Computable Isomorphisms, Degree Spectra of Relations, and Scott Families

Bakhadyr Khoussainov*
Cornell University, Ithaca NY 14850 USA
The University of Auckland, Auckland, New Zealand

Richard A. Shore§
Cornell University, Ithaca NY 14853 USA

1 Introduction

In studying effective structures we investigate the effective content of typical notions and constructions in many branches of mathematics including universal algebra and model theory. In particular, we are interested in the possibilities of effectivizing model–theoretic or algebraic constructions and the limits on these possibilities. For instance, we try to understand whether certain results of model theory (or universal algebra) can be carried out effectively. If not, we then try to discover sharp effective counterexamples.

The systematic study of effectiveness in algebraic structures goes back to pioneering papers by Frölich and Shepherdson [11], Malcev [31][32], and Rabin [37] in the early 60s. Later in the early 70s, Nerode and his collaborators initiated combining algebraic constructions with priority arguments from computability theory thus beginning a new era in the development of the subject.

Nowadays, there various approaches to effectiveness in structures. For example, Cenzer, Nerode, Remmel have been developing theory of p-time

*Partially supported by ARO through MSI, Cornell University, DAAL-03-C-0027.
§Partially supported by NSF Grant DMS-9503503 and ARO through MSI, Cornell University, DAAL-03-C-0027.

structures [6]. Khoussainov and Nerode have began the development of the theory of automatic structures [29]. In this paper we are interested in those structures in which the basic computations can be performed by Turing machines.

Definition 1.1 *A structure $\mathcal{A}$ for a language $\mathcal{L}$ is* **computable** *(***decidable***) if the domain A of $\mathcal{A}$ is computable and there is a computable enumeration a_i of A such that the atomic (elementary) diagram of $(\mathcal{A}, \mathbf{a}_i)$ is computable. A structure isomorphic to a computable (decidable) structure is called* **(decidably) computably presentable***. Any isomorphism from a structure $\mathcal{A}$ to a computable (decidable) structure is called a* **computable (decidable) presentation** *of $\mathcal{A}$.*

Here are some examples of computable or computably presentable structures. The natural numbers with addition and multiplication is a computable structure. The ordered field of rational numbers, finitely presented algebras with decidable word problem, free groups, vector spaces over the field of rationals, etc. all have computable presentations.

We also give another, more general, notion of effective structure.

Definition 1.2 *A structure $\mathcal{A}$ for a language $\mathcal{L}$ is* **computably enumerable (c.e.)** *if there is a computable enumeration a_i of A such that the positive atomic diagram of $(\mathcal{A}, \mathbf{a}_i)$ is computably enumerable. A structure isomorphic to a computably enumerable structure is called* **c.e. (computably enumerably) presentable***. Any isomorphism from a structure $\mathcal{A}$ to a computably enumerable structure is called a* **computably enumerable (c.e.) presentation** *of $\mathcal{A}$.*

Finitely presented algebras are natural examples of c.e. structures. Note that every computable structure is also computably enumerable.

In model theory and universal algebra we identify isomorphic structures. When introducing effectiveness consideration into the area, we naturally want to understand the relationship between classical isomorphism types and effective isomorphism types. Thus, while model theory and universal algebra identify isomorphic structures, effective model theory is concerned with computable isomorphisms and finding characterizations for structures which have the same computable isomorphism type.

Definition 1.3 *Two computable (c.e.) structures $\mathcal{A}$ and $\mathcal{B}$ are of the same* **computable (c.e.) isomorphism type** *if there is a computable isomorphism taking $\mathcal{A}$ to $\mathcal{B}$. The* **(c.e.) dimension** *of a structure $\mathcal{A}$ is the number of its computable (c.e.) isomorphism types.*

Thus the dimension of a structure is the maximal number of noncomputably isomorphic computable (c.e.) presentations of the structure. We would like to stress that the dimension of a structure in the class of computable presentations can differ from the dimension of the structure in the class of c.e. presentations. For example, the dimension of $(Q, \leq)$, where Q is the set of all rationals, is 1 in the class of computable presentation. However, one can show that the dimension of this linear ordering is ω in the class of all c.e. presentations. In this paper we deal mainly with computable presentations. Therefore dimensions are considered with respect to the class of computable presentations of structures unless we specify otherwise. Similarly, all structures considered will be computable unless otherwise mentioned.

How far computable isomorphism types can be from classical ones can be seen in the following result of Goncharov:

Theorem 1.4 *([14]) For each $n \leq \omega$ there is a computable structure with computable dimension n.*

There has been a significant interest in trying to understand the nature of the structures of dimension 1. The basic model-theoretic notion which motivated the consideration of structures of dimension 1 is the notion of countable categoricity. A theory T is countably categorical if all (computable and noncomputable) countable models of T are isomorphic. An arbitrary countable structure $\mathcal{A}$ is categorical if its theory is countably categorical. The analogous concept for effective model theory deals only with computable structures and isomorphisms:

Definition 1.5 *A structure $\mathcal{A}$ is* **computably categorical** *if every computable structure isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}$.*

The following result of Nurtasin was one of the first results about the nature of computably categorical structures:

Theorem 1.6 *([36]) For a structure $\mathcal{A}$ the following two conditions are equivalent:*

1. *Any two decidable presentations of $\mathcal{A}$ are computably isomorphic.*

2. *There exists a finite number of constants $\bar{c} \in A$ such that $(\mathcal{A}, \bar{c})$ is the prime model of the theory $Th(\mathcal{A}, \bar{c})$ and the set of atoms of this theory is computable.*

In the late 70's, Goncharov–Dzgoev and Remmel independently gave an algebraic characterization of computably categorical Boolean Algebras and Linear Orderings.

