ELSEVIER
Annals of Pure and Applied Logic 115 (2002) 233–277
ANNALS OF PURE AND APPLIED LOGIC
www.elsevier.com/locate/apal

Degree spectra of relations on structures of finite computable dimension

Denis R. Hirschfeldt *
School of Mathematical and Computing Sciences, Victoria University of Wellington, New Zealand

Received 27 January 2001; received in revised form 11 April 2001; accepted 11 April 2001
Communicated by R.I. Soare

Abstract

We show that for every computably enumerable (c.e.) degree $\mathbf{a} > \mathbf{0}$ there is an intrinsically c.e. relation on the domain of a computable structure of computable dimension 2 whose degree spectrum is $\{\mathbf{0}, \mathbf{a}\}$, thus answering a question of Goncharov and Khoussainov (Dokl. Math. 55 (1997) 55–57). We also show that this theorem remains true with α-c.e. in place of c.e. for any $\alpha \in \omega \cup \{\omega\}$. A modification of the proof of this result similar to what was done in Hirschfeldt (J. Symbolic Logic, to appear) shows that for any $\alpha \in \omega \cup \{\omega\}$ and any α-c.e. degrees $\mathbf{a}_0, \ldots, \mathbf{a}_n$ there is an intrinsically α-c.e. relation on the domain of a computable structure of computable dimension $n+1$ whose degree spectrum is $\{\mathbf{a}_0, \ldots, \mathbf{a}_n\}$. These results also hold for m-degree spectra of relations.

MSC: 03C57; 03D45; 03C15

Keywords: Degree spectra of relations; Computable structures; Computable dimension

1. Introduction

The study of the effective content of model theory has proved quite fertile, and has attracted the attention of a large number of researchers. The recent publication of the *Handbook of Recursive Mathematics* [9], the first volume of which is dedicated to effective model theory, attests to the growth of the field. (This handbook is a valuable reference; in particular, the introduction and the articles by Ershov and Goncharov [8] and Harizanov [16] give useful overviews, while the articles by Ash [1] and Goncharov

* 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: Department of Mathematics, University of Chicago, 5734 S. University Ave., Chicago, IL 60637, USA. Tel.: +1-773-702-5353; fax: +1-773-702-9787.
E-mail address: drh@math.uchicago.edu (D.R. Hirschfeldt).

0168-0072/02/$ - see front matter
PII: S0168-0072(01)00094-X

[12] cover material related to the topic of this paper. Another relevant survey article is [21].)

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

Definition 1.1. A structure $\mathscr{A}$ in a computable language is *computable* if both its domain $|\mathscr{A}|$ and the atomic diagram of $\langle \mathscr{A}, a\rangle_{a\in|\mathscr{A}|}$ are computable.

Providing effective analogs of theorems of classical model theory (and showing that in certain cases there are none) is part of the work of computable model theory. Another part consists of analyzing phenomena that only arise in the computable setting, such as the fact that isomorphic computable structures, which are considered to be essentially identical in classical model theory, might behave quite differently from a computability-theoretic point of view.

For example, under the standard ordering of ω, the successor relation is computable, but it is not hard to construct a computable linear ordering of type ω in which the successor relation is not computable (see, for instance, [6]). In fact, for every computably enumerable (c.e.) degree **a**, we can construct a computable linear ordering of type ω in which the successor relation has degree **a**. It is also possible to build two isomorphic computable groups, only one of which has a computable center, or two isomorphic Boolean algebras, only one of which has a computable set of atoms. This leads us to study computable structures up to *computable* isomorphism, a point of view reflected in the following definition.

Definition 1.2. An isomorphism from a structure $\mathscr{M}$ to a computable structure is called a *computable presentation* of $\mathscr{M}$. (We often abuse terminology and refer to the image of a computable presentation as a computable presentation.)

If $\mathscr{M}$ has a computable presentation then it is *computably presentable*.

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

A structure of computable dimension 1 is said to be *computably categorical*.

We will also have occasion to consider c.e. structures.

Definition 1.3. A structure $\mathscr{A}$ is c.e. if its domain $|\mathscr{A}|$ is computable and the atomic diagram of $\langle \mathscr{A}, a\rangle_{a\in|\mathscr{A}|}$ is c.e.

An isomorphism from a structure $\mathscr{M}$ to a c.e. structure is called a *c.e. presentation* of $\mathscr{M}$. (As in the computable case, we often refer to the image of a c.e. presentation as a c.e. presentation.)

If $\mathscr{M}$ has a c.e. presentation then it is *c.e. presentable*.

The *c.e. dimension* of a c.e. presentable structure $\mathscr{M}$ is the number of c.e. presentations of $\mathscr{M}$ up to computable isomorphism.

The examples mentioned above of structures that are isomorphic but not computably isomorphic, as well as many other natural ones, suggest the idea of attempting to understand the differences between noncomputably isomorphic computable presentations of a structure $\mathcal{M}$ by comparing (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 [3] and has been continued in a large number of papers. (References can be found in the aforementioned articles in [9], as well as in [21].)

Ash and Nerode [3] were concerned with relations that maintain some degree of effectiveness in different computable presentations of a structure.

Definition 1.4. Let U be a relation on the domain of a computably presentable structure $\mathcal{M}$ and let $\mathfrak{C}$ be a class of relations. U is *intrinsically* $\mathfrak{C}$ on $\mathcal{M}$ if the image of U in any computable presentation of $\mathcal{M}$ is in $\mathfrak{C}$.

In [3], conditions that guarantee that a relation is intrinsically computable or intrinsically 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 [4], for example).

A different approach to the study of relations on computable structures, which began with the work of Harizanov [14] (although there is some earlier work, for instance by Remmel in [23], that can be thought of in this light), is to look at the (Turing) degrees of the images of a relation in different computable presentations of a structure.

Definition 1.5. Let U be a relation on the domain of a computably presentable structure $\mathcal{M}$. The *degree spectrum* of U on $\mathcal{M}$, $\mathrm{DgSp}_{\mathcal{M}}(U)$, is the set of degrees of the images of U in all computable presentations of $\mathcal{M}$.

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

Definition 1.6. Let r be a reducibility, such as many–one reducibility (m-reducibility) or weak truth-table reducibility (wtt-reducibility). Let U be a relation on the domain of a computably presentable structure $\mathcal{M}$. The *r-degree spectrum* of U on $\mathcal{M}$, $\mathrm{DgSp}^{\mathrm{r}}_{\mathcal{M}}(U)$, is the set of r-degrees of the images of U in all computable presentations of $\mathcal{M}$.

Ash–Nerode type conditions often 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 [2] or [17], for example). Motivated by these considerations, as well as by Goncharov's examples [11] of structures of finite computable

dimension, Harizanov and Millar suggested the study of relations with finite degree spectra.

Harizanov [15] was the first to give an example of an intrinsically Δ^0_2 relation with a two-element degree spectrum that includes $\mathbf{0}$. (Harizanov also noted that Goncharov's example of a rigid structure of computable dimension 2 can be converted into an example of an intrinsically Δ^0_3 relation with a two-element degree spectrum that includes $\mathbf{0}$.)

Theorem 1.7 (Harizanov). *There exist a Δ^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* 2 *such that* $\mathrm{DgSp}_{\mathcal{A}}(U) = \{\mathbf{0}, \mathbf{a}\}$.

Khoussainov and Shore [20] and Goncharov and Khoussainov [13] showed the existence of an intrinsically c.e. relation with a two-element degree spectrum.

Theorem 1.8 (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* 2 *such that* $\mathrm{DgSp}_{\mathcal{A}}(U) = \{\mathbf{0}, \mathbf{a}\}$.

This left open the question, asked explicitly in [13], of which (c.e.) degrees can be the nonzero element of a two-element degree spectrum of a relation on a structure of computable dimension 2. A partial answer to this question was given by the author in [18], where the following result was established.

Theorem 1.9. *Let* $\mathbf{a} > \mathbf{0}$ *be a c.e. degree. There is an intrinsically c.e. relation U on the domain of a computable structure $\mathcal{A}$ such that* $\mathrm{DgSp}_{\mathcal{A}}(U) = \{\mathbf{0}, \mathbf{a}\}$.

In this paper, we improve this result by showing that $\mathcal{A}$ can be chosen to have computable dimension 2, thus fully answering the question mentioned above in the c.e. case. Our proof is such that we are able to control not only the computable dimension but also the c.e. dimension of the structures we build (which was also the case in [20]).

Theorem 1.10. *Let* $\mathbf{a} > \mathbf{0}$ *be a c.e. degree. There is an intrinsically c.e. relation U on the domain of a computable structure $\mathcal{A}$ of computable dimension* 2 *such that* $\mathrm{DgSp}_{\mathcal{A}}(U) = \{\mathbf{0}, \mathbf{a}\}$. *In addition, $\mathcal{A}$ can be picked so that every c.e. presentation of $\mathcal{A}$ is computable, which implies that $\mathcal{A}$ has c.e. dimension* 2.

This result and its extensions, Theorems 1.12 and 1.14 below, are also due independently to Khoussainov and Shore [22], whose proofs use a complicated modification of their proof of Theorem 1.8.

The proof of Theorem 1.10, which appears in Section 2, is based on the proof of Theorem 1.9, and uses techniques from [20], which in turn builds on work of Goncharov [10, 11] and Cholak, Goncharov, Khoussainov and Shore [5].

We can extend Theorem 1.10 by broadening our focus from the c.e. degrees to larger classes of Δ^0_2 degrees.

Definition 1.11. Let $A \subseteq \omega$ be a set. A computable sequence $a_0, a_1, \ldots$ is a Δ^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 Δ^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 Δ^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 α-c.e. if it contains an α-c.e. set. A collection of sets $\{A_i\}_{i \in \omega}$ is *uniformly α-c.e.* if $\bigoplus_{i \in \omega} A_i = \{\langle i, x\rangle \mid x \in A_i\}$ is α-c.e.

Remark. The above definition of ω-c.e. is the one that will be useful in Section 3. There is an equivalent definition which can be generalized to define the concepts of α-c.e. set and α-c.e. degree for any computable ordinal α (see [7]). It is also interesting to note that a set is ω-c.e. if and only if it is wtt-reducible to $\emptyset'$.

Theorem 1.12. *Let $\alpha \in \omega \cup \{\omega\}$ and let $\mathbf{b} > \mathbf{0}$ be an α-c.e. degree. There is an intrinsically α-c.e. relation V on the domain of a computable structure $\mathscr{B}$ of computable dimension* 2 *such that* $\mathrm{DgSp}_{\mathscr{B}}(V) = \{\mathbf{0}, \mathbf{b}\}$. *In addition, $\mathscr{B}$ can be picked so that every c.e. presentation of $\mathscr{B}$ is computable, which implies that $\mathscr{B}$ has c.e. dimension* 2.

The structure $\mathscr{B}$, which will be described in Section 3, will be an extension of the structure $\mathscr{A}$ constructed in the proof of Theorem 1.10 for an appropriate c.e. degree $\mathbf{a}$.

In [18], the following extension of Theorem 1.9 was established.

Theorem 1.13. *Let $\{A_i\}_{i \in \omega}$ be a uniformly c.e. collection of sets. There is an intrinsically c.e. relation U on the domain of a computable structure $\mathscr{A}$ such that* $\mathrm{DgSp}_{\mathscr{A}}(U) = \{\deg(A_i) \mid i \in \omega\}$.

Khoussainov and Shore [20] showed that, for each $n \in \omega$, there exist c.e. degrees $\mathbf{a}_0, \ldots, \mathbf{a}_n$ and an intrinsically c.e. relation U on the domain of a computable structure $\mathscr{A}$ of computable dimension $n+2$ such that $\mathrm{DgSp}_{\mathscr{A}}(U) = \{\mathbf{0}, \mathbf{a}_0, \ldots, \mathbf{a}_n\}$. It is straightforward to combine the proofs of Theorems 1.10 and 1.12 with that of Theorem 1.13, given in Section 3 of [18], to yield the following strengthening of that result.

Theorem 1.14. *Let $\alpha \in \omega \cup \{\omega\}$ and let $\mathbf{a}_0, \ldots, \mathbf{a}_n$ be α-c.e. degrees. There is an intrinsically α-c.e. relation U on the domain of a computable structure $\mathscr{A}$ of computable dimension $n+1$ such that* $\mathrm{DgSp}_{\mathscr{A}}(U) = \{\mathbf{a}_0, \ldots, \mathbf{a}_n\}$. *In addition, $\mathscr{A}$ can be picked so that every c.e. presentation of $\mathscr{A}$ is computable, which implies that $\mathscr{A}$ has c.e. dimension $n+1$.*

The proofs of Theorems 1.10 and 1.12 are such that these theorems remain true with *degree* replaced by *m-degree* and $\mathrm{DgSp}_{\mathscr{A}}(U)$ replaced by $\mathrm{DgSp}^{\mathrm{m}}_{\mathscr{A}}(U)$. The same

holds of Theorem 1.14 if we require that the m-degrees of $\emptyset$ and ω not be on the list $\mathbf{a}_0, \ldots, \mathbf{a}_n$. Thus, for any reducibility r weaker than m-reducibility, these theorems remain true with *degree* replaced by *r-degree* and $\mathrm{DgSp}_{\mathscr{A}}(U)$ replaced by $\mathrm{DgSp}^{\mathrm{r}}_{\mathscr{A}}(U)$.

By the results of [19], for each of the following theories, Theorems 1.10, 1.12, and 1.14 remain true if we also require that the structures mentioned in them be models of the given theory, and that the relations mentioned in them be submodels: symmetric, irreflexive graphs; partial orderings; lattices; rings (with zero-divisors); integral domains of arbitrary characteristic; commutative semigroups; and 2-step nilpotent groups.

2. Proof of Theorem 1.10

In this section we prove the following theorem.[1]

Theorem 1.10. *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* $\mathscr{A}$ *of computable dimension* 2 *such that* $\mathrm{DgSp}_{\mathscr{A}}(U) = \{\mathbf{0}, \mathbf{a}\}$. *In addition,* $\mathscr{A}$ *can be picked so that every c.e. presentation of* $\mathscr{A}$ *is computable, which implies that* $\mathscr{A}$ *has c.e. dimension* 2.

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 $A[0] = \emptyset$ and $A[s+1] = \{a_0, \ldots, a_s\}$. We wish to construct computable structures $\mathscr{A}^0$ and $\mathscr{A}^1$ and unary relations U^0 and U^1 on the domains of $\mathscr{A}^0$ and $\mathscr{A}^1$, respectively, so that the following properties hold:

(2.1) $\mathscr{A}^0 \cong \mathscr{A}^1$ 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 $\mathscr{G} \cong \mathscr{A}^0$ is a computable structure then $\mathscr{G}$ is computably isomorphic to either $\mathscr{A}^0$ or $\mathscr{A}^1$.

(2.4) $\mathscr{A}^0$ is rigid.

(2.5) Every c.e. presentation of $\mathscr{A}^0$ with computable equality relation is computable.

The reason that (2.5) is enough to establish the last part of Theorem 1.10 is that we can let $\mathscr{A}$ be the result of adding to $\mathscr{A}^0$ the binary relation Q that holds of x and y if and only if $x \neq y$. Clearly, $\mathscr{A}$ shares all the relevant computable properties of $\mathscr{A}^0$, and any c.e. presentation of $\mathscr{A}$ restricts to a c.e. presentation of $\mathscr{A}^0$ with computable equality relation.

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

[1] The construction in this section will be similar in many ways to what was done in [18] to prove the result we have numbered Theorem 1.9, as will the proof that properties (2.1) and (2.2) below hold. (The construction in [18] also satisfied (2.4) below, but this was not mentioned in that paper because it was not needed to prove Theorem 1.9.) There will also be similarities with certain aspects of the proof in [20] of the result we have numbered Theorem 1.8. However, we will not assume that the reader is familiar with these papers, and will make our discussion below, as well as the formal proof that follows it, self-contained.

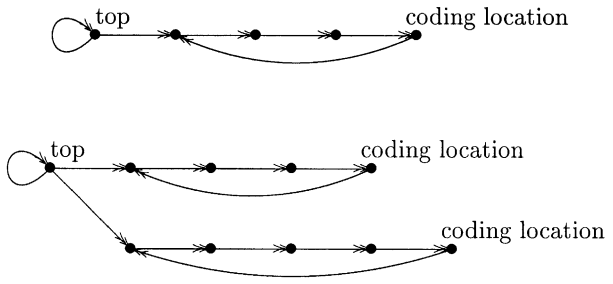

Fig. 1. [2] and [{2, 3}].

Definition 2.1. Let $n \in \omega$. The directed graph $[n]$ consists of $n+3$ many 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* of $[n]$ 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.

Fig. 1 shows [2] and $[\{2,3\}]$ as examples.

The description of the construction of $\mathscr{A}^0$, $\mathscr{A}^1$, U^0, and U^1 will be organized as follows. In Section 2.1, we discuss how we satisfy (2.1) and (2.2). Before dealing with the satisfaction of (2.3) in Section 2.3, we review in Section 2.2 what was done in the proof of Theorem 1.9 to satisfy the corresponding property. Formal definitions and conventions are given in Section 2.4; this is followed by the formal construction in Section 2.5 and its verification in Section 2.6.

We will ignore (2.5) until we give the formal definitions used in the full construction; at that point, we will introduce a few minor changes to ensure the satisfaction of this property. We will also not make explicit mention in our informal discussion below of how (2.4) is satisfied, but it should be clear that the construction we describe ensures the rigidity of the $\mathscr{A}^i$, and we will assume this fact in our discussion. We will also assume in our discussion that our construction is such that no connected component of $\mathscr{A}^i$ is embeddable in another component of $\mathscr{A}^i$.

2.1. *Satisfying* (2.1) *and* (2.2)

We build $\mathscr{A}^0$ and $\mathscr{A}^1$ in stages. We begin by letting $\mathscr{A}^0_0$ and $\mathscr{A}^1_0$ be computable structures with co-infinite domains, each consisting of one copy of $[k]$ for each $k \in \omega$. (This will change slightly when we introduce the changes needed to satisfy (2.5).) If at each stage $s+1$ we enumerate the coding location of the copy of $[3a_s]$ in $\mathscr{A}^0_0$ 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 $\mathscr{A}^0 \cong \mathscr{A}^1$ via an isomorphism that carries U^0 to U^1. To describe how we can do this, we need two more definitions.

Definition 2.2. 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 performed on $\mathcal{G}$, so that no two operations in the sequence affect the same component of $\mathcal{G}$, we can perform all the operations in the sequence simultaneously on $\mathcal{G}$ to get a structure extending $\mathcal{G}$. In this case we will say that we have performed the sequence of operations on $\mathcal{G}$.

