_k \in U$ for $k < d(x)$. So $S = \{x \in K \mid c(x) < d(x)\}$, $T = K - S$, and $V \cap S$ are computable.

Now let $x \in T$ be a $j$-coding node and let $y_0, \ldots, y_{f(j)-1}$ be as above. By the definition of $T$, $y_0, \ldots, y_{d(x)-1} \in U$, so $\langle j,k\rangle \in A$ for all $k < d(x)$. But, by the definition of $d(x)$, for each $k \geqslant d(x)$, $y_k \in U$ if and only if $\langle j,k\rangle \in A$. So $c(x) = |\{k \mid \langle j,k\rangle \in A\}| = |\{t \mid b_t = j\}|$. Thus $x \in V$ if and only if $j \in B$, and hence $V \cap T \equiv_m B$. Since $V = (V \cap S) \cup (V \cap T)$, it follows that $V \equiv_m B$. □

**4.3 Corollary.** *$V^0 \equiv_m B$ and $V^1$ is computable.*

**4.4 Corollary.** *Suppose that $\mathcal{G}$ is computable and $h : \mathcal{B}^0 \cong \mathcal{G}$, and let $V = h(V^0)$. Then either $V$ is computable or $V \equiv_m B$.*

Theorem 1.11 follows from Lemma 4.1 and Corollaries 4.3 and 4.4. ■

As mentioned in Section 1, the modifications to the proof of Theorem 1.7 presented in this section can be combined with those presented in Section 3 to yield the following result.

**1.12. Theorem.** *Let $\alpha \in \omega \cup \{\omega\}$ and let $\{A_i\}_{i\in\omega}$ be a uniformly $\alpha$-c.e. collection of sets. There exists an intrinsically $\alpha$-c.e. relation $V$ on the domain of a computable structure $\mathcal{B}$ such that $DgSp_{\mathcal{B}}(V) = \{\deg(A_i) \mid i \in \omega\}$.*

# References


[1] C.J. Ash, P. Cholak, and J.F. Knight, Permitting, Forcing, and Copying of a Given Recursive Relation, *Ann. Pure Appl. Logic* **86** (1997), 219–236.

[2] C.J. Ash and A. Nerode, Intrinsically Recursive Relations, in J.N. Crossley (ed.), *Aspects of Effective Algebra (Clayton, 1979)*, Upside Down A Book Co., Yarra Glen, Australia, 1981, 26–41.

[3] E. Barker, Intrinsically $\Sigma^0_\alpha$ Relations, *Ann. Pure Appl. Logic* **39** (1988), 105–130.

[4] P. Cholak, S.S. Goncharov, B. Khoussainov, and R.A. Shore, Computably Categorical Structures and Expansions by Constants, *J. Symbolic Logic* **64** (1999), 13–37.

[5] S.B. Cooper, L. Harrington, A.H. Lachlan, S. Lempp, and R.I. Soare, The d.r.e. Degrees are Not Dense, *Ann. Pure Appl. Logic* **55** (1991), 125–151.

[6] R.L. Epstein, R. Haas, and R.L. Kramer, Hierarchies of Sets and Degrees below $\mathbf{0}'$, in M. Lerman, J.H. Schmerl, and R.I. Soare (eds.), *Logic Year 1979–80 (Proc. Seminars and Conf. Math. Logic, Univ. Connecticut, Storrs, Conn., 1979/80)*, Lecture Notes in Math. **859**, Springer-Verag, Heidelberg, 1981, 32–48.

[7] Yu.L. Ershov, S.S. Goncharov, A. Nerode, J.B. Remmel (eds.), *Handbook of Recursive Mathematics*, Stud. Logic Found. Math. **138–139**, Elsevier Science, Amsterdam, 1998.

[8] V.S. Harizanov, The Possible Turing Degree of the Nonzero Member in a Two Element Degree Spectrum, *Ann. Pure Appl. Logic* **60** (1993), 1–30.

[9] D.R. Hirschfeldt, B. Khoussainov, A.M. Slinko, and R.A. Shore, Degree Spectra and Computable Dimension in Algebraic Structures, to appear.

[10] S.S. Goncharov, Autostability of Models and Abelian Groups, *Algebra and Logic* **19** (1980), 13–27.

[11] S.S. Goncharov, Computable Single-Valued Numerations, *Algebra and Logic* **19** (1980), 325–356.

[12] S.S. Goncharov, Problem of the Number of Non-Self-Equivalent Constructivizations, *Algebra and Logic* **19** (1980), 401–414.

[13] S.S. Goncharov and B. Khoussainov, On the Spectrum of Degrees of Decidable Relations, *Doklady Math.* **55** (1997), 55–57 (research announcement).

[14] B. Khoussainov and R.A. Shore, Computable Isomorphisms, Degree Spectra of Relations, and Scott Families, *Ann. Pure Appl. Logic* **93** (1998), 153–193.

[15] R.I. Soare, *Recursively Enumerable Sets and Degrees*, Perspect. Math. Logic, Springer-Verlag, Heidelberg, 1987.