Definition 2.3. 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 performing the sequence of operations $X_0 \cdot X_1,\ X_1 \cdot X_2, \ldots, X_n \cdot X_0$ on $\mathcal{G}$.
- The **R**-operation $\mathbf{R}(X_0, \ldots, X_n)$ consists of performing the sequence of operations $X_0 \cdot X_n,\ X_1 \cdot X_0, \ldots, X_n \cdot X_{n-1}$ on $\mathcal{G}$.

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

Fig. 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). It is also true that 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.

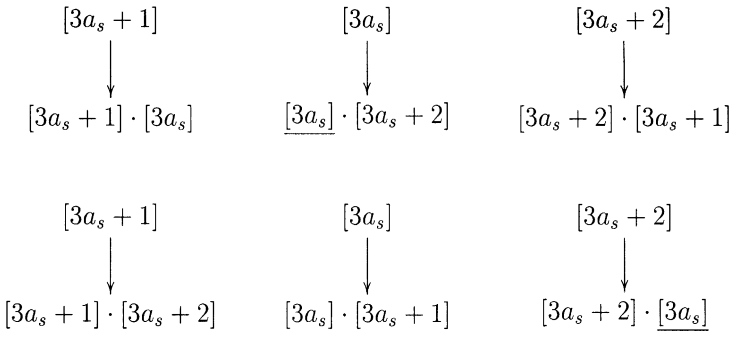

Fig. 2. The basic coding strategy (top: $\mathscr{A}^0$ / bottom: $\mathscr{A}^1$).

Furthermore, it is still the case 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 $\mathscr{A}_0^0$ for some $a \in A$. On the other hand, any number put into U^1 at a stage $s+1$ is a new number, and is therefore greater than s, which implies that U^1 is computable.

2.2. The proof of Theorem 1.9

Before we turn to the satisfaction of (2.3), it will be useful to discuss what was done in the proof of Theorem 1.9 to satisfy the following weaker condition, which, together with (2.1) and (2.2), clearly implies that theorem.

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

Our strategy for satisfying (2.3) will be quite similar to that used to satisfy (2.3′), although the proof that it succeeds will be significantly more involved.

The way in which (2.3′) can be satisfied for a single $\mathscr{G}$ is based on the following observation.

Let U be the image of U^0 in $\mathscr{G}$ and let $\mathscr{G}[s]$ denote the stage s approximation to $\mathscr{G}$. Assume that, for all $s \in \omega$, no component of $\mathscr{A}_s^i$ is embeddable in another component of $\mathscr{A}_s^i$ and $\mathscr{G}[s]$ is embeddable in $\mathscr{A}_s^0$. The latter assumption can be made because we only care about $\mathscr{G}$ if it is isomorphic to $\mathscr{A}^0$.

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

So if $\mathscr{G} \cong \mathscr{A}^0$ 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 $\mathscr{G}$ "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 $\mathscr{G}$ "goes to the right".

Now if the coding location of X_s^0 is put into U^0 and the coding location of the new copy of X_s^1 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 $\mathscr{G}$ goes to the left then the coding location of X_s in $\mathscr{G}[s]$ is in U, while if $\mathscr{G}$ goes to the right then the coding location of the copy of X_s in $\mathscr{G} - \mathscr{G}[s]$ is in U. It is easy to conclude from this that if $\mathscr{G}$ goes to the left at all but finitely many stages then $U \equiv_{\mathrm{m}} A$, while if $\mathscr{G}$ goes to the right at all but finitely many stages then U is computable.

So to satisfy (2.3′) it is enough to ensure that $\mathscr{G}$ 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 $\mathscr{G}$, which we will call the *special component* of $\mathscr{G}$, as S_s.

That is, we first pick some component of $\mathscr{G}$ to be its special component. Say we pick the one that extends the first copy of [0] to appear in $\mathscr{G}$. (Let us assume that $0 \notin A$). At stage 0, we define $\mathscr{A}_0^i$ as above and wait until a copy of [0] is enumerated into $\mathscr{G}$. We also define r_0 to be 0. The value of r_s will code whether $\mathscr{G}$ 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 $\mathscr{A}_s^i$ of $[3a_s]$, $[3a_s+1]$, and $[3a_s+2]$, respectively, and let S_s^i be the isomorphic copy in $\mathscr{A}_s^i$ of the special component S_s of $\mathscr{G}[s]$. We wait until copies of X_s^i, Y_s^i, and Z_s^i are enumerated into $\mathscr{G}[s]$ 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 $\mathscr{G}$. 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 $\mathscr{G} \cong \mathscr{A}^0$ then the special component of $\mathscr{G}$ is infinite. On the other hand, it also guarantees that if $\mathscr{G}$ changes direction infinitely often (that is, if r_s does not have a limit) then no component of $\mathscr{A}^0$ is infinite, so that $\mathscr{G} \ncong \mathscr{A}^0$. This is because, for each $s \in \omega$, the copy of the special component of $\mathscr{G}[s+1]$ in $\mathscr{A}_{s+1}^{1-r_{s+1}}$ is a component that participates in an operation for the first time at stage $s+1$. Fig. 3 illustrates the case $r_{s+1} = 0$. In this figure, the special components of $\mathscr{G}[s]$ and $\mathscr{G}[s+1]$ and their images are shown in boxes.

However, there are two problems with this construction. First of all, by the same reasoning as in the last paragraph, if $\mathscr{G}$ almost always goes to the left then no component of $\mathscr{A}^1$ is infinite, while if $\mathscr{G}$ almost always goes to the right then no component of $\mathscr{A}^0$ is infinite. In either case, (2.1) no longer holds.

This problem can be solved 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 $\mathscr{A}_s^0$ and isomorphic components B_s^1 and C_s^1 of $\mathscr{A}_s^1$, perform

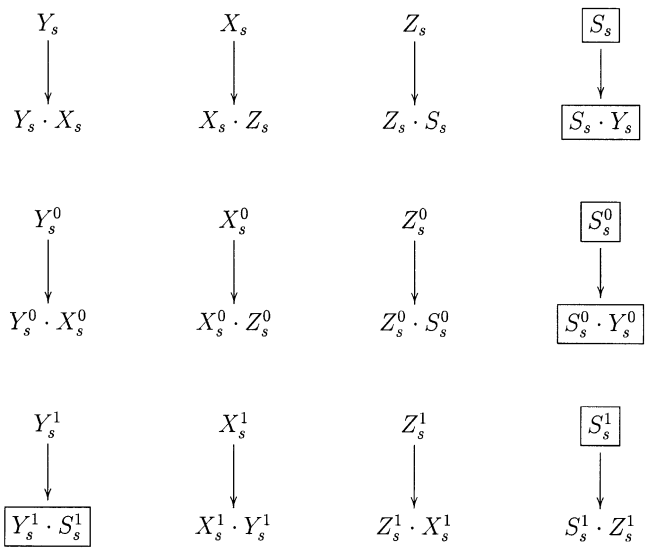

Fig. 3. The images of the special component (top: $\mathscr{G}$ / middle: $\mathscr{A}^0$ / bottom: $\mathscr{A}^1$).

$\mathbf{L}(Y_s^0, X_s^0, Z_s^0, B_s^0, S_s^0, C_s^0)$ on $\mathscr{A}_s^0$ to get $\mathscr{A}_{s+1}^0$, and perform $\mathbf{R}(Y_s^1, X_s^1, Z_s^1, B_s^1, S_s^1, C_s^1)$ on $\mathscr{A}_s^1$ to get $\mathscr{A}_{s+1}^1$. (In order to accommodate the extra components, X_s^i can be the copy of $[6a_s]$ in $\mathscr{A}_s^i$, and a similar change can be made for the other components.)

As long as $\mathscr{G}$ 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 $\mathscr{A}_s^0$ that extends B_{s-1}^0 and let C_s^1 be the isomorphic component of $\mathscr{A}_s^1$. On the other hand, if $r_s = 1$ then we let B_s^1 be the component of $\mathscr{A}_s^1$ that extends C_{s-1}^1 and let B_s^0 be the isomorphic component of $\mathscr{A}_s^0$. Whenever $\mathscr{G}$ 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 $\mathscr{G}$ in $\mathscr{A}^0$ and $\mathscr{A}^1$ are isomorphic, while still ensuring that if r_s does not have a limit then no component of $\mathscr{A}^0$ or $\mathscr{A}^1$ is infinite. We will give an example below to illustrate isomorphism recovery.

Another problem that had to be faced in the proof of Theorem 1.9, and will have to be faced here also, is that, in general, we cannot know in advance whether a given computable structure $\mathscr{G}$ is isomorphic to $\mathscr{A}^0$, so it is not possible to wait at each stage until the appropriate components are enumerated into $\mathscr{G}$. To get around this, the notion of a *recovery stage* can be used.

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

A recovery stage in the sense of the proof of Theorem 1.9 is then a stage $s+1$ such that

1. $\mathscr{G}[s]$ 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 $s+1$, $\mathscr{G}[s]$ contains components that can be used as Y_t, X_t, Z_t, B_t, and C_t if $a_t = j$ for some $t > s$.

(As we will see in the next section, we will need a somewhat more complicated version of this concept.)

Now suppose that $\mathscr{G} \cong \mathscr{A}^0$. Say that $\mathscr{G}$ 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 $\mathscr{G}$ can be active while waiting for recovery.

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

Arguing as before, we conclude that if $\mathscr{G}$ almost always goes to the left then $U \cap M \equiv_{\mathrm{m}} A$, while if $\mathscr{G}$ 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 jth recovery stage then we can tell whether $j \in P$, and if $j \notin P$ then $j \in A$. So if $\mathscr{G}$ almost always goes to the left then $U \equiv_{\mathrm{m}} A$, while if $\mathscr{G}$ almost always goes to the right then U is computable. Thus $(2.3')$ is satisfied for this $\mathscr{G}$.

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 a stage cannot be an isomorphism recovery stage unless it is a *first stage*, that is, the first stage at which $\mathscr{G}$ 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 in the construction described above when $\mathscr{G}$ 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 $\mathscr{G}$ is active.

Fig. 4 pictures what happens on the $\mathscr{A}^0$ side of the construction. From now on, we will use the notation R^i_s 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 $\mathscr{G}[w]$ if $w+1$ is not a recovery stage.

Note that, by the definition of recovery stage, the special component of $\mathscr{G}[s]$ is isomorphic to R^0_s and, for each $w = s, t, u$, $\mathscr{G}[s]$ 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.

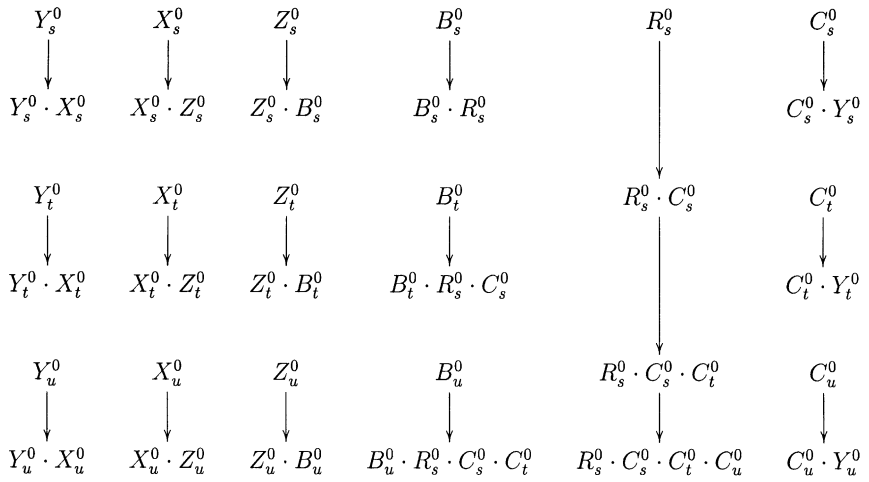

Fig. 4. Recovery.

Since $\mathscr{G}$ recovers at stage $v+1$, there are two possibilities. The first one is that the special component of $\mathscr{G}[v]$ is isomorphic to one of $B_s^0 \cdot R_s^0$, $B_t^0 \cdot R_s^0 \cdot C_s^0$, or $B_u^0 \cdot R_s^0 \cdot C_s^0 \cdot C_t^0$. In this case, $r_{v+1} = 1$.

The second possibility is that the special component of $\mathscr{G}[v]$ is isomorphic to $R_s^0 \cdot C_s^0 \cdot C_t^0 \cdot C_u^0$. In this case, the component of $\mathscr{G}[v]$ that extends C_u must be the one isomorphic to $C_u^0 \cdot Y_u^0$. From this it follows that the component of $\mathscr{G}[v]$ that extends Y_u must be the one isomorphic to $Y_u^0 \cdot X_u^0$. Proceeding in this fashion, we see that for each $w = s,t,u$, the component of $\mathscr{G}[v]$ that extends X_w is the one isomorphic to $X_w^0 \cdot Z_w^0$.

Note that in the previous argument it is crucial that no component of $\mathscr{A}^0$ other than the one that extends R_s^0 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 $\mathscr{G}$ is active, and $w+1$ is the first stage after stage $v+1$ at which $\mathscr{G}$ is active. Suppose further that $r_{s+1} = r_{t+1} = r_{u+1} = r_{v+1} = r_{w+1} = 0$.

Fig. 5 pictures what happens on either side of the construction. The key point to note here is that if $R_t^0 \cong R_t^1$ then R_w^0 extends R_t^0, R_w^1 extends R_t^1, and $R_w^0 \cong R_w^1$. This pattern would allow us to prove by induction that if r_s has a limit then each $\mathscr{A}^i$ has a unique infinite component S^i and $S^0 \cong S^1$.

In the full construction in the proof of Theorem 1.9, we of course had to satisfy (2.3′) for every computable directed graph. Let $\mathscr{G}_0, \mathscr{G}_1, \ldots$ be a standard enumeration of all partial computable directed graphs. In that construction, we defined the concepts of n-recovery stage, n-isomorphism recovery stage, $r_{n,s}$, and so forth in the same way as the corresponding concepts have been defined above, with $\mathscr{G}_n$ in place of $\mathscr{G}$. We

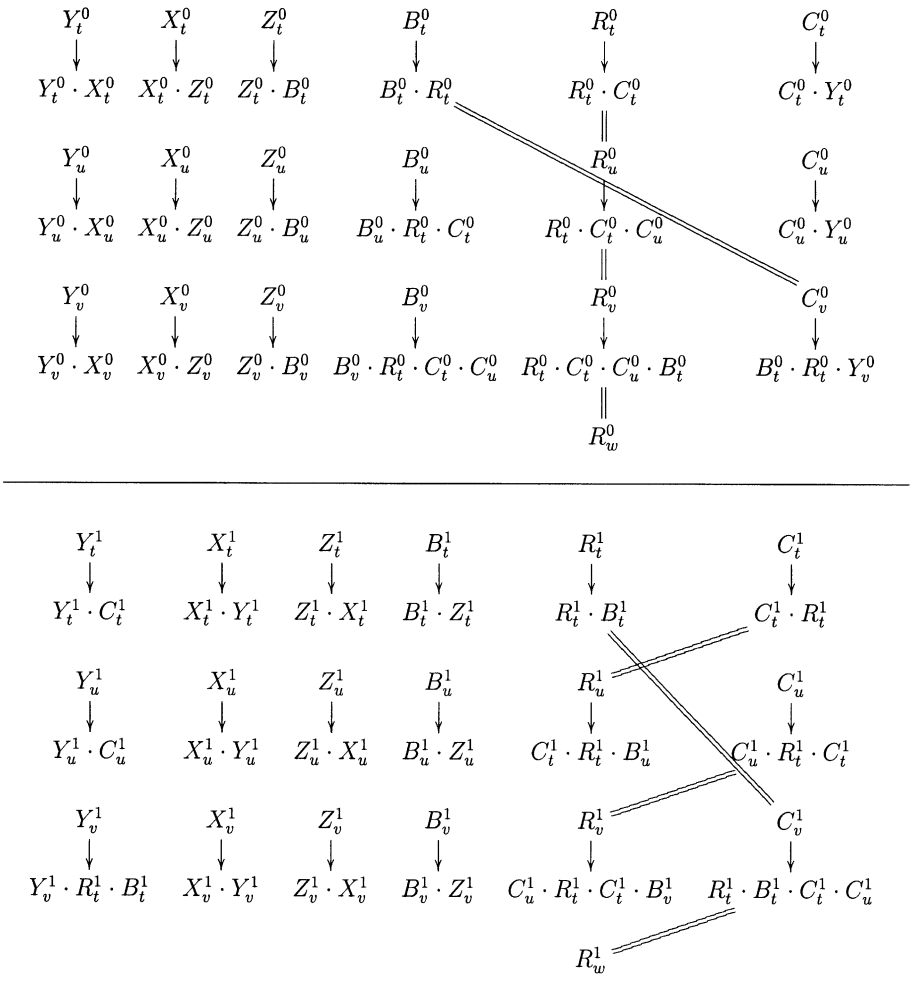

Fig. 5. Isomorphism recovery (top: $\mathscr{A}^0$ / bottom: $\mathscr{A}^1$).

also said that n was active at a given stage if copies of the special component of $\mathscr{G}_n$ participated in operations at that stage.

Remark. For the sake of definiteness, we make the following definition, although we will make no explicit use of it. A *partial computable directed graph* $\mathscr{G}$ consists of two 0, 1-valued partial computable functions Φ and Ψ, the former unary and the latter binary, such that if $\Phi(x)[s]\downarrow = \Phi(y)[s]\downarrow = 1$ then $\Psi(x, y)[s]\downarrow$. The graph $\mathscr{G}$ (resp. $\mathscr{G}[s]$) is the graph whose domain has characteristic function Φ ($\Phi[s]$) and whose edge relation has characteristic function Ψ ($\Psi[s]$).

We were able to satisfy (2.3′) for each $\mathscr{G}_n$ independently. In order to describe how this was done, we first need some notation to allow us to distinguish the components

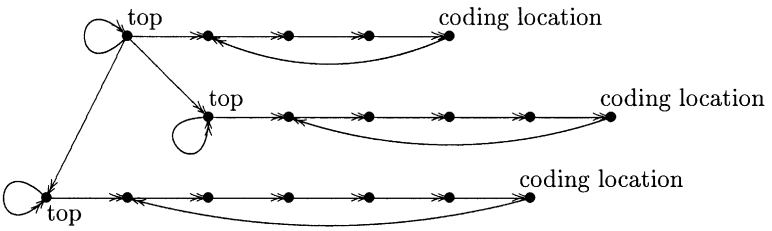

Fig. 6. The result of either of the operations $([3],[4])\cdot[2]$ or $[2]\cdot([3],[4])$.

that were used to satisfy (2.3′) for a particular $\mathscr{G}_n$. We will denote by $(\mathscr{A}^i)_n$ the subgraph of $\mathscr{A}_i$ consisting of those components used in the construction to satisfy (2.3′) for $\mathscr{G}_n$; that is, those components that act as Y, Z, B, S, and C components at some stage in the strategy for satisfying (2.3′) for $\mathscr{G}_n$ described above. The corresponding components of $\mathscr{G}_m$ for m possibly but not necessarily equal to n will be denoted by $(\mathscr{G}_m)_n$.

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

Definition 2.4. Let $\mathscr{G}$ 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 $\mathscr{G}$ 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 $\mathscr{G}$ to a new computable structure extending $\mathscr{G}$:

- The operation $(K_0, K_1, \ldots, K_n)\cdot L$ consists of performing the following steps, and otherwise leaving $\mathscr{G}$ unchanged. Create a new copy of $[x]$ using numbers not in the domain of $\mathscr{G}$. 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 $\mathscr{G}$ unchanged. For each $i \leqslant n$, create a new copy of $[y_i]$ using numbers not in the domain of $\mathscr{G}$. 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 Fig. 6, while the operation $L\cdot(K_0, K_1)$ consists of extending L to a copy of that same graph.

Definition 2.5. Let $\mathscr{G}$ be a computable structure in the language of directed graphs whose domain is co-infinite. We say that a component C of $\mathscr{G}$ 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 $\mathscr{G}$ 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 $\mathscr{G}$ to a new computable structure extending $\mathscr{G}$.

- The **L**-*operation*

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

consists of performing the following sequence of operations on $\mathscr{G}$:

$$(Y_0,\dots,Y_n)\cdot X,\ X\cdot(Z_0,\dots,Z_n),\ Z_0\cdot B_0,\dots,Z_n\cdot B_n,$$

$$B_0\cdot S_0,\dots,B_n\cdot S_n,\ S_0\cdot C_0,\dots,S_n\cdot C_n,\ C_0\cdot Y_0,\dots,C_n\cdot Y_n$$

- 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 performing the following sequence of operations on $\mathscr{G}$:

$$Y_0\cdot C_0,\dots,Y_n\cdot C_n,\ C_0\cdot S_0,\dots,C_n\cdot S_n,\ S_0\cdot B_0,\dots,S_n\cdot B_n,$$

$$B_0\cdot Z_0,\dots,B_n\cdot Z_n,\ (Z_0,\dots,Z_n)\cdot X,\ X\cdot(Y_0,\dots,Y_n)$$

Note that if $\mathscr{H}$ is the structure obtained by performing

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

on $\mathscr{G}$ and $\mathscr{H}'$ is the structure obtained by performing

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

on $\mathscr{G}$ then $\mathscr{H}\cong\mathscr{H}'$.

The idea now is that, at any given stage in the construction, there is a certain number of $\mathscr{G}_n$'s that need to have components Y_n^i, Z_n^i, B_n^i, S_n^i, and C_n^i of $(\mathscr{A}^i)_n$ participate in operations at that stage in order for the strategy for satisfying (2.3′) for $\mathscr{G}_n$ to proceed as described above. (The construction is organized in such a way that these components are distinct for different n's.) On the other hand, there is a unique component X^0 of $\mathscr{A}^0$ whose coding location will go into U^0, as well as a unique corresponding component X^1 of $\mathscr{A}^1$. Letting $n_0,\dots,n_k$ be all the n such that components of $(\mathscr{G}_n)_n$ need to participate in an operation at this stage, we can now perform

$$\mathbf{L}(Y_{n_0}^0,\dots,Y_{n_k}^0;X^0;Z_{n_0}^0,\dots,Z_{n_k}^0;B_{n_0}^0,S_{n_0}^0,C_{n_0}^0;\dots;B_{n_k}^0,S_{n_k}^0,C_{n_k}^0)$$

on $\mathscr{A}^0$ and perform

$$\mathbf{R}(Y_{n_0}^1,\dots,Y_{n_k}^1;X^1;Z_{n_0}^1,\dots,Z_{n_k}^1;B_{n_0}^1,S_{n_0}^1,C_{n_0}^1;\dots;B_{n_k}^1,S_{n_k}^1,C_{n_k}^1)$$

on $\mathscr{A}^1$. It is easy to check that the argument sketched out above still applies, and thus that, in this way, we can satisfy (2.3′) for all $\mathscr{G}_n$.

2.3. *Satisfying* (2.3)

As we have seen, the construction in the proof of Theorem 1.9 was an injury-free one in which the satisfaction of (2.3′) for a given $\mathscr{G}_n$ was handled by a single strategy, which worked with the components of $(\mathscr{A}^i)_n$ and acted independently from strategies for the satisfaction of (2.3′) for other $\mathscr{G}_m$. The trade-off was forgoing any control of $(\mathscr{G}_n)_m$ for $m \neq n$.

In order to satisfy (2.3), we need to control more of $\mathscr{G}_n$ than just $(\mathscr{G}_n)_n$. In order to illustrate how we do this, we consider the following sample situation. We have two graphs $\mathscr{G}_0$ and $\mathscr{G}_1$. We proceed with a construction like that described above, except that, in order for $\mathscr{G}_0$ to recover at stage $s+1$, we require not only that $\mathscr{G}_0[s]$ have the components that were necessary for 0-recovery, but also those that were necessary for 1-recovery, and we do not allow 1-recovery unless there is 0-recovery, which means that 1 is not active unless 0 is active. We claim that we will succeed in controlling $(\mathscr{G}_0)_1$ in the same sense that we controlled $(\mathscr{G}_0)_0$ before.

An example should be helpful here. Suppose that $s<t<u<v$ are such that $s+1$ is a first stage, $v+1$ is the next recovery stage after stage $s+1$, $r_{0,s+1}=r_{0,v+1}=0$, and $t+1$ and $u+1$ are the only two stages in the interval $(s+1, v+1)$ at which 0 is active. Suppose further that 1 is also active at stages $t+1$ and $u+1$. Notice that, since we do not allow 1 to be active unless 0 is active, $t+1$ and $u+1$ are the only two stages in the interval $(s+1, v+1)$ at which 1 is active. Figs. 7–9 picture what happens on the $\mathscr{A}^0$ side of the construction, depending on whether $r_{1,s+1}=0$ or $r_{1,s+1}=1$.

We are assuming the definition of recovery stage is such that the special component of $\mathscr{G}_0[s]$ is isomorphic to R^0_s, $\mathscr{G}_0[s]$ has a component $R_{1,s}$ isomorphic to $R^0_{1,s}$, and, for each $w=s,t,u$ and $i=0,1$, $\mathscr{G}_0[s]$ has components $Y_{i,w}$, X_w, $Z_{i,w}$, $B_{i,w}$, and $C_{i,w}$ isomorphic to $Y^0_{i,w}$, X^0_w, $Z^0_{i,w}$, $B^0_{i,w}$, and $C^0_{i,w}$, respectively.

Since $\mathscr{G}_0$ recovers at stage $v+1$ and $r_{0,v+1}=0$, the special component of $\mathscr{G}_0[v]$ is isomorphic to $R^0_{0,s}\cdot C^0_{0,s}\cdot C^0_{0,t}\cdot C^0_{0,u}$. So, arguing as before, we see that, for each $w=s,t,u$, the components of $\mathscr{G}_0[v]$ that extend $Y_{0,w}$, X_w, $Z_{0,w}$, $B_{0,w}$, and $C_{0,w}$ are isomorphic to the components of $\mathscr{A}^0_v$ that extend $Y^0_{0,w}$, X^0_w, $Z^0_{0,w}$, $B^0_{0,w}$, and $C^0_{0,w}$, respectively. In other words, all of $(\mathscr{G}_0)_0$ goes in the same direction as $(\mathscr{A}^0)_0$.

We wish to show that $(\mathscr{G}_0)_1$ also goes in the same direction as $(\mathscr{A}^0)_1$. Let $R'_{1,s}$ be the component of $\mathscr{G}_0[v]$ that extends $R_{1,s}$ and, for each $w=s,t,u$, let $Y'_{1,w}$, X'_w, $Z'_{1,w}$, $B'_{1,w}$, and $C'_{1,w}$ be the components of $\mathscr{G}_0[v]$ that extend $Y_{1,w}$, X_w, $Z_{1,w}$, $B_{1,w}$, and $C_{1,w}$, respectively.

In the $r_{1,s+1}=0$ case, we can argue as follows. As we have mentioned above, for each $w=s,t,u$, $X'_w\cong X^0_w\cdot(Z^0_{0,w},Z^0_{1,w})$, which implies that $Z'_{1,w}\cong Z^0_{1,w}\cdot B^0_{1,w}$. This in turn implies that $B'_{1,s}\cong B^0_{1,s}\cdot R^0_{1,s}$, $B'_{1,t}\cong B^0_{1,t}\cdot R^0_{1,s}\cdot C^0_{1,s}$, and $B'_{1,u}\cong B^0_{1,u}\cdot R^0_{1,s}\cdot C^0_{1,s}\cdot C^0_{1,t}$. So the only component of $\mathscr{A}^0_v$ left for $R'_{1,s}$ to be isomorphic to is $R^0_{1,s}\cdot C^0_{1,s}\cdot C^0_{1,t}\cdot C^0_{1,u}$. This implies that, for each $w=s,t,u$, $C'_{1,w}\cong C^0_{1,w}\cdot Y^0_{1,w}$, which in turn implies that $Y'_{1,w}\cong(Y^0_{0,w},Y^0_{1,w})\cdot X^0_w$. Thus, in this case, we see that $(\mathscr{G}_0)_1$ goes in the same direction as $(\mathscr{A}^0)_1$.

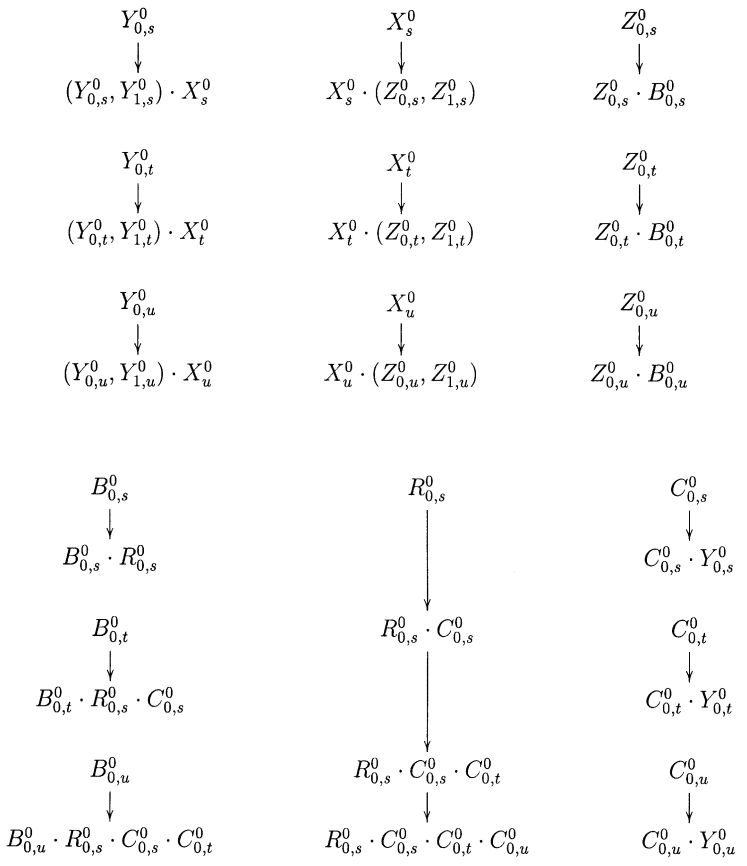

Fig. 7. Recovery in a two-strategy scenario: $(\mathscr{A}^0)_0$.

In the $r_{1,s+1} = 1$ case, the argument that $(\mathscr{G}_0)_1$ goes in the same direction as $(\mathscr{A}^0)_1$ is as follows. As before, for each $w = s, t, u$, $X'_w \cong X^0_w \cdot (Z^0_{0,w}, Z^0_{1,w})$, which implies that $Z'_{1,w} \cong Z^0_{1,w} \cdot B^0_{1,w}$. This implies that $B'_{1,u} \cong B^0_{1,u} \cdot B^0_{1,t} \cdot B^0_{1,s} \cdot R^0_{1,s}$, which implies that $B'_{1,t} \cong B^0_{1,t} \cdot B^0_{1,s} \cdot R^0_{1,s} \cdot C^0_{1,u}$, which implies that $B'_{1,s} \cong B^0_{1,s} \cdot R^0_{1,s} \cdot C^0_{1,t}$, which implies that $R'_{1,s} \cong R^0_{1,s} \cdot C^0_{1,s}$. Now, for each $w = s, t, u$, we have $C'_{1,w} \cong C^0_{1,w} \cdot Y^0_{1,w}$, which implies that $Y'_{1,w} \cong (Y^0_{0,w}, Y^0_{1,w}) \cdot X^0_w$. Thus, in this case also, $(\mathscr{G}_0)_1$ goes in the same direction as $(\mathscr{A}^0)_1$.

In either case, we have the same kind of control over $(\mathscr{G}_0)_1$ as we have over $(\mathscr{G}_0)_0$. Now assume that $\mathscr{G}_0 \cong \mathscr{A}^0$ and $\lim_s r_{0,s} = 0$. We claim that, if there are no other elements to the construction, so that from some stage s on all of $\mathscr{G}_0$ goes in the same direction as $\mathscr{A}^0$, then the unique isomorphism $f : \mathscr{A}^0 \to \mathscr{G}_0$ is computable. (Recall that we are assuming that $\mathscr{A}^0$ is rigid.) Indeed, the following is an effective procedure for computing $f(x)$ given $x \in \mathscr{A}^0$. Find the least stage $t \geqslant s$ such that x is contained in a component K of $\mathscr{A}^0_t$ and there is an isomorphism g from K to some component L

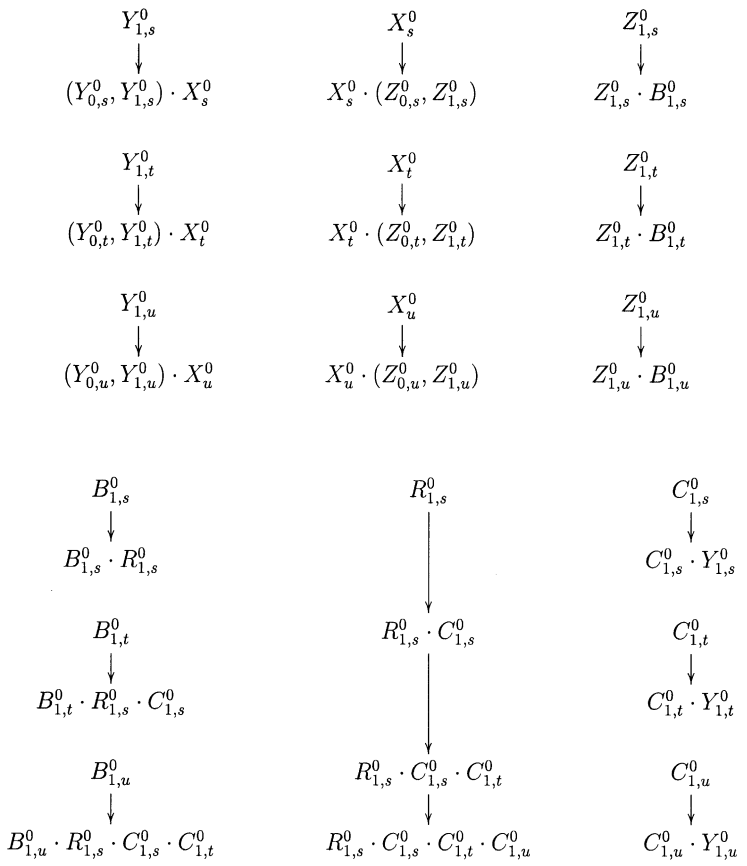

Fig. 8. Recovery in a two-strategy scenario: $(\mathscr{A}^0)_1$ in case $r_{1,s+1} = 0$.

of $\mathscr{G}_0[t]$. Such a stage must exist by the definition of 0-recovery, and, since all of $\mathscr{G}_0$ goes in the same direction as $\mathscr{A}^0$ from stage s on, $f(x) = g(x)$.

Of course, the strategy for $\mathscr{G}_0$ that we have just described works at the expense of the corresponding strategy for $\mathscr{G}_1$. Indeed, if $\mathscr{G}_0$ does not recover infinitely often then $\mathscr{G}_1$ is not allowed to recover infinitely often, even though it might be the case that $\mathscr{G}_1 \cong \mathscr{A}^0$. We solve this problem in the standard way, by having multiple strategies for satisfying (2.3) for a given $\mathscr{G}_n$ and organizing these on a tree. (The reader unfamiliar with the technique of organizing priority constructions on a tree should consult [24].) More specifically, for each finite binary string σ, there will be a strategy for satisfying (2.3) for $\mathscr{G}_{|\sigma|}$, where $|\sigma|$ is the length of σ. The string σ represents a guess about which $\mathscr{G}_m$, $m < |\sigma|$, recover infinitely often, with $\sigma(m) = 0$ representing a guess that $\mathscr{G}_m$ recovers infinitely often and $\sigma(m) = 1$ representing a guess that it does not.

For each σ of length n, $\mathscr{G}_n$ will have a σ-special component. We will say that σ is active whenever this component participates in an operation, and will define the

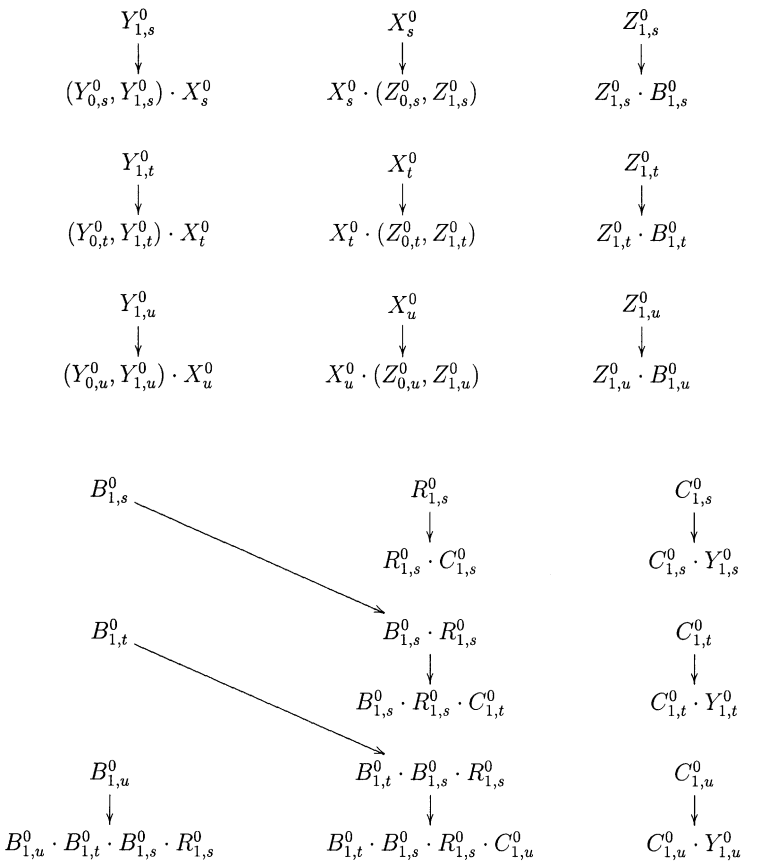

Fig. 9. Recovery in a two-strategy scenario: $(\mathscr{A}^0)_1$ in case $r_{1,s+1} = 1$.

concepts of σ-recovery, σ-isomorphism recovery, and so forth. We will write $\sigma^\frown i$ to mean the concatenation of σ with the string of length 1 whose only element is i.

For all τ such that $\sigma^\frown 0 \subseteq \tau$, τ will not be accessible except at σ-recovery stages, which means that there will not be τ-recovery at a given stage unless there is also σ-recovery. As in the two-strategy scenario above, the requirements for σ-recovery will be such that the components that must be provided by $\mathscr{G}_{|\sigma|}$ for it to σ-recover include all the components that must be provided by $\mathscr{G}_{|\tau|}$ for it to τ-recover. (Since there are infinitely many τ extending $\sigma^\frown 0$ and we can only require $\mathscr{G}_{|\sigma|}$ to provide finitely many components for each σ-recovery, we will not allow such a τ to recover until σ has recovered $|\tau| + 1$ many times.) In this way, if σ is on the true path of the construction (which will be defined, as usual, as the leftmost path visited infinitely often) and $\mathscr{G}_{|\sigma|} \cong \mathscr{A}^0$ then we will be able to control not only $(\mathscr{G}_{|\sigma|})_\sigma$, but also $(\mathscr{G}_{|\sigma|})_\tau$ for all τ such that $\sigma^\frown 0 \subseteq \tau$, by a similar argument to that in the two-strategy scenario.

It is important to note that σ might be active at stages at which it is not accessible. This is because, as in the simpler construction described above, in order for $\mathscr{G}_{|\sigma|}$ to

σ-recover, we will require that it provide enough components to allow σ to be active whenever a number less than the number of times $\mathscr{G}_{|\sigma|}$ has σ-recovered enters A. Whenever such a number does enter A, we will allow σ to be active, unless $\mathscr{G}_{|\sigma|}$ has not σ-recovered since the last time σ was initialized (that is, the last time the construction moved to the left of σ).

The reason we require $\mathscr{G}_{|\sigma|}$ to σ-recover at least once following an initialization before σ can be active again is that the components that can be used by the strategy corresponding to σ (including the σ-special component) will change each time σ is initialized (more on this below). This restriction will not hamper the strategies on the true path, since these will be initialized only finitely often.

In the following discussion, we will denote by (k) the component of $\mathscr{A}^i$ that extends the unique copy of $[k]$ in $\mathscr{A}^i_0$, and by $\langle A^i\rangle_\sigma$ we will mean the union of the components of $\mathscr{A}^i$ that might potentially be used by the strategy for satisfying (2.3) for $\mathscr{G}_{|\sigma|}$ corresponding to σ; once we give the formal details of the construction, it will be clear which components these are. (As was the case with the corresponding notations in [18], $(A^i)_\sigma$ and $(\mathscr{G}_n)_\sigma$, $n\in\omega$, will refer to the union of those components that are actually used by the strategy corresponding to σ.) By $\langle A^i\rangle$ we will mean the union of the components of $\mathscr{A}^i$ of the form $(6k)$, $k\in\omega$. (These are the components that might not be in $\langle\mathscr{A}^i\rangle_\sigma$ for any σ.)

Fix σ on the true path such that $\mathscr{G}_{|\sigma|}\cong\mathscr{A}^0$. These conditions on σ will imply that, for all $\tau\subsetneq\sigma$, τ recovers infinitely often if and only if $\tau^\frown 0\subseteq\sigma$. They will also imply that $\sigma^\frown 0$ is on the true path, so that σ recovers infinitely often, and that $\lim_s r_{\sigma,s}$ exists (where $r_{\sigma,s}$ will be defined analogously to $r_{n,s}$). Let $i=\lim_s r_{\sigma,s}$ and let f be the unique isomorphism from $\mathscr{A}^i$ to $\mathscr{G}_{|\sigma|}$. As discussed above, we will be able to compute both $f\upharpoonright\langle\mathscr{A}^i\rangle_\sigma$ and $f\upharpoonright\bigcup_{\tau\supseteq\sigma^\frown 0}\langle\mathscr{A}^i\rangle_\tau$.

Of course, this leaves the problem of uniformly computing $f\upharpoonright\langle\mathscr{A}^i\rangle_\tau$ for other τ, as well as $f\upharpoonright\langle A^i\rangle$. Our strategy for computing f will be to break up the domain of $\mathscr{A}^i$ into finitely many c.e. sets and show that the restriction of f to each of these sets is computable. Most of the cases will be handled by making use of the fact that, for a c.e. union T of finite components of $\mathscr{A}^i$, if each component of T participates in operations only finitely often and there is a computable bound on the last stage (if any) at which each component of T participates in an operation then $f\upharpoonright T$ is computable. This is because if a component K of T does not participate in operations after stage s then K is a component of $\mathscr{A}^i_s$, and hence the unique embedding from K into $\mathscr{G}_{|\sigma|}$ can be found effectively. (Recall that we are assuming that our construction is such that no component of $\mathscr{A}^i$ is embeddable in another component of $\mathscr{A}^i$.)

We begin by looking at $\langle A^i\rangle$. As discussed above, we will have a computable bound $h(k)$ such that if $(6k)$ has not participated in an operation by stage $h(k)$ then, whenever it does participate in an operation, σ is active. Let T_0 be the union of those components $(6k)$ of $\langle A^i\rangle$ that do not participate in an operation by stage $h(k)$. Then $f\upharpoonright T_0$ will be computable for the same reason as $f\upharpoonright\langle\mathscr{A}^i\rangle_\sigma$. On the other hand, since no component of $\langle A^i\rangle$ will participate in operations more than once, $f\upharpoonright(\langle A^i\rangle - T_0)$ will be computable because $h(k)$ will be a computable bound on the last stage at

which a component $(6k)$ of $\langle A^i\rangle - T_0$ participates in an operation. Thus $f\restriction\langle A^i\rangle$ will be computable.

Now let T_1 be the union of all $\langle\mathscr{A}^i\rangle_\tau$ such that τ is to the left of σ. By the definition of the true path, only finitely many components of T_1 will ever participate in operations, and those that do, will do so only finitely often. Thus, there will exist a computable bound on the last stage at which each component of T_1 participates in an operation, and hence $f\restriction T_1$ will be computable.

Let T_2 be the union of all $\langle\mathscr{A}^i\rangle_\tau$ such that $\tau^\frown 1\subseteq\sigma$. The fact that there are only finitely many τ-recovery stages will imply that only finitely many components of T_2 participate in operations, and those that do, do so only finitely often. Thus there will exist a computable bound on the last stage at which each component of T_2 participates in an operation, and hence $f\restriction T_2$ will be computable.

Let T_3 be the union of all $\langle\mathscr{A}^i\rangle_\tau$ such that τ is to the right of $\sigma^\frown 0$. Every time the construction moves to the left of τ, we will guarantee, as part of the initialization process, that a certain set of components of $\langle\mathscr{A}^i\rangle_\tau$ will never again participate in an operation, in such a way that if the construction moves to the left of τ infinitely often then every component of $\langle\mathscr{A}^i\rangle_\tau$ will eventually be guaranteed never again to participate in an operation. Since $\sigma^\frown 0$ is on the true path, this will mean that there exists a computable bound on the last stage at which each component of T_3 participates in an operation, and hence $f\restriction T_3$ will be computable.

We are left with the case of $\langle\mathscr{A}^i\rangle_\tau$ such that $\tau^\frown 0\subseteq\sigma$. We will show that, for each such τ, if $r_{\tau,s}$ has a limit then $\langle\mathscr{A}^i\rangle_\tau$ has a unique infinite component S^i_τ, while if $r_{\tau,s}$ does not have a limit then all components of $\langle\mathscr{A}^i\rangle_\tau$ are finite. Let T_4 be the union of the S^i_τ, $\tau^\frown 0\subseteq\sigma$, $r_{\tau,s}$ has a limit. Given a copy K of $[m]$ contained in a component C of T_4 with top x, we will be able to find effectively the unique copy L of $[m]$ in the component of $\mathscr{G}_{|\sigma|}$ with top $f(x)$, and f will extend the unique isomorphism from K to L. Since T_4 has only finitely many components, this will mean that $f\restriction T_4$ is computable.

Finally, let T_5 be the union of all finite components of $\langle\mathscr{A}^i\rangle_\tau$, $\tau^\frown 0\subseteq\sigma$. Examining the construction in the proof of Theorem 1.9, we see that, given an n such that $\mathscr{G}_n\cong\mathscr{A}^0$, once a finite component K of $(\mathscr{A}^i)_n$ participates in an operation at a stage s, we can effectively find a stage t such that K does not participate in an operation after stage t. Indeed, we can take t to be the first stage after stage s such that, for some $u<t$, K does not participate in an operation in the interval $[u,t]$ and there is an n-isomorphism recovery stage in $[u,t]$.

The analogous situation will hold here, but this will not quite be enough to show that $f\restriction T_5$ is computable. We will also need an effective procedure that, for each component K of T_5, gives us a stage s such that if K has not participated in an operation by stage s then it will not participate in an operation after stage s. In order to do this, every time τ recovers, we will guarantee that a certain set of components of $\langle\mathscr{A}^i\rangle_\tau$ that have not yet participated in an operation will never participate in an operation, in such a way that if τ recovers infinitely often then every singleton component of $\langle\mathscr{A}^i\rangle_\tau$ will eventually be guaranteed never to participate in an operation. (That is, we will add

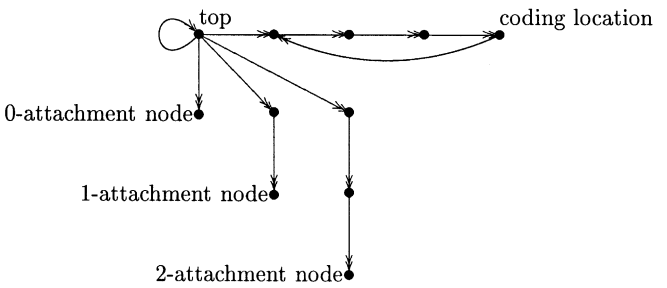

Fig. 10. $[2]^+$.

an extra condition to the definition of τ-recovery to ensure that, for each singleton component that had been available at the last τ-recovery stage to be used for the sake of the strategy corresponding to τ and that has not yet been used, there is a new component that can be used in its place. A similar procedure was employed in [20].) Thus $f \upharpoonright T_5$ will be computable.

2.4. Formal definitions and conventions

For the sake of satisfying (2.5), we need a new kind of building block, whose use will be made clear shortly. (Basically, if $\mathcal{G}$ is a c.e. graph with computable equality relation and K and L are different components of $\mathcal{G}[s]$, $s \in \omega$, then it cannot be the case that K and L are both extended by a component of the form $K \cdot L$ in $\mathcal{G}$. However, K and L might both be extended by the same component of $\mathcal{G}$ if this component is of the form $K \cdot (L)$, for example, since the fact that there is no edge from the top of K to the top of L in $\mathcal{G}[s]$ does not mean that the same is true in $\mathcal{G}$. We will avoid this possibility by only performing operations of the form $K \cdot (L_0, \ldots, L_k)$ when K is of the form $[n]^+$, $n \in \omega$, as defined below.)

Definition 2.6. The directed graph $[n]^+$ consists of the following nodes and edges:
1. A copy of $[n]$ with top x.
2. For each $i \leqslant n$, $i+1$ many nodes $x_{i,0}, \ldots, x_{i,i}$, with an edge from x to $x_{i,0}$ and, for each $j < i$, an edge from $x_{i,j}$ to $x_{i,j+1}$. We call $x_{i,i}$ the *i-attachment node* of $[n]^+$.

Fig. 10 shows $[2]^+$ as an example.

We also need a new version of Definition 2.4.

Definition 2.7. 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 $[k_0], [k_1], \ldots, [k_n]$ and $[l]^+$, respectively, where $k_0, k_1, \ldots, k_n, l \in \omega$ and $n \leqslant l$. 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 creating a new copy of $[l]^+$, using the top of K_i as the i-attachment node for $i \leqslant n$ and numbers not in the domain of $\mathscr{G}$ as the other nodes, and otherwise leaving $\mathscr{G}$ unchanged.
- The operation $L \cdot (K_0, K_1, \ldots, K_n)$ consists of creating a new copy of $[k_i]$ for each $i \leqslant n$, using the i-attachment node of L as the top and numbers not in the domain of $\mathscr{G}$ as the other nodes, and otherwise leaving $\mathscr{G}$ unchanged.

We define the **L**- and **R**-operations as in Definition 2.5, except that we now require that X be of the form $[k]^+$, $k \in \omega$.

Fix a computable one-to-one function from $2^{<\omega}$ onto $\omega - \{0\}$ and let $\ulcorner \sigma \urcorner$ denote the image under this function of the string σ.

Definition 2.8. Let $\mathscr{G}$ be a directed graph. We denote by $(\mathscr{G})_\sigma$ the subgraph of $\mathscr{G}$ consisting of those components C of $\mathscr{G}$ that satisfy both of the following conditions:
1. C is not isomorphic to $[x]$ or $[x]^+$ for any $x \in \omega$.
2. C contains a copy of $[6\langle \ulcorner \sigma \urcorner, j\rangle + 3]$, $j \in \omega$, or a copy of $[6\langle \ulcorner \sigma \urcorner, j, k\rangle + l]$, $j, k \in \omega$, $l \in \{1, 2, 4, 5\}$.

Define $(\mathscr{G})_{\supseteq \sigma} = \bigcup_{\tau \supseteq \sigma} (\mathscr{G})_\tau$.

For $\sigma, \tau \in 2^{\leqslant \omega}$, $\sigma \leqslant_{\mathrm{L}} \tau$ means that either $\sigma \subseteq \tau$ or there exists an $n < |\sigma|, |\tau|$ such that $\sigma(m) = \tau(m)$ for all $m < n$, $\sigma(n) = 0$, and $\tau(n) = 1$. If $\sigma \leqslant_{\mathrm{L}} \tau$ and $\sigma \not\subseteq \tau$ then we say that σ is *to the left* of τ and that τ is *to the right* of σ.

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

Let $t \geqslant s$. We say that a component L of $\mathscr{A}^i_t$ or $\mathscr{A}^i$ (resp. $\mathscr{G}_n[t]$ or $\mathscr{G}_n$) *extends* a component K of $\mathscr{A}^i_s$ ($\mathscr{G}_n[s]$) 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 saying 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 $\mathscr{A}^i_s$ ($\mathscr{G}_n[s]$).

It will be the case that if K and L are distinct components of $\mathscr{A}^0_s$ 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 $\mathscr{A}^0$. Thus, since we are not interested in $\mathscr{G}_n$ unless it is isomorphic to $\mathscr{A}^0$, we may assume without loss of generality that, for each $n, s \in \omega$, there is an embedding of $\mathscr{G}_n[s]$ into $\mathscr{A}^0_s$ such that if K and L are distinct components of $\mathscr{G}_n[s]$ 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 $\mathscr{A}^0_s$.

Let k be the number of times σ has been initialized (defined below) before stage t. Suppose there is a least stage $s \leqslant t$ such that $\mathscr{G}_{|\sigma|}[s]$ has a component K isomorphic to $[6\langle \ulcorner \sigma \urcorner, k\rangle + 3]$. We call the component of $\mathscr{G}_{|\sigma|}[t]$ that extends K the *σ-special component of* $\mathscr{G}_{|\sigma|}[t]$. If σ is initialized only finitely often, say k many times, and there is a least

stage s such that $\mathscr{G}_{|\sigma|}[s]$ has a component K isomorphic to $[6\langle\ulcorner\sigma\urcorner,k\rangle+3]$ then we call the component of $\mathscr{G}_{|\sigma|}$ that extends K the *σ-special component of $\mathscr{G}_{|\sigma|}$*.

2.5. The construction

We now proceed with the construction of $\mathscr{A}^0$, $\mathscr{A}^1$, U^0, and U^1. It will be easy to check as we go along that the following are properties of the construction:

1. For each $s\in\omega$, $\mathscr{A}^0_s\cong\mathscr{A}^1_s$ and no component of $\mathscr{A}^i_s$ is embeddable in another component of $\mathscr{A}^i_s$.
2. Let $t<s$. No component of $\mathscr{A}^i_t$ isomorphic to one of $[6a_s]^+$ or $[6\langle j,a_s,k\rangle+l]$, $j,k\in\omega$, $l\in\{1,2,4,5\}$, participates in an operation at stage $t+1$.

Stage 0: Let $\mathscr{A}^0_0$ and $\mathscr{A}^1_0$ be computable structures with co-infinite domains, each consisting of one copy of $[6k+l]$ and one of $[6k]^+$ for each $k\in\omega$ and $0<l<6$. For each $\sigma\in 2^{<\omega}$, let $r_{\sigma,0}=0$.

Stage $s+1$: For $\sigma\in 2^{<\omega}$, let $recov(\sigma,s)$ be the number of σ-recovery stages before stage $s+1$, let $init(\sigma,s)$ be the number of times σ has been initialized before stage $s+1$, and let $c(\sigma,s)=\max(recov(\sigma,s),init(\sigma,s))$.

Define the string $\sigma[s+1]\in 2^{[0,s]}$ by recursion as follows, beginning with $n=0$. Let $\sigma=\sigma[s+1]\upharpoonright n$. Say that $s+1$ is a *σ-recovery stage* if all of the following conditions hold:

1. Every τ such that $\tau^\frown 0\subseteq\sigma$ has recovered at least $|\sigma|+1$ many times.
2. $\mathscr{G}_n[s]$ has a σ-special component isomorphic to some component of $\mathscr{A}^0_s$.
3. If $\tau\supseteq\sigma^\frown 0$ has not yet recovered since the last time it was initialized and $|\tau|\leqslant recov(\sigma,s)$ then $\mathscr{G}_n[s]$ has a component isomorphic to $[6\langle\ulcorner\tau\urcorner,init(\tau,s)\rangle+3]$.
4. $(\mathscr{G}_n[s])_\sigma\cong(\mathscr{A}^0_s)_\sigma$.
5. $(\mathscr{G}_n[s])_{\supseteq\sigma^\frown 0}\cong(\mathscr{A}^0_s)_{\supseteq\sigma^\frown 0}$.
6. Let τ be such that either $\tau=\sigma$ or both $\tau\supseteq\sigma^\frown 0$ and $|\tau|\leqslant recov(\sigma,s)$. Let $j\notin A[s]$ be less than or equal to $recov(\tau,s)$. There is a component of $\mathscr{G}_n[s]$ isomorphic to $[6j]^+$ and, for each $l\in\{1,2,4,5\}$, there is a component of $\mathscr{G}_n[s]$ isomorphic to $[6\langle\ulcorner\tau\urcorner,j,c(\tau,s)\rangle+l]$.

If $s+1$ is a σ-recovery stage then let $\sigma[s+1](n)=0$. Otherwise, let $\sigma[s+1](n)=1$.

For each σ such that $s+1$ is a σ-recovery stage, proceed as follows. For $i=0,1$, let $S^i_{\sigma,s}$ be the component of $\mathscr{A}^i_s$ that is isomorphic to the σ-special component of $\mathscr{G}_{|\sigma|}[s]$. If $s+1$ is either the first σ-recovery stage ever or the first σ-recovery stage since the last time σ was initialized then let $r_{\sigma,s+1}=0$. Otherwise, proceed as follows. Let $i=r_{\sigma,s}$ and let $t+1$ be the last σ-recovery stage before stage $s+1$. If $S^i_{\sigma,s}$ extends $S^i_{\sigma,t}$ then let $r_{\sigma,s+1}=i$, and otherwise let $r_{\sigma,s+1}=1-i$.

For each $\sigma\in 2^{<\omega}$ such that $s+1$ is not a σ-recovery stage, let $r_{\sigma,s+1}=r_{\sigma,s}$.

Declare each σ to the right of $\sigma[s+1]$ to have been *initialized*. For each $\sigma\leqslant_L\sigma[s+1]$, if there has been a σ-recovery stage since the last time σ was initialized, $a_s\geqslant|\sigma|$, and a_s is less than the number of σ-recovery stages less than or equal to $s+1$ then say that σ is *active* at stage $s+1$.

For $i=0,1$, let X^i_s be the component of $\mathscr{A}^i_s$ isomorphic to $[6a_s]^+$.

Let $\sigma_0,\dots,\sigma_m$ be all the strings that are active at stage $s+1$. For $i=0,1$ and $j\leqslant m$, let $Y^i_{\sigma_j,s}$ and $Z^i_{\sigma_j,s}$ be the components of $\mathscr{A}^i_s$ isomorphic to $[6\langle{}^\ulcorner\sigma_j{}^\urcorner,a_s,c(\sigma_j,s)\rangle+1]$ and $[6\langle{}^\ulcorner\sigma_j{}^\urcorner,a_s,c(\sigma_j,s)\rangle+2]$, respectively.

For each $j\leqslant m$, let $t_j+1\leqslant s+1$ be the last σ_j-recovery stage. We say that $s+1$ is a *σ_j-first stage* if it is the first stage after stage t_j at which σ_j is active. We say that $s+1$ is a *σ_j-change stage* if it is a σ_j-first stage and one of the following holds: t_j+1 was the first σ_j-recovery stage ever, t_j+1 was the first σ_j-recovery stage since the last time σ_j was initialized, or $r_{\sigma_j,t_j+1}\neq r_{\sigma_j,t_j}$. We say that $s+1$ is a *σ_j-isomorphism recovery stage* if it is a σ_j-first stage but not a σ_j-change stage and one of the following conditions holds:

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

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

1. $s+1$ is a σ_j-isomorphism recovery stage. If the first condition in the definition of σ_j-isomorphism recovery stage holds then let $t+1$ be the last σ_j-first stage, and otherwise let $t+1$ be the first stage after the last σ_j-isomorphism recovery stage at which σ_j was active. There are two subcases.
   (a) If $r_{\sigma_j,s+1}=0$ then let $C^0_{\sigma_j,s}$ be the component of $\mathscr{A}^0_s$ that extends $B^0_{\sigma_j,t}$ and let $C^1_{\sigma_j,s}$ be its isomorphic image in $\mathscr{A}^1_s$. For $i=0,1$, let $B^i_{\sigma_j,s}$ be the component of $\mathscr{A}^i_s$ isomorphic to $[6\langle\sigma_j,a_s,c(\sigma_j,s)\rangle+4]$.
   (b) If $r_{\sigma_j,s+1}=1$ then let $B^1_{\sigma_j,s}$ be the component of $\mathscr{A}^1_s$ that extends $C^1_{\sigma_j,t}$ and let $B^0_{\sigma_j,s}$ be its isomorphic image in $\mathscr{A}^0_s$. For $i=0,1$, let $C^i_{\sigma_j,s}$ be the component of $\mathscr{A}^i_s$ isomorphic to $[6\langle\sigma_j,a_s,c(\sigma_j,s)\rangle+5]$.
2. $s+1$ is not a σ_j-isomorphism recovery stage. For $i=0,1$, let $B^i_{\sigma_j,s}$ be the component of $\mathscr{A}^i_s$ isomorphic to $[6\langle\sigma_j,a_s,c(\sigma_j,s)\rangle+4]$ and let $C^i_{\sigma_j,s}$ be the component of $\mathscr{A}^i_s$ isomorphic to $[6\langle\sigma_j,a_s,c(\sigma_j,s)\rangle+5]$.

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

Now perform

$$\mathbf{L}(Y^0_{\sigma_0,s},\dots,Y^0_{\sigma_m,s};X^0_s;Z^0_{\sigma_0,s},\dots,Z^0_{\sigma_m,s};B^0_{\sigma_0,s},R^0_{\sigma_0,s},C^0_{\sigma_0,s};$$
$$B^0_{\sigma_1,s},R^0_{\sigma_1,s},C^0_{\sigma_1,s};\dots;B^0_{\sigma_m,s},R^0_{\sigma_m,s},C^0_{\sigma_m,s})$$

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

$$\mathbf{R}(Y^1_{\sigma_0,s},\dots,Y^1_{\sigma_m,s};X^1_s;Z^1_{\sigma_0,s},\dots,Z^1_{\sigma_m,s};B^1_{\sigma_0,s},R^1_{\sigma_0,s},C^1_{\sigma_0,s};$$
$$B^1_{\sigma_1,s},R^1_{\sigma_1,s},C^1_{\sigma_1,s};\dots;B^1_{\sigma_m,s},R^1_{\sigma_m,s},C^1_{\sigma_m,s})$$

on $\mathscr{A}^1_s$ to get $\mathscr{A}^1_{s+1}$. (If no σ is active at stage $s+1$ then, for $j=0,1$, let Y^j_s, Z^j_s, B^j_s, R^j_s, and C^j_s be the components of $\mathscr{A}^j_s$ isomorphic to $[6\langle 0,a_s,s\rangle+1]$, $[6\langle 0,a_s,s\rangle+2]$,

$[6\langle 0, a_s, s\rangle+4]$, $[6\langle 0, s\rangle + 3]$, and $[6\langle 0, a_s, s\rangle+5]$, respectively. Perform $\mathbf{L}(Y_s^0; X_s^0; Z_s^0; B_s^0; R_s^0; C_s^0)$ on $\mathscr{A}_s^0$ to get $\mathscr{A}_{s+1}^0$ and perform $\mathbf{R}(Y_s^1; X_s^1; Z_s^1; B_s^1; R_s^1; C_s^1)$ on $\mathscr{A}_s^1$ to get $\mathscr{A}_{s+1}^1$.)

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

This completes the construction. Let $\mathscr{A}^0 = \bigcup_{s\in\omega} \mathscr{A}_s^0$ and $\mathscr{A}^1 = \bigcup_{s\in\omega} \mathscr{A}_s^1$. Define the *true path TP* of the construction to be the leftmost path of 2^ω such that there are infinitely many stages s with $\sigma[s] \in TP$.

2.6. Verification

Since, for each $s \in \omega$ and $i = 0, 1$, all numbers in $\mathscr{A}_{s+1}^i - \mathscr{A}_s^i$ are greater than s, $\mathscr{A}^0$ and $\mathscr{A}^1$ are computable. We will now argue that properties (2.1)–(2.5) hold. Theorem 1.10 will then follow immediately.

Properties (2.2) and (2.5) are easy to establish, so we deal with them first.

Lemma 2.9. *$U^0 \equiv_{\mathrm{m}} A$ and U^1 is computable.*

Proof. The numbers in U^0 are all coding locations of components of $\mathscr{A}_0^0$ of the form $[6j]$, $j \in \omega$, and the coding location of the copy of $[6j]$ in $\mathscr{A}_0^0$ 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 $\mathscr{A}_0^0$ 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 $\mathscr{A}_s^1$, and hence is greater than s. Thus U^1 is computable. □

Lemma 2.10. *If $\mathscr{G}$ is a c.e. presentation of $\mathscr{A}^0$ with computable equality relation then $\mathscr{G}$ is computable.*

Proof. Since the equality relation in $\mathscr{G}$ is computable, we can assume without loss of generality that the enumeration of $\mathscr{G}$ is such that, for all $s, w, x, y, z \in \omega$, if $w, x, y, z \in |\mathscr{G}[s]|$ and the pairs (w, x) and (y, z) satisfy the equality relation in $\mathscr{G}$ then there is an edge from w to y in $\mathscr{G}[s]$ if and only if there is an edge from x to z in $\mathscr{G}[s]$. Let $x_0, x_1 \in \mathscr{G}$. Wait until a stage s in the enumeration of $\mathscr{G}$ such that, for each $i = 0, 1$, x_i is in a copy of either $[n_i]$ for some $n_i \not\equiv 0 \bmod 6$ or $[n_i]^+$ for some $n_i \equiv 0 \bmod 6$. It is easy to check from the definition of $\mathscr{A}^0$ that, for each $i = 0, 1$, there is an edge from x_i to x_{1-i} if and only if there already is such an edge at stage s. □

In showing that (2.1), (2.3), and (2.4) 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. Figs. 4 and 5 should be helpful here.

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

Lemma 2.11. *Let $\mathcal{G} \cong \mathcal{A}^0$ 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.*

Lemma 2.12. *Let K and L be distinct components of $\mathcal{A}^i_s$ 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 $\mathcal{A}^i$.*

Lemmas 2.11 and 2.12 will be used without explicit mention of several times below.

Lemma 2.13. *Each component of $\mathcal{A}^i$ is rigid and contains at most one copy of $[k]$ for each $k \in \omega$.*

Lemma 2.14. *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 $\mathcal{A}^i_s$. 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 $\mathcal{A}^{1-i}_s$.*

Lemma 2.15. *A component of $\mathcal{A}^i$ is infinite if and only if it participates in operations infinitely often.*

Lemma 2.16. *Let $k, j \in \omega$ and $\sigma \in 2^{<\omega}$. Any component of $\mathcal{A}^i$ containing a copy of $[6k]$ or $[6\langle \ulcorner \sigma \urcorner, j, k\rangle + l]$, $l \in \{1,2\}$, can participate in an operation at most once. Any component of $\mathcal{A}^i$ containing a copy of $[6\langle \ulcorner \sigma \urcorner, j\rangle + 3]$ or $[6\langle \ulcorner \sigma \urcorner, j, k\rangle + l]$, $l \in \{1,2,4,5\}$, can participate in operations only at stages at which σ is active.*

Lemma 2.17. *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 $[6\langle n, a_s, k\rangle + 1]$ for some $n, k \in \omega$, in which case $x \notin U^i$, or K contains a copy of $[6\langle n, a_s, k\rangle + 2]$ for some $n, k \in \omega$, in which case $x \in U^i$.*

Lemma 2.18. *If a component K of $(\mathcal{A}^i)_\sigma$ participates in operations at stages $s < t+1$ but does not participate in an operation at any stage in the interval $(s,t]$ then there are no σ-change stages or σ-isomorphism recovery stages in $(s,t]$.*

Proof. Let w be the last σ-first stage before stage $t+1$. If K extends $R^i_{\sigma,t}$ then it is easy to check that K must have participated in an operation in the interval $[w,t]$, which means that $w \leqslant s$. Since every σ-change or σ-isomorphism recovery stage is a σ-first stage, in this case we are done.

Otherwise, t is an isomorphism-recovery stage and either $r_{\sigma,t+1} = 0$ and K extends $C^i_{\sigma,t}$ or $r_{\sigma,t+1} = 1$ and K extends $B^i_{\sigma,t}$. Suppose for a contradiction that there is at least one σ-change stage or σ-isomorphism recovery stage in $(s,t]$, and let u be maximal among such stages.

If u is a σ-change stage then it must be the last σ-first stage before stage $t+1$, since the next σ-first stage after a σ-change stage is either a σ-change stage or a

σ-isomorphism recovery stage. In this case, by the way components that participate in an operation at an isomorphism-recovery stage are chosen, K participated in an operation at stage u.

If u is a σ-isomorphism recovery stage then there must be at least one stage at which σ is active in the interval $(u,t]$, since otherwise $t+1$ could not be a σ-isomorphism recovery stage. Let v be the least stage in $(u,t]$ at which σ is active. In this case, again by the way components that participate in an operation at an isomorphism-recovery stage are chosen, K participated in an operation at stage v.

In either case, we have a contradiction. □

Lemma 2.19. *Suppose that $r_{\sigma,s}=i\neq r_{\sigma,s+1}$. Of all the components of $(\mathscr{A}^i)_\sigma$ 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 $S^i_{\sigma,s}$.*

Proof. Let t be the first stage after stage s at which σ is active. Then t is a σ-change stage, and hence not a σ-isomorphism recovery stage. It follows that, of all the components of $(\mathscr{A}^i)_\sigma$ that participate in operations before stage $s+1$, the only one that participates in an operation at stage t is the one that extends $S^i_{\sigma,s}$. The lemma now follows from Lemma 2.18. □

Lemma 2.20. *Suppose that $r_{\sigma,s}=i$ for all $s>t$, σ is not initialized at any stage after stage t, and σ is active at stages s_0+1 and s_1+1, where $s_1>s_0\geqslant t$. Then R^i_{σ,s_1} extends R^i_{σ,s_0}.*

Lemma 2.21. *Let u be a stage after which σ is never initialized. Let $s+1$ and $t+1$ be σ-recovery stages such that $s+1>t+1>u$ and there is no σ-recovery stage in the interval $(t+1,s+1)$. If $r_{\sigma,s}=0\neq r_{\sigma,s+1}$ then $S^0_{\sigma,s}$ extends $B^0_{\sigma,v}$ for some $v\in[t,s)$. Similarly, if $r_{\sigma,s}=1\neq r_{\sigma,s+1}$ then $S^1_{\sigma,s}$ extends $C^1_{\sigma,v}$ for some $v\in[t,s)$.*

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

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

Lemma 2.22. *Suppose that $r_{\sigma,t}=0$ (resp. $r_{\sigma,t}=1$) for all $t\geqslant s_0$. Then no component of $(\mathscr{A}^0)_\sigma$ ($(\mathscr{A}^1)_\sigma$) can participate in an operation more than twice after stage s_0 unless it extends $R^0_{\sigma,t}$ ($R^1_{\sigma,t}$) for some $t\geqslant s_0$, while no component of $(\mathscr{A}^1)_\sigma$ ($(\mathscr{A}^0)_\sigma$) can participate in an operation more than twice after stage s_0 unless it extends $C^1_{\sigma,t}$ ($B^0_{\sigma,t}$) for some $t\geqslant s_0$ such that $t+1$ is a σ-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 $(\mathscr{A}^0)_\sigma$ 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 $R^0_{\sigma,u}$ or $u+1$ is a σ-isomorphism recovery stage and K extends $C^0_{\sigma,u}$. We claim

that the latter case cannot hold. Indeed, if K extends $C^0_{\sigma,u}$ 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 $r_{\sigma,t+1} = 0$, $B^0_{\sigma,t}$ is a singleton component, which means that K does not participate in an operation at stage $s+1$, contrary to hypothesis.

Now suppose that component L of $(\mathscr{A}^1)_\sigma$ 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 $R^1_{\sigma,t}$ or $t+1$ is a σ-isomorphism recovery stage and L extends $C^1_{\sigma,t}$. But in the former case, $u+1$ is a σ-isomorphism recovery stage and, since K does not participate in operations at any stage in $(t+1, u+1)$, L extends $C^1_{\sigma,u}$. □

Lemma 2.23. *Let s_0 be a stage after which σ is never initialized. Suppose that $s_0 \leqslant s < t < v$ are such that $s+1$ is a σ-isomorphism recovery stage, $r_{\sigma,u} = r_{\sigma,s+1}$ for all $u > s$, $t+1$ is the next stage after stage $s+1$ at which σ is active, and $v+1$ is the next σ-isomorphism recovery stage after stage $s+1$. For $i = 0, 1$, let B^i, R^i, and C^i be the components of $\mathscr{A}^i_{t+1}$ that extend $B^i_{\sigma,t}$, $R^i_{\sigma,t}$, and $C^i_{\sigma,t}$, respectively, and let $\hat{B}^i$, $\hat{R}^i$, and $\hat{C}^i$ be the components of $\mathscr{A}^i_v$ that extend B^i, R^i, and C^i, respectively. If $r_{\sigma,s+1} = 0$ then $\hat{B}^0 \cong B^0$ and $\hat{R}^1 \cong R^1$, while if $r_{\sigma,s+1} = 1$ then $\hat{C}^1 \cong C^1$ and $\hat{R}^0 \cong R^0$.*

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 $(\mathscr{A}^0)_\sigma$ and $(\mathscr{A}^1)_\sigma$ 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 $(\mathscr{A}^0)_\sigma$ participates in operations at stages $t+1$ and $u+1$, where $t < u < v$. Since no stage in $(t+1, v+1)$ is a σ-isomorphism recovery stage, K extends $R^0_{\sigma,u}$, which in turn extends $R^0_{\sigma,t}$. Thus K does not extend B^0.

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

Lemma 2.24. *If σ is to the left of TP then $(\mathscr{A}^i)_{\supseteq\sigma}$ is finite.*

Lemma 2.25. *If σ is initialized at stage $s+1$ then no components of $(\mathscr{A}^i)_\sigma$ that participate in operations at stages before stage $s+1$ can participate in an operation after stage s. Thus if σ is to the right of TP then $(\mathscr{A}^i)_\sigma$ has no infinite components.*

Proof. Let t be the first stage after stage s at which σ is active. (If there are no such stages then we are done.) Then t is a σ-change stage, and hence not a σ-isomorphism recovery stage. It is easy to check that, together with the fact that σ is initialized at stage $s+1$, this implies that none of the components of $(\mathscr{A}^i)_\sigma$ that participate in operations before stage $s+1$ can participate in an operation at stage t. Thus the first part of the lemma follows from Lemma 2.18. The second part of the lemma now follows from Lemma 2.15. □

We are now ready to show that (2.1) holds. In the course of doing so, we will also be able to show that (2.4) holds. It follows from Lemmas 2.14, 2.15, and 2.17 that, to show that (2.1) holds, it is enough to show that for each infinite component of $\mathscr{A}^i$ there is a corresponding isomorphic component of $\mathscr{A}^{1-i}$. The first step in establishing this result is characterizing the infinite components of $\mathscr{A}^i$. Clearly, each infinite component of $\mathscr{A}^i$ is in $(\mathscr{A}^i)_\sigma$ for some $\sigma \in 2^{<\omega}$. By Lemmas 2.24 and 2.25, if σ is not on TP then no component of $(\mathscr{A}^i)_\sigma$ is infinite. By Lemmas 2.15 and 2.16, if σ is not active infinitely often then no component of $(\mathscr{A}^i)_\sigma$ is infinite. Thus, we can restrict our attention to the components of $(\mathscr{A}^i)_\sigma$, $\sigma \in TP$, such that σ is active infinitely often.

Lemma 2.26. *Let $\sigma \in TP$. If $r_{\sigma,s}$ does not have a limit then no component of $(\mathscr{A}^i)_\sigma$ is infinite.*

Proof. Suppose that $r_{\sigma,s} = 0 \neq r_{\sigma,s+1}$ and let $t+1$ be the last σ-recovery stage before stage $s+1$. By Lemma 2.19, of all the components of $(\mathscr{A}^0)_\sigma$ 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 $S^0_{\sigma,s}$. By Lemma 2.21, L extends $B^0_{\sigma,u}$ for some $u \in [t,s)$. But the fact that $r_{\sigma,t+1} = 0$ means that, for all $u \in [t,s)$, $B^0_{\sigma,u}$ is a singleton component, and hence did not participate in an operation at any stage before stage $t+1$.

Thus no component of $(\mathscr{A}^0)_\sigma$ that participates in an operation before stage $t+1$ can do so again after stage s. A similar argument shows that if $r_{\sigma,s} = 1 \neq r_{\sigma,s+1}$ and $t+1$ is the last σ-recovery stage before stage $s+1$ then no component of $(\mathscr{A}^1)_\sigma$ that participates in an operation before stage $t+1$ can do so again after stage s. The lemma now follows from Lemma 2.15. □

Thus the only components of $\mathscr{A}^i$ that can be infinite are those that are in $(\mathscr{A}^i)_\sigma$ for some $\sigma \in TP$ such that $r_{\sigma,s}$ has a limit and σ is active infinitely often. So, by the comments preceding Lemma 2.26, to establish that (2.1) holds it is enough to show that if $\sigma \in TP$, $r_{\sigma,s}$ has a limit, and σ is active infinitely often, then there is exactly one infinite component S^i_σ of $(\mathscr{A}^i)_\sigma$ for each $i = 0, 1$, and $S^0_\sigma \cong S^1_\sigma$. Together with the fact that no two infinite components of $\mathscr{A}^i$ are isomorphic, this will also be enough to show that (2.4) holds.

Lemma 2.27. *Let $\sigma \in TP$. There are infinitely many σ-recovery stages if and only if σ is active infinitely often.*

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

For the other direction, suppose that there are infinitely many σ-recovery stages but only finitely many stages at which σ is active. Let s be a stage after which σ is never active or initialized and such that there has been a σ-recovery stage since the last time

σ was initialized. Now, given $x > |\sigma|$, let $t+1$ be the first stage after stage s by which there have been $x+1$ many σ-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 σ would be active at stage $u+1$. But this means that A is computable, contrary to hypothesis. □

Lemma 2.28. *If $\sigma \in TP$ is active infinitely often and $r_{\sigma,s}$ has a limit then there are infinitely many σ-isomorphism recovery stages.*

Proof. If σ is active infinitely often then, by Lemma 2.27, there are infinitely many σ-recovery stages, and thus infinitely many σ-first stages. The fact that $r_{\sigma,s}$ has a limit and that σ is initialized only finitely often implies that only finitely many of these can be σ-change stages. The lemma now follows directly from the definition of σ-isomorphism recovery stage. □

Lemma 2.29. *Suppose that $\sigma \in TP$ is active infinitely often and s and i are such that σ is not initialized after stage s and $r_{\sigma,t} = r_{\sigma,s} = i$ for all $t \geqslant s$. By Lemma* 2.28, *there are infinitely many σ-isomorphism recovery stages. Let $s_0 + 1 < s_1 + 1 < \cdots$ be the σ-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 σ is active. (Note that $t_j < s_{j+1}$ for all $j \in \omega$.) For $t \geqslant t_0$, let K_t^l be the component of $\mathscr{A}_t^l$ that extends R_{σ,t_0}^l. Then $K_{t_j}^l = R_{\sigma,t_j}^l$ for all $j \in \omega$.*

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

That $K_{t_j}^0 = R_{\sigma,t_j}^0$ for all $j \in \omega$ follows from Lemma 2.20. Now assume by induction that $K_{t_j}^1 = R_{\sigma,t_j}^1$. Let B be the component of $\mathscr{A}_{t_j+1}^0$ that extends B_{σ,t_j}^0. By construction, $B \cong K_{t_j+1}^1$. Since $s_{j+1} + 1$ is a σ-isomorphism recovery stage, $C_{\sigma,s_{j+1}}^0$ extends B. Thus, by Lemma 2.23, $C_{\sigma,s_{j+1}}^0 \cong B$. By the same lemma, $K_{s_{j+1}}^1 \cong K_{t_j+1}^1$, so $C_{\sigma,s_{j+1}}^0 \cong K_{s_{j+1}}^1$, and hence $C_{\sigma,s_{j+1}}^1 = K_{s_{j+1}}^1$. Let R be the component of $\mathscr{A}_{s_{j+1}+1}^0$ that extends $R_{\sigma,s_{j+1}}^0$. Then $R \cong K_{s_{j+1}+1}^1$. But, by Lemma 2.16, $R_{\sigma,t_{j+1}}^0 \cong R$ and $K_{t_{j+1}}^1 \cong K_{s_{j+1}+1}^1$, so $K_{t_{j+1}}^1 \cong R_{\sigma,t_{j+1}}^0$, and hence $K_{t_{j+1}}^1 = R_{\sigma,t_{j+1}}^1$. □

Now assume the hypotheses of Lemma 2.29 and adopt its notation. For $l = 0, 1$, let S_σ^l be the component of $\mathscr{A}^l$ that extends R_{σ,s_0}^l.

Lemma 2.30. *S_σ^l is the only infinite component of $(\mathscr{A}^l)_\sigma$.*

Proof. This follows immediately from Lemmas 2.15, 2.22, and 2.29 and the observation that, for all $j \in \omega$, if $i = 0$ in the hypotheses of Lemma 2.29 then R_{σ,t_j}^1 extends C_{σ,s_j}^1, while if $i = 1$ then R_{σ,t_j}^0 extends B_{σ,s_j}^0. □

Lemma 2.31. $S_\sigma^0 \cong S_\sigma^1$.

Proof. This follows immediately from Lemma 2.29, since, by definition, $R_{\sigma,t_j}^0 \cong R_{\sigma,t_j}^1$ for all $j \in \omega$, and $S_\sigma^i = \bigcup_{j \in \omega} R_{\sigma,t_j}^i$ for $i = 0, 1$. □

As we have argued above, Lemmas 2.30 and 2.31 suffice to establish that (2.1) holds.

Lemma 2.32. *$\mathscr{A}^0 \cong \mathscr{A}^1$ via an isomorphism that carries U^0 to U^1.*

To show that (2.4) holds, we need to check that if $\sigma \neq \tau$ and $(\mathscr{A}^i)_\sigma$ and $(\mathscr{A}^i)_\tau$ have infinite components S^i_σ and S^0_τ, respectively, then $S^i_\sigma \not\cong S^0_\tau$. This is a consequence of the following lemma, which will also be useful later on.

Lemma 2.33. *No component of $\mathscr{A}^i$ is embeddable in another component of $\mathscr{A}^i$.*

Proof. For finite components this follows from Lemma 2.14. For infinite components it follows from Lemma 2.30 and the fact that if $(\mathscr{A}^i)_\tau$ has an infinite component S^i_τ then S^i_τ contains a copy of $[6\langle \ulcorner \sigma \urcorner, k\rangle + 3]$ for some $k \in \omega$ if and only if $\tau = \sigma$. □

Lemma 2.34. *$\mathscr{A}^0$ is rigid.*

Proof. By Lemma 2.13, it is enough to show that no two components of $\mathscr{A}^0$ are isomorphic. By Lemma 2.14, for each $s \in \omega$, no component of $\mathscr{A}^0_s$ is embeddable in another component of $\mathscr{A}^0_s$, which implies that no two finite components of $\mathscr{A}^0$ are isomorphic. Since the only infinite components of $\mathscr{A}^0$ are the S^0_σ defined above and, by Lemma 2.33, $S^0_\sigma \cong S^0_\tau$ if and only if $\tau = \sigma$, it is also the case that no two infinite components of $\mathscr{A}^0$ are isomorphic. □

We are left with showing that (2.3) holds. This is where this proof differs most significantly from that of Theorem 1.9. We begin by showing that if $\sigma \in TP$ and $\mathscr{G}_{|\sigma|} \cong \mathscr{A}^0$ then $\lim_s r_{\sigma,s}$ is well-defined.

Lemma 2.35. *If $\sigma \in TP$ and $\mathscr{G}_{|\sigma|} \cong \mathscr{A}^0$ then there are infinitely many σ-recovery stages, and hence the σ-special component of $\mathscr{G}_{|\sigma|}$ is infinite.*

Proof. Assume for a contradiction that there are only m many σ-recovery stages and let s_0 be the last σ-recovery stage. (If $m = 0$ then let $s_0 = 0$.) By Lemma 2.27, there is a stage $s_1 > s_0$ such that σ is not active at any stage $t \geqslant s_1$. By the definition of TP and the hypothesis that $\mathscr{G}_{|\sigma|} \cong \mathscr{A}^0$, there is a stage $s_2 \geqslant s_1$ satisfying the following conditions: every τ such that $\tau^\frown 0 \subseteq \sigma$ has recovered at least $|\sigma| + 1$ many times by stage s_2, $\mathscr{G}_{|\sigma|}[s_2]$ has a σ-special component, and σ is not initialized at any stage greater than or equal to s_2. If $m = a_u$ for some $u > s_2$ then let $s = u + 1$; otherwise, let $s = s_2$.

By the definition of s, the first condition in the definition of σ-recovery stage is met at every stage greater than or equal to s.

Consider the components of $\mathscr{A}^0$ that contain a copy of the σ-special component of $\mathscr{G}_{|\sigma|}$. By Lemma 2.16, each such component is finite. Thus, if the second condition in the definition of σ-recovery stage is not eventually satisfied after stage s then the σ-special component of $\mathscr{G}_{|\sigma|}$ is not isomorphic to any component of $\mathscr{A}^0$.

Since we are assuming that $\sigma^\frown 0$ is to the left of TP, there is a stage $t \geqslant s$ after which no τ such that $\tau \supseteq \sigma^\frown 0$ is initialized. Any such τ that has not recovered since the last time it was initialized never again recovers, and hence there is a component of $\mathscr{A}^0$ isomorphic to $[6\langle \ulcorner \tau \urcorner, \mathit{init}(\tau, t)\rangle + 3]$. Since there are only finitely many τ such that $|\tau| \leqslant \mathit{recov}(\sigma, s)$, if the third condition in the definition of σ-recovery stage is not eventually satisfied after stage s then $\mathscr{G}_{|\sigma|} \ncong \mathscr{A}^0$.

Now consider $(\mathscr{A}^0)_\sigma$. Again by Lemma 2.16, $(\mathscr{A}^0)_\sigma$ is finite. So if the fourth condition in the definition of σ-recovery stage is not eventually satisfied after stage s then $(\mathscr{G}_{|\sigma|})_\sigma \ncong (\mathscr{A}^0)_\sigma$.

Since we are assuming that there are only finitely many σ-recovery stages, $\sigma^\frown 1 \in TP$. Thus, by Lemma 2.24, $(\mathscr{A}^0_s)_{\supseteq \sigma^\frown 0}$ is finite. So if the fifth condition in the definition of σ-recovery stage is not eventually satisfied after stage s then $(\mathscr{G}_{|\sigma|})_{\supseteq \sigma^\frown 0} \ncong (\mathscr{A}^0)_{\supseteq \sigma^\frown 0}$.

Finally, let τ be such that either $\tau = \sigma$ or both $\tau \supseteq \sigma^\frown 0$ and $|\tau| \leqslant \mathit{recov}(\sigma, s)$. Let $j \notin A[s]$ be less than or equal to $\mathit{recov}(\tau, s)$. Clearly, $c(\tau, t)$ reaches a limit $c(\tau)$. By the choice of s, $j \notin A[s] \Rightarrow j \notin A$. So, for each $l \in \{1, 2, 4, 5\}$, there is a unique component of $\mathscr{A}^0$ that contains a copy of $[6\langle \ulcorner \tau \urcorner, j, c(\tau)\rangle + l]$, and it is isomorphic to $[6\langle \ulcorner \tau \urcorner, j, c(\tau)\rangle + l]$. Similarly, there is a unique component of $\mathscr{A}^0$ that contains a copy of $[6j]^+$, and it is isomorphic to $[6j]^+$. Thus, if the last condition in the definition of σ-recovery stage is not eventually satisfied after stage s then there is a component of $\mathscr{A}^0$ that is not isomorphic to any component of $\mathscr{G}_{|\sigma|}$.

In any case, $\mathscr{G}_{|\sigma|}$ cannot be isomorphic to $\mathscr{A}^0$, contrary to hypothesis. So there are infinitely many σ-recovery stages.

Now, let v be a stage after which σ is never initialized. Given any two σ-recovery stages $v < t + 1 < u + 1$ such that there is a stage in $(t, u]$ at which σ is active, the σ-special component of $\mathscr{G}_{|\sigma|}[u]$ properly extends the σ-special component of $\mathscr{G}_{|\sigma|}[t]$. Since, by Lemma 2.27, σ is active at infinitely many stages, this establishes the second part of the lemma. □

Lemma 2.36. *If $\sigma \in TP$ and $\mathscr{G}_{|\sigma|} \cong \mathscr{A}^0$ then $\lim_s r_{\sigma,s}$ is well-defined.*

Proof. This follows immediately from Lemmas 2.26 and 2.35. □

Now fix $\sigma \in TP$ such that $\mathscr{G}_{|\sigma|} \cong \mathscr{A}^0$ and let $n = |\sigma|$. By Lemma 2.36, $r = \lim_s r_{\sigma,s}$ is well-defined. We wish to show that $\mathscr{G}_n$ is computably isomorphic to $\mathscr{A}^r$. The two cases, $r = 0$ and $r = 1$, are symmetrical, so we will assume that $r = 0$.

Let $f : \mathscr{A}^0 \cong \mathscr{G}_n$. Since $\mathscr{A}^0$ is rigid, f is the unique isomorphism from $\mathscr{A}^0$ to $\mathscr{G}_n$, so we need to show that f is computable. As outlined at the beginning of this section, our strategy will be to break up the domain of $\mathscr{A}^0$ into a finite number of c.e. sets and show that the restriction of f to each of these sets is computable. (If P is c.e. then we say that $f \upharpoonright P$ is computable if there exists a partial computable function Φ such that $x \in P \Rightarrow \Phi(x)\downarrow = f(x)$.) We will need the following definition.

Table 1
$D_0, \ldots, D_6$

D_0	$\{6\langle 0,k\rangle + 3,\ 6\langle 0,j,k\rangle + l \mid j,k \in \omega,\ l \in \{1,2,4,5\}\}$
D_1	$\{6\langle \ulcorner\tau\urcorner,k\rangle + 3,\ 6\langle \ulcorner\tau\urcorner,j,k\rangle + l \mid \tau$ to the left of σ or $\tau^\frown 1 \subseteq \sigma$, $j,k \in \omega,\ l \in \{1,2,4,5\}\}$
D_2	$\{6\langle \ulcorner\tau\urcorner,k\rangle + 3,\ 6\langle \ulcorner\tau\urcorner,j,k\rangle + l \mid \tau$ to the right of $\sigma^\frown 0,\ j,k \in \omega,\ l \in \{1,2,4,5\}\}$
D_3	$\{m \in \omega \mid (m)$ is the unique infinite component of some $(\mathscr{A}^0)_\tau,\ \tau^\frown 0 \subseteq \sigma\}$
D_4	$\{6\langle \ulcorner\tau\urcorner,j\rangle + 3,\ 6\langle \ulcorner\tau\urcorner,j,k\rangle + l \mid \tau^\frown 0 \subseteq \sigma,\ j,k \in \omega,\ l \in \{1,2,4,5\}\} - D_3$
D_5	$\{6k \mid k < n\} \cup \{6a_s \mid a_s \geqslant recov(\sigma, s+1)$ or s is less than the first σ-recovery stage after the last time σ is initialized$\}$
D_6	$\{6\langle \ulcorner\tau\urcorner,j\rangle + 3,\ 6\langle \ulcorner\tau\urcorner,j,k\rangle + l \mid \tau = \sigma$ or $\sigma^\frown 0 \subseteq \tau,\ j,k \in \omega,\ l \in \{1,2,4,5\}\} \cup \{6k \mid k \in \omega\} - D_5$

Definition 2.37. Let $k,s \in \omega$. We denote by (k) and $(k)_s$ the components of $\mathscr{A}^0$ and $\mathscr{A}^0_s$, respectively, that extend the unique copy of $[k]$ in $\mathscr{A}^0_0$.

For $D \subseteq \omega$, let $P_D = \bigcup_{k \in D}(k)$.

Note that, for any $k,s \in \omega$, $(k)_s$ is finite. Note also that, since every component of $\mathscr{A}^0$ extends some component of $\mathscr{A}^0_0$, $\bigcup_{k\in\omega}(k) = \mathscr{A}^0$; similarly, $\bigcup_{k\in\omega}(k)_s = \mathscr{A}^0_s$. It is not the case that $k \neq l \Rightarrow (k) \neq (l)$, but, as we will see, this will not matter for our purposes.

Lemma 2.38. *Let $D_0, \ldots, D_m$ be computable subsets of ω such that $\bigcup_{i=0}^m D_i = \omega$. If $f \upharpoonright P_{D_i}$ is computable for each $i \leqslant m$ then f is computable.*

Proof. Since $D_0, \ldots, D_m$ are computable, $P_{D_0}, \ldots, P_{D_m}$ are c.e. Since $\bigcup_{i=0}^m D_i = \omega$, $\bigcup_{i=0}^m P_{D_i} = \mathscr{A}^0$. Thus, to compute $f(x)$ for some $x \in \mathscr{A}^0$, all we need to do is wait until x is enumerated into some P_{D_i} and then compute $(f \upharpoonright P_{D_i})(x)$. □

We will partition ω into the pairwise disjoint computable sets $D_0, \ldots, D_6$ shown in Table 1. (The corresponding P_{D_i} will not be pairwise disjoint, but this does not matter, since it was not required to prove Lemma 2.38.) We will then show that, for each $i \leqslant 6$, $f \upharpoonright P_{D_i}$ is computable, which will enable us to apply Lemma 2.38 to conclude that f is computable. The following two lemmas provide a useful tool for our task.

Lemma 2.39. *Let $k \in \omega$ and suppose there is a stage s such that, for each $t \geqslant s$, $(k)_t$ does not participate in an operation at stage $t+1$. Then $(k) \cong (k)_s$.*

Proof. Clearly, if $(k)_t$ does not participate in an operation at stage $t+1$ then $(k)_{t+1} \cong (k)_t$. So, by induction, $(k)_t \cong (k)_{s+1}$ for all $t \geqslant s$. Since $(k) = \bigcup_{t\in\omega}(k)_t$, the lemma follows. □

Lemma 2.40. *Let both $D \subseteq \omega$ and $h : D \to \omega$ be computable. Suppose that, for each $k \in D$ and $t \geqslant h(k)$, $(k)_t$ does not participate in an operation at stage $t+1$. Then $f \restriction P_D$ is computable.*

Proof. Let $x \in P_D$ and let $k \in D$ be such that $x \in (k)$. By Lemma 2.39, $(k)_{h(k)} \cong (k)$, so (k) is finite. By Lemma 2.33, there is a unique finite set $T \subset \mathscr{G}_n$ such that there is an isomorphism $g_x : (k) \cong T$. Clearly, g_x can be extended to an isomorphism from $\mathscr{A}^0$ to $\mathscr{G}_n$. By the uniqueness of f, $f(x) = g_x(x)$. Since g_x can be computably determined given $x \in P_D$, this implies that $f \restriction P_D$ is computable. □

Lemma 2.41. *Let D_0 consist of all numbers of the form $6\langle 0, k\rangle + 3$ or $6\langle 0, j, k\rangle + l$, $j, k \in \omega$, $l \in \{1, 2, 4, 5\}$. Then $f \restriction P_{D_0}$ is computable.*

Proof. Let m be of the form $6\langle 0, k\rangle + 3$ or $6\langle 0, j, k\rangle + l$, $j, k \in \omega$, $l \in \{1, 2, 4, 5\}$. Recall that, for all $\tau \in 2^{<\omega}$, $\ulcorner \tau \urcorner \neq 0$. Thus the only time (m) can participate in an operation is at stage $k+1$. (This happens if no element of $2^{<\omega}$ is active at stage $k+1$.) So if we define $h(m) = k+1$ then the hypotheses of Lemma 2.40 are satisfied for $D = D_0$. □

Lemma 2.42. *There exists a stage s such that if τ is either to the left of σ or such that $\tau^\frown 1 \subseteq \sigma$ then τ is not active after stage s.*

Proof. Let T be the set of all τ which are either to the left of σ or such that $\tau^\frown 1 \subseteq \sigma$. Since $\sigma \in TP$, only finitely many elements of T ever recover, and those that do recover, do so only finitely often. But, by definition, no $\tau \in 2^{<\omega}$ can be active at a stage $t+1$ unless a_t is less than the number of τ-recovery stages less than or equal to $t+1$, and hence no τ can be active more often than it recovers. □

Lemma 2.43. *Let D_1 be the set of all numbers of the form $6\langle \ulcorner \tau \urcorner, k\rangle + 3$ or $6\langle \ulcorner \tau \urcorner, j, k\rangle + l$, τ to the left of σ or $\tau^\frown 1 \subseteq \sigma$, $j, k \in \omega$, $l \in \{1, 2, 4, 5\}$. Then $f \restriction P_{D_1}$ is computable.*

Proof. Let s be as in Lemma 2.42. By Lemma 2.16, for each $m \in D_1$ and $t \geqslant s$, $(m)_t$ does not participate in an operation at stage $t+1$. So if we let $h(m) = s$ for all $m \in D_1$ then the hypotheses of Lemma 2.40 are satisfied for $D = D_1$. □

Lemma 2.44. *Let τ be to the right of $\sigma^\frown 0$. Let m be of the form $6\langle \ulcorner \tau \urcorner, k\rangle + 3$ or $6\langle \ulcorner \tau \urcorner, j, k\rangle + l$, $l \in \{1, 2, 4, 5\}$. Let $s+1$ be a stage by which by τ has been initialized $k+1$ many times. Then (m) does not participate in an operation after stage s.*

Proof. If a singleton component of $\mathscr{A}_t^0$ of the form $[6\langle \ulcorner \tau \urcorner, p\rangle + 3]$ participates in an operation at a stage $t+1 > s$ then $p = \mathit{init}(\tau, t) \geqslant k+1$. If a singleton component of $\mathscr{A}_t^0$ of the form $[6\langle \ulcorner \tau \urcorner, j, p\rangle + l]$, $l \in \{1, 2, 4, 5\}$, participates in an operation at a

stage $t+1>s$ then $p=c(\tau,t) \geqslant init(\tau,t) \geqslant k+1$. So if (m) does not participate in an operation before stage $s+1$ then it does not participate in an operation after stage s.

On the other hand, if (m) participates in an operation before stage $s+1$ then the fact that it does not participate in an operation after stage s follows from Lemma 2.25. □

Lemma 2.45. *Let D_2 be the set of all numbers of the form $6\langle \ulcorner \tau \urcorner, k\rangle+3$ or $6\langle \ulcorner \tau \urcorner, j,k\rangle+l$, τ to the right of $\sigma^\frown 0$, $j,k \in \omega$, $l \in \{1,2,4,5\}$. Then $f \upharpoonright P_{D_2}$ is computable.*

Proof. If $m \in D_2$ is of the form $6\langle \ulcorner \tau \urcorner, k\rangle + 3$ or $6\langle \ulcorner \tau \urcorner, j,k\rangle + l$ then define $h(m)$ to be the first stage by which τ has been initialized $k+1$ many times (which exists, since $\sigma^\frown 0 \in TP$). Then, by Lemma 2.44, the hypotheses of Lemma 2.40 are satisfied for $D=D_2$. □

If $\tau^\frown 0 \subseteq \sigma$ and $r_{\tau,s}$ has a limit then, by Lemma 2.30, $(\mathscr{A}^0)_\tau$ has a unique infinite component. On the other hand, if $\tau^\frown 0 \subseteq \sigma$ and $r_{\tau,s}$ does not have a limit then, by Lemma 2.26, all components of $(\mathscr{A}^0_\tau)$ are finite. Let D_3 be the set of all $m \in \omega$ such that (m) is the unique infinite component of some $\tau^\frown 0 \subseteq \sigma$ such that $r_{\tau,s}$ has a limit. Note that D_3 is finite.

Lemma 2.46. *$f \upharpoonright P_{D_3}$ is computable.*

Proof. Let $T=\{x_0,\ldots,x_m\}$ be the tops of the components of P_{D_3}. Let $x \in P_{D_3}-T$. By Lemma 2.13, there is a unique k such that x is in a copy K of $[k]$. The top of K is x_i for some $i \leqslant m$. Let L be the unique copy of $[k]$ in $\mathscr{G}_n$ with top $f(x_i)$ and let g_x be the unique isomorphism form K to L. By the uniqueness of f, $f(x)=g_x(x)$. Since g_x can be computably determined given $x \in P_{D_3}-T$ and T is finite, this implies that $f \upharpoonright P_{D_3}$ is computable. □

Lemma 2.47. *Let D_4 be the set of all numbers not in D_3 that are of the form $6\langle \ulcorner \tau \urcorner, j\rangle + 3$ or $6\langle \ulcorner \tau \urcorner, j,k\rangle + l$, $\tau^\frown 0 \subseteq \sigma$, $j,k \in \omega$, $l \in \{1,2,4,5\}$. Then $f \upharpoonright P_{D_4}$ is computable.*

Proof. Let $m \in D_4$. If m is of the form $6\langle \ulcorner \tau \urcorner, j,k\rangle + l$, $l \in \{1,2,4,5\}$, then let s be the first stage by which τ has recovered $k+1$ many times. If (m) has not participated in an operation before stage s then, by the same reasoning as in the proof of Lemma 2.44, it does not participate in an operation after stage s. In this case, let $h(m)=s$.

Now suppose that m is of the form $6\langle \ulcorner \tau \urcorner, j\rangle + 3$. Let $init(\tau)=\lim_s init(\tau,s)$, which exists since $\tau \in TP$. If $j<init(\tau)$ then let s be the least stage by which τ has been initialized $k+1$ many times. Arguing as in the proof of Lemma 2.44, we see that (m) does not participate in an operation after stage s. In this case, let $h(m)=s$. If $j>init(\tau)$ then (m) never participates in an operation. In this case, let $h(m)=0$.

If $h(m)$ has not yet been defined then (m) participates in an operation at least once. However, since (m) is finite, (m) participates in operations only finitely often, so there exist stages $s<t$ such that (m) does not participate in an operation in the interval $(s,t]$ and there is a τ-isomorphism recovery stage in $(s,t]$. By Lemma 2.18, (m) does not participate in an operation after stage t. In this case, let $h(m)=t$.

Now the hypotheses of Lemma 2.40 are satisfied for $D=D_4$. □

Lemma 2.48. *Let D_5' be the set of all numbers of the form $6k$, $k<n$. Let D_5'' be set of all numbers of the form $6a_s$ such that $a_s \geqslant recov(\sigma,s+1)$ or s is less than the first σ-recovery stage after the last time σ is initialized. Let $D_5=D_5'\cup D_5''$. Then $f\upharpoonright P_{D_5}$ is computable.*

Proof. By Lemma 2.16, there is a stage t such that no $(6k)$, $k<n$, participates in an operation after stage t. For $k<n$, let $h(6k)=t$. For $6a_s\in D_5''$, let $h(6a_s)=s+1$. Again by Lemma 2.16, $(6a_s)$ does not participate in an operation after stage $h(6a_s)$. Since D_5' is finite, h is computable, and hence the hypotheses of Lemma 2.40 are satisfied for $D=D_5$. □

Let D_6' be the set of all numbers of the form $6\langle\ulcorner\tau\urcorner,j\rangle+3$ or $6\langle\ulcorner\tau\urcorner,j,k\rangle+l$, $\tau=\sigma$ or $\sigma^\frown 0\subseteq\tau$, $j,k\in\omega$, $l\in\{1,2,4,5\}$. Let D_6'' be the set of all numbers of the form $6k$ that are not in D_5. Let $D_6=D_6'\cup D_6''$. In order to show that f is computable, we are left with showing that $f\upharpoonright P_{D_6}$ is computable. Roughly speaking, the idea is to show that, once $r_{\sigma,s}$ has reached its final value, $\mathscr{G}_n$ and $\mathscr{A}^0$ always go in the same direction at stages at which components of P_{D_6} participate in operations.

Lemma 2.49. *Let τ be such that $\tau=\sigma$ or $\sigma^\frown 0\subseteq\tau$. Let u be a stage after which σ is never initialized and such that, for all $s\geqslant u$, $r_{\sigma,s}=0$. Let $s+1$ and $t+1$ be σ-recovery stages such that $s+1>t+1>u$ and there is no σ-recovery stage in the interval $(t+1,s]$, and let $s_0+1<s_1+1<\cdots<s_m+1$ be the stages in the interval $(t,s]$ at which τ is active. For each $k\leqslant m$, let Y_k, X_k, Z_k, B_k, R_k, and C_k be Y^0_{τ,s_k}, $X^0_{s_k}$, Z^0_{τ,s_k}, B^0_{τ,s_k}, R^0_{τ,s_k}, and C^0_{τ,s_k}, respectively, and let Y_k', X_k', Z_k', B_k', R_k', and C_k' be the components of $\mathscr{A}^0_s$ 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 $\mathscr{A}^0_t$, and so is R_0. If $r_{\tau,t+1}=0$ then, for every $k,l\leqslant m$, $R_k'=R_l'$. If $r_{\tau,t+1}=1$ then, for every $0<k\leqslant m$, $R_k'=B_{k-1}'$.*
2. *There exists a component $\hat{R}_0$ of $\mathscr{G}_n[t]$ such that $\hat{R}_0\cong R_0$ and, for each $k\leqslant m$, there exist components $\hat{Y}_k$, $\hat{X}_k$, $\hat{Z}_k$, $\hat{B}_k$, and $\hat{C}_k$ of $\mathscr{G}_n[t]$ such that $\hat{Y}_k\cong Y_k$, $\hat{X}_k\cong X_k$, $\hat{Z}_k\cong Z_k$, $\hat{B}_k\cong B_k$, and $\hat{C}_k\cong C_k$.*
3. *Let $\hat{R}_0'$ be the component of $\mathscr{G}_n[s]$ that extends $\hat{R}_0$ and, for each $k\leqslant m$, let $\hat{Y}_k'$, $\hat{X}_k'$, $\hat{Z}_k'$, $\hat{B}_k'$, and $\hat{C}_k'$ be the components of $\mathscr{G}_n[s]$ that extend $\hat{Y}_k$, $\hat{X}_k$, $\hat{Z}_k$, $\hat{B}_k$, and $\hat{C}_k$, respectively. $\hat{R}_0'\cong R_0'$ and, for each $k\leqslant m$, $\hat{Y}_k'\cong Y_k'$, $\hat{X}_k'\cong X_k'$, $\hat{Z}_k'\cong Z_k'$, $\hat{B}_k'\cong B_k'$, and $\hat{C}_k'\cong C_k'$.*

Proof. There are no τ-recovery stages in the interval $(t+1,s]$, which implies that if τ is initialized in the interval $(t,s]$ then this initialization happens after stage s_m+1. So the first part of the lemma follows from the way Y^0_{τ,s_k}, $X^0_{s_k}$, Z^0_{τ,s_k}, B^0_{τ,s_k}, R^0_{τ,s_k}, and C^0_{τ,s_k} are defined. The second part of the lemma follows from the definition of σ-recovery stage. We prove the third part of the lemma. Figs. 7–9 might be helpful here.

We begin with the $\tau=\sigma$ case.

By definition, $\hat{R}_0$ and $\hat{R}'_0$ are the special components of $\mathscr{G}_n[t]$ and $\mathscr{G}_n[s]$, respectively. Thus, since $r_{\sigma,s+1}=r_{\sigma,s}=0$ and $s+1$ is a σ-recovery stage, $\hat{R}'_0\cong R'_0$. 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 $\hat{R}'_0$, $\hat{Y}'_k$, $\hat{X}'_k$, $\hat{Z}'_k$, $\hat{B}'_k$, and $\hat{C}'_k$, $k\leqslant m$, and $L\neq K$ is taken from among $\hat{R}'_0$, $\hat{Y}'_l$, $\hat{X}'_l$, $\hat{Z}'_l$, $\hat{B}'_l$, and $\hat{C}'_l$, $l\leqslant m$, then $K\ncong L$. Furthermore, if K is one of $\hat{Y}'_k$, $\hat{X}'_k$, $\hat{Z}'_k$, $\hat{B}'_k$, or $\hat{C}'_k$, and L is a component of $\mathscr{A}^0_s$ 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 $j>k$, $\hat{Y}'_j\cong Y'_j$, $\hat{X}'_j\cong X'_j$, $\hat{Z}'_j\cong Z'_j$, $\hat{B}'_j\cong B'_j$, and $\hat{C}'_j\cong C'_j$, we may assume that if K is one of $\hat{Y}'_k$, $\hat{X}'_k$, $\hat{Z}'_k$, $\hat{B}'_k$, or $\hat{C}'_k$ and L is a component of $\mathscr{A}^0_s$ 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 $\hat{C}_k$ are R'_0 and C'_k. Since $\hat{R}'_0\cong R'_0$, it must be the case that $\hat{C}'_k\cong C'_k$.

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

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

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

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

This completes the $\tau=\sigma$ case. We now handle the $\tau\supseteq\sigma^\frown 0$ case. There are two subcases.

First suppose that $r_{\tau,t+1}=0$.

Let $k\leqslant m$. Since σ is active whenever τ is active, it follows from the $\tau=\sigma$ case that $\hat{X}'_k\cong X'_k$.

The only components of $\mathscr{A}^0_s$ that contain copies of $\hat{Z}_k$ are X'_k and Z'_k. Since $\hat{X}'_k\cong X'_k$, it must be the case that $\hat{Z}'_k\cong Z'_k$.

The only components of $\mathscr{A}^0_s$ that contain copies of $\hat{B}_k$ are Z'_k and B'_k. Since $\hat{Z}'_k\cong Z'_k$, it must be the case that $\hat{B}'_k\cong B'_k$.

The only components of $\mathscr{A}^0_s$ that contain copies of $\hat{R}_0$ are R'_0 and $B'_0,\ldots,B'_m$. We have shown that, for every $k\leqslant m$, $\hat{B}'_k\cong B'_k$. Thus it must be the case that $\hat{R}'_0\cong R'_0$.

We now proceed by reverse induction, beginning with m. Let $k\leqslant m$. Assume by induction that, for all $j>k$, $\hat{Y}'_j\cong Y'_j$, $\hat{X}'_j\cong X'_j$, $\hat{Z}'_j\cong Z'_j$, $\hat{B}'_j\cong B'_j$, and $\hat{C}'_j\cong C'_j$. As in the $\tau=\sigma$ case, we may assume that if K is one of $\hat{Y}'_k$, $\hat{X}'_k$, $\hat{Z}'_k$, $\hat{B}'_k$, or $\hat{C}'_k$ and L is a component of $\mathscr{A}^0_s$ such that $K\cong L$ then L is one of R'_0, Y'_k, X'_k, Z'_k, B'_k, or C'_k.

We have already seen that $\hat{X}'_k \cong X'_k$, $\hat{Z}'_k \cong Z'_k$, $\hat{B}'_k \cong B'_k$, and $\hat{R}'_0 \cong R'_0$.

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

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

This completes the $r_{\tau,t+1} = 0$ case. Now suppose that $r_{\tau,t+1} = 1$.

As before, it follows from the $\tau = \sigma$ case that $\hat{X}'_k \cong X'_k$ for all $k \leqslant m$.

We first proceed by reverse induction, beginning with m, to show that $\hat{Z}'_k \cong Z'_k$, $\hat{B}'_k \cong B'_k$, and $\hat{R}'_0 \cong R'_0$. Let $k \leqslant m$. We may assume by induction that, for all $k < j \leqslant m$, $\hat{B}'_j \cong B'_j$.

The only components of $\mathscr{A}^0_s$ that contain copies of $\hat{Z}_k$ are X'_k and Z'_k. Since $\hat{X}'_k \cong X'_k$, it must be the case that $\hat{Z}'_k \cong Z'_k$.

The only components of $\mathscr{A}^0_s$ that contain copies of $\hat{B}_k$ are Z'_k, and B'_j, $k \leqslant j \leqslant m$. Since $\hat{Z}'_k \cong Z'_k$ and, for all $k < j \leqslant m$, $\hat{B}'_j \cong B'_j$, it must be the case that $\hat{B}'_k \cong B'_k$.

The only components of $\mathscr{A}^0_s$ that contain copies of $\hat{R}_0$ are R'_0 and $B'_0, \ldots, B'_m$. We have shown that, for every $k \leqslant m$, $\hat{B}'_k \cong B'_k$. Thus it must be the case that $\hat{R}'_0 \cong R'_0$.

Now let $0 < k \leqslant m$. The only components of $\mathscr{A}^0_s$ that contain copies of $\hat{C}_k$ are B'_{k-1} and C'_k. Since $\hat{B}'_{k-1} \cong B'_{k-1}$, it must be the case that $\hat{C}'_k \cong C'_k$.

The only components of $\mathscr{A}^0_s$ that contain copies of $\hat{C}_0$ are R'_0 and C'_0. Since $\hat{R}'_0 \cong R'_0$, it must be the case that $\hat{C}'_0 \cong C'_0$.

Let $k \leqslant m$. The only components of $\mathscr{A}^0_s$ that contain copies of $\hat{Y}_k$ are C'_k and Y'_k. Since $\hat{C}'_k \cong C'_k$, it must be the case that $\hat{Y}'_k \cong Y'_k$.

This completes the $r_{\tau,t+1} = 1$ case. □

The following lemma can be easily checked from the way components that participate in operations in the construction are chosen.

Lemma 2.50. *Let $m \in \omega$ be of the form $6\langle \ulcorner \tau \urcorner, j\rangle + 3$ or $6\langle \ulcorner \tau \urcorner, j, k\rangle + l$, $\tau = \sigma$ or $\sigma^\frown 0 \subseteq \tau$, $j, k \in \omega$, $l \in \{1, 2, 4, 5\}$. If $(m)_s$ participates in an operation at stage $s+1$ then it is one of $Y^0_{\tau,s}$, $Z^0_{\tau,s}$, $B^0_{\tau,s}$, $R^0_{\tau,s}$, or $C^0_{\tau,s}$.*

Let $m \in D''_6$. If $(m)_s$ participates in an operation at stage $s+1$ then it is X^0_s and σ is active at stage $s+1$.

Lemma 2.51. *Let u be a stage after which σ is never initialized and such that, for all $s \geqslant u$, $r_{\sigma,s} = 0$. Let $s+1 > u$ be a σ-recovery stage and let $t+1$ be the next σ-recovery stage after stage $s+1$. Let $m \in D_6$. Suppose there exists a component L of $\mathscr{G}_n[s]$ that is isomorphic to $(m)_s$. Then the component L' of $\mathscr{G}_n[t]$ that extends L is isomorphic to $(m)_t$.*

Proof. If (m) does not participate in an operation in the interval $(s, t]$ then $(m)_t \cong (m)_s$. Since $L' \supseteq L$, $(m)_t$ is not embeddable in another component of $\mathscr{A}^0_t$, and, by convention (see Sections 2.4 and 2.5), $\mathscr{G}_n[t]$ is embeddable in $\mathscr{A}^0_t$, this means that $L' \cong (m)_t$.

Otherwise, the lemma follows from Lemmas 2.49 and 2.50. □

Lemma 2.52. *Let $x \in P_{D_6}$ and let u be a stage after which σ is never initialized and such that, for all $s \geqslant u$, $r_{\sigma,s} = 0$. There exists a σ-recovery stage $s+1 \geqslant u$ such that x is contained in $(k)_s$ for some $k \in D_6$ and $\mathscr{G}_n[s]$ has a component $L \cong (k)_s$. For any such s, if we let g be the unique isomorphism from $(k)_s$ to L then $f(x) = g(x)$.*

Proof. If x is contained in a finite component of $\mathscr{A}^0$ then the existence of s follows from the fact that $\mathscr{G}_n \cong \mathscr{A}^0$. Otherwise, there are $t \geqslant s > u$ such that $s+1$ is a σ-recovery stage, there are no σ-recovery stages in the interval $(s+1, t+1]$, x is contained in $(k)_t$, $k \in D_6$, and $(k)_t$ is involved in an operation at stage $t+1$. Now it follows from Lemma 2.49 that x is contained in $(k)_s$ and $\mathscr{G}_n[s]$ has a component $L \cong (k)_s$.

Let $s+1 = s_0 + 1 < s_1 + 1 < \cdots$ be the σ-recovery stages greater than or equal to $s+1$. Let L_i be the component of $\mathscr{G}_n[s_i]$ that extends L and let L' be the component of $\mathscr{G}_n$ that extends L. Using Lemma 2.51 and induction, we see that, for each $i \geqslant 0$, there exists a unique isomorphism $g_i : (k)_{s_i} \cong L_i$. Furthermore, if $j > i$ then g_j extends g_i. Thus the limit g' of the g_i is well-defined and is an isomorphism from (k) to L'. By the uniqueness of f, $f(x) = g'(x) = g_0(x) = g(x)$. □

Lemma 2.53. *$f \upharpoonright P_{D_6}$ is computable.*

Proof. Let u be a stage after which σ is never initialized and such that, for all $s \geqslant u$, $r_{\sigma,s} = 0$. Given $x \in P_{D_6}$, find the least σ-recovery stage $s+1 \geqslant u$ such that x is contained in a component $(m)_s$, $m \in D_6$, of $\mathscr{A}^0_s$ and there exists a component L of $\mathscr{G}_n[s]$ isomorphic to $(m)_s$. Such a stage exists by Lemma 2.52. Let g_x be the unique isomorphism from $(m)_s$ to L. Again by Lemma 2.52, $f(x) = g_x(x)$. Since g_x can be computably determined given $x \in P_{D_6}$, $f \upharpoonright P_{D_6}$ is computable. □

By Lemmas 2.41, 2.43, 2.45–2.48, and 2.53, $f \upharpoonright P_{D_i}$ is computable for each $i \leqslant 6$. As can be easily checked by referring to Table 1, $D_0, \ldots, D_6$ are computable and $\bigcup_{i=0}^{6} D_i = \omega$. Thus, by Lemma 2.38, we have the following result.

Lemma 2.54. *The unique isomorphism $f : \mathscr{A}^0 \cong \mathscr{G}_n$ is computable.*

Theorem 1.10 follows from Lemmas 2.9, 2.10, 2.32, 2.34, and 2.54. □

3. Proof of Theorem 1.12

In this section we prove the following theorem, using a construction similar to that of Section 4 of [18].

Theorem 1.12. *Let $\alpha \in \omega \cup \{\omega\}$ and let $\mathbf{b} > \mathbf{0}$ be an α-c.e. degree. There exists an intrinsically α-c.e. relation V on the domain of a computable structure $\mathscr{B}$ of computable dimension 2 such that $\mathrm{DgSp}_{\mathscr{B}}(V) = \{\mathbf{0}, \mathbf{b}\}$. In addition, $\mathscr{B}$ can be picked so that every c.e. presentation of $\mathscr{B}$ is computable, which implies that $\mathscr{B}$ has c.e. dimension 2.*

Proof. Let $\alpha \in \omega \cup \{\omega\}$ and let B be an α-c.e. set that is not computable. It follows immediately from Definition 1.11 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 $\mathscr{B}^0$ and $\mathscr{B}^1$ and unary relations V^0 and V^1 on the domains of $\mathscr{B}^0$ and $\mathscr{B}^1$, respectively, so that the following properties hold:

(3.1) $\mathscr{B}^0 \cong \mathscr{B}^1$ via an isomorphism that carries V^0 to V^1.
(3.2) $V^0 \equiv_{\mathrm{m}} B$ and V^1 is computable.
(3.3) If $\mathscr{G} \cong \mathscr{B}^0$ is a computable structure then $\mathscr{G}$ is computably isomorphic to either $\mathscr{B}^0$ or $\mathscr{B}^1$.
(3.4) $\mathscr{B}^0$ is rigid.
(3.5) Every c.e. presentation of $\mathscr{B}^0$ with computable equality relation is computable.

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 $\mathscr{A}^0$ and $\mathscr{A}^1$ and relations U^0 and U^1 on the domains of $\mathscr{A}^0$ and $\mathscr{A}^1$, respectively, satisfying properties (2.1)–(2.5). (We assume that the construction has been carried out in such a way that the domains of $\mathscr{A}^0$ and $\mathscr{A}^1$ are co-infinite.)

Now, for $i = 0, 1$, proceed as follows. Add an element, which we will call the *identifying node* of $\mathscr{B}^i$, to the domain of $\mathscr{A}^i$ and add an edge from this node to each node of $\mathscr{A}^i$. 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 $\mathscr{A}^i$, add an edge from x to the coding location of the copy of $[6\langle j, k \rangle]$ in L_k for each $k < f(j)$, and add an edge from x to the identifying node of $\mathscr{B}^i$. The resulting graph is $\mathscr{B}^i$.

Clearly, we can build each $\mathscr{B}^i$ so that it is a computable graph, and the following lemma can be easily checked, using the fact that $\mathscr{A}^0$ is rigid.

Lemma 3.1. *$\mathscr{B}^0$ is rigid.*

It is also not hard to establish that (3.5) holds.

Lemma 3.2. *If $\mathscr{G}$ is a c.e. presentation of $\mathscr{B}^0$ with computable equality relation then $\mathscr{G}$ is computable.*

Proof. Let z be the image of the identifying node of $\mathscr{B}^0$ in $\mathscr{G}$. Let $\mathscr{G}'$ be the subgraph of $\mathscr{G}$ consisting of all elements y of $\mathscr{G}$ such that there is an edge from z to y, and let $\mathscr{G}''$ be the subgraph of $\mathscr{G}$ consisting of all elements y of $\mathscr{G}$ such that there is an edge from y to z. Since $|\mathscr{G}'| \cap |\mathscr{G}''| = \emptyset$ and $|\mathscr{G}'| \cup |\mathscr{G}''| \cup \{z\} = |\mathscr{G}|$, both $|\mathscr{G}'|$ and $|\mathscr{G}''|$ are computable. Since $\mathscr{G}' \cong \mathscr{A}^0$, it follows from (2.5) that $\mathscr{G}'$ is computable.

For each element $x \in \mathcal{G}$, there is either an edge from z to x or an edge from x to z, but not both. Furthermore, there are no edges between elements of $\mathcal{G}''$ or from an element of $\mathcal{G}'$ to an element of $\mathcal{G}''$. Thus it suffices to show that there is an effective procedure for determining, given $x \in \mathcal{G}''$ and $y \in \mathcal{G}'$, whether there is an edge from x to y.

Fix x and y as above. Then x is a j-coding node for some $j \in \omega$, and this j can be found effectively. There are exactly $f(j)$ many elements $w \neq z$ of $\mathcal{G}$ for which there is an edge from x to w. Since f is computable, we can find these elements and check whether y is among them. □

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 $V^i = \{x \in K^i \mid c^i(x) \text{ is odd}\}$.

Lemma 3.3. *$\mathcal{B}^0 \cong \mathcal{B}^1$ 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 $c^0(x) = c^1(h(x))$, which in turn implies that $h(V^0) = V^1$. □

Lemma 3.4. *V_1 is computable and $V_0 \equiv_m B$.*

Proof. Since U^1 is computable, there is a computable procedure for determining $c(x)$ given $x \in K^1$, and thus V_1 is computable.

Let $x \in K^0$. By construction, there exist components $L_0, \ldots, L_{f(j)-1}$ of $\mathcal{A}^0$ 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 $d(x)$ be the least k such that, for all $m \geqslant k$, y_m is the coding location of the copy of $[6\langle j,m\rangle]$ in $\mathcal{A}_0^0$, if such a k exists, and let $d(x) = f(j)$ otherwise. Note that there is a computable procedure for determining $d(x)$ given $x \in K^0$.

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 $y_k \in U^0$ for $k < d(x)$. So if we define $S = \{x \in K^0 \mid c(x) < d(x)\}$ and $T = K^0 - S$ then S, T, and $V^0 \cap S$ 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, $y_0, \ldots, y_{d(x)-1} \in U^0$, so $\langle j,k\rangle \in A$ for all $k < d(x)$. But, by the definition of $d(x)$, for each $k \geqslant d(x)$, $y_k \in U^0$ 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 $x \in V^0$ if and only if $j \in B$, and hence $V^0 \cap T \equiv_m B$. Since $V^0 = (V^0 \cap S) \cup (V^0 \cap T)$, it follows that $V^0 \equiv_m B$. □

Lemma 3.5. *If $\mathcal{G} \cong \mathcal{B}^0$ is a computable structure then $\mathcal{G}$ is computably isomorphic to either $\mathcal{B}^0$ or $\mathcal{B}^1$.*

Proof. Let z be the image of the identifying node of $\mathscr{B}^0$ in $\mathscr{G}$. Let $\mathscr{G}'$ be the computable subgraph of $\mathscr{G}$ consisting of all elements y of $\mathscr{G}$ such that there is an edge from z to y. By the definition of $\mathscr{B}^0$, $\mathscr{G}' \cong \mathscr{A}^0$. Thus, by (2.3), there exists a computable isomorphism $h : \mathscr{A}^i \cong \mathscr{G}'$ for some $i \leqslant 1$.

To extend this isomorphism to a computable isomorphism $\hat{h} : \mathscr{B}^i \cong \mathscr{G}$, we first define $\hat{h} \upharpoonright \mathscr{A}^i \equiv h$ and $\hat{h}(u) = z$, where u is the identifying node of $\mathscr{B}^i$. Now let $x \in \mathscr{B}^i - (\mathscr{A}^i \cup \{u\})$. Then x is a j-coding node for some $j \in \omega$, and we can computably determine the $f(j)$ many coding locations $y_0, \ldots, y_{f(j)-1}$ attached to x. There is a unique $w \in \mathscr{G} - \mathscr{G}'$ attached to $h(y_0), \ldots, h(y_{f(j)-1})$. Define $\hat{h}(x) = w$. It is now easy to check that $\hat{h}$ is a computable isomorphism from $\mathscr{B}^i$ to $\mathscr{G}$. □

Theorem 1.12 follows from Lemmas 3.1–3.5. □

References

[1] C.J. Ash, Isomorphic recursive structures, in: Ershov et al. (Eds.), Handbook of Recursive Mathematics, Studies in Logic and the Foundations of Mathematics, vols. 138–139, Elsevier Science, Amsterdam, 1998, pp. 167–182.

[2] C.J. Ash, P. Cholak, J.F. Knight, Permitting, forcing, and copying of a given recursive relation, Ann. Pure Appl. Logic 86 (1997) 219–236.

[3] C.J. Ash, 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, pp. 26–41.

[4] E. Barker, Intrinsically Σ^0_α relations, Ann. Pure Appl. Logic 39 (1988) 105–130.

[5] P. Cholak, S.S. Goncharov, B. Khoussainov, R.A. Shore, Computably categorical structures and expansions by constants, J. Symbolic Logic 64 (1999) 13–37.

[6] R.G. Downey, Computability theory and linear orderings, in: Ershov et al. (Eds.), Handbook of Recursive Mathematics, Studies in Logic and the Foundations of Mathematics, vols. 138–139, Elsevier Science, Amsterdam, 1998, pp. 823–976.

[7] R.L. Epstein, R. Haas, R.L. Kramer, Hierarchies of sets and degrees below $\mathbf{0}'$, in: M. Lerman, J.H. Schmerl, R.I. Soare (Eds.), Logic Year 1979–80, Proc. Seminars and Conf. Math. Logic, Univ. Connecticut, Storrs, Conn., 1979/80, Lecture Notes in Mathematics, vol. 859, Springer, Heidelberg, 1981, pp. 32–48.

[8] Y.L. Ershov, S.S. Goncharov, Elementary theories and their constructive models, in: Ershov et al. (Eds.), Handbook of Recursive Mathematics, Studies in Logic and the Foundations of Mathematics, vols. 138–139, Elsevier Science, Amsterdam, 1998, pp. 115–166.

[9] Y.L. Ershov, S.S. Goncharov, A. Nerode, J.B. Remmel (Eds.), Handbook of Recursive Mathematics, Studies in Logic and the Foundations of Mathematics, vols. 138–139, Elsevier Science, Amsterdam, 1998.

[10] S.S. Goncharov, Computable single-valued numerations, Algebra and Logic 19 (1980) 325–356.

[11] S.S. Goncharov, Problem of the number of non-self-equivalent constructivizations, Algebra and Logic 19 (1980) 401–414.

[12] S.S. Goncharov, Autostable models and algorithmic dimensions, in: Ershov et al. (Eds.), Handbook of Recursive Mathematics, Studies in Logic and the Foundations of Mathematics, vols. 138–139, Elsevier Science, Amsterdam, 1998, pp. 261–288.

[13] S.S. Goncharov, B. Khoussainov, On the spectrum of degrees of decidable relations, Dokl. Math. 55 (1997) 55–57, research announcement.

[14] V.S. Harizanov, Degree spectrum of a recursive relation on a recursive structure, Ph.D. Thesis, University of Wisconsin, Madison, WI, 1987.

[15] 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.

[16] V.S. Harizanov, Pure computable model theory, in: Ershov et al. (Eds.), Handbook of Recursive Mathematics, Studies in Logic and the Foundations of Mathematics, vols. 138–139, Elsevier Science, Amsterdam, 1998, pp. 3–114.
[17] V.S. Harizanov, Turing degrees of certain isomorphic images of computable relations, Ann. Pure Appl. Logic 93 (1998) 103–113.
[18] D.R. Hirschfeldt, Degree spectra of intrinsically c.e. relations, J. Symbolic Logic, to appear.
[19] D.R. Hirschfeldt, B. Khoussainov, R.A. Shore, A.M. Slinko, Degree spectra and computable dimension in algebraic structures, Ann. Pure Appl. Logic, to appear.
[20] B. Khoussainov, R.A. Shore, Computable isomorphisms, degree spectra of relations, and Scott families, Ann. Pure Appl. Logic 93 (1998) 153–193.
[21] B. Khoussainov, R.A. Shore, Effective model theory: the number of models and their complexity, in: S.B. Cooper, J.K. Truss (Eds.), Models and Computability, London Mathematical Society Lecture Note Series, vol. 259, Cambridge University Press, Cambridge, 1999, pp. 193–239.
[22] B. Khoussainov, R.A. Shore, Solution of the Goncharov–Ash problem and the spectrum problem in the theory of computable models, Dokl. Math. 61 (2000) 178–179, research announcement (Russian version in Dokl. Akad. Nauk 371 (2000) 30–31).
[23] J.B. Remmel, Recursive isomorphism types of recursive Boolean algebras, J. Symbolic Logic 46 (1981) 572–594.
[24] R.I. Soare, Recursively Enumerable Sets and Degrees, Perspectives in Mathematical Logic, Springer, Heidelberg, 1987.