Theorem 1.7 *([38] [13] [17])*

1. *A Boolean Algebra is computably categorical if and only if it has finitely many atoms.*

2. *A linear ordering is computably categorical if and only if the number of adjacent pairs in the ordering is finite.*

In [15] Goncharov proved that if a structure $\mathcal{A}$ has two presentations, one of them computable but not decidable and the other decidable, then the dimension of the structure is ω. In a conversation with Khoussainov, Goncharov asked if an analogous result can be obtained to answer the following question:

Question 1.8 *If a structure $\mathcal{A}$ has two presentations one of them computably enumerable but not computable and the other computable, is the dimension of the structure ω in the class of c.e. presentations?*

We answer this question in Theorem 4.1, using the proof of our basic result, Theorem 2.1:

Theorem 4.1 *There exists a structure $\mathcal{B}$ which has exactly two computably enumerable presentations $\mathcal{B}_1$ and $\mathcal{B}_2$ such that the following properties hold:*

1. *$\mathcal{B}_1$ is a computably enumerable but not computable structure.*

2. *$\mathcal{B}_2$ is a computable structure.*

3. *Any computably enumerable presentation $\mathcal{C}$ of $\mathcal{B}$ is computably isomorphic to either $\mathcal{B}_1$ or $\mathcal{B}_2$.*

Interestingly, all the structures which have been shown to be computably categorical have one common property. They all have Scott families.

Definition 1.9 *A* **Scott family** *for a structure $\mathcal{A}$ is a computable sequence*

$$\phi_0(\bar{a}, x_1, \ldots, x_{n_0}), \phi_1(\bar{a}, x_1, \ldots, x_{n_1}), \ldots,$$

of $\exists$-formulas satisfiable in $\mathcal{A}$, where $\bar{a}$ is a finite tuple of elements from $\mathcal{A}$, such that every n-tuple of elements from $\mathcal{A}$ satisfies one these formulas and any two tuples satisfying the same formula from the above sequence can be interchanged by an automorphism of $\mathcal{A}$.

The basic idea behind this definition is the following. If $\mathcal{A}$ has a Scott family and $\mathcal{B}$ is a computable structure isomorphic to $\mathcal{A}$, then the existence of the Scott family allows one to effectively carry out a back and forth argument to construct a computable isomorphism from $\mathcal{A}$ to $\mathcal{B}$. Therefore:

Theorem 1.10 *If a structure $\mathcal{A}$ has a Scott family, then $\mathcal{A}$ is computably categorical. Moreover, for any n-tuple $(c_1, \ldots, c_n)$ from $\mathcal{A}$, the expanded structure $(\mathcal{A}, c_1, \ldots, c_n)$also has a Scott family and hence is computably categorical.*

Proof. Let $\phi_0(\bar{a}, x_1, \ldots, x_{n_0}), \phi_1(\bar{a}, x_1, \ldots, x_{n_1}), \ldots$ be a Scott family for $\mathcal{A}$, where $\bar{a} = (a_0, \ldots, a_{m-1})$. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be computable presentations of $\mathcal{A}$. We define a mapping $f : A_1 \to A_2$ by stages. We can assume that for each $j \in \{0, \ldots, m-1\}$, a^i_j is the element in A_i corresponding to the constant a_j under a classical isomorphism between $\mathcal{A}_1$ and $\mathcal{A}_2$. At even stages we define images of elements from $\mathcal{A}_1$, at odd stages we define preimages of elements from $\mathcal{A}_2$.

Stage 0. Set $f_1 = \{(a^1_0, a^2_0), \ldots, (a^1_{m-1}, a^2_{m-1})\}$.

Stage 2k. We can suppose that the function f_{2k-1} has been defined. Assume that $f_{2k-1} = \{(a^1_0, a^2_0), \ldots, (a^1_{m-1}, a^2_{m-1}), (b_1, d_1), \ldots, (b_s, d_s)\}$ and that f_{2k-1} can be extended to an isomorphism from $\mathcal{A}_1$ to $\mathcal{A}_2$. Let b be the first number not in the domain of f_{2k-1}. Consider the tuple $(b_1, \ldots, b_s, b)$. Find an i such that $\phi_i(\bar{a}, b_1, \ldots, b_s, b)$ holds in $\mathcal{A}_1$. Hence $\exists x \phi_i(\bar{a}, d_1, \ldots, d_s, x)$ holds in $\mathcal{A}_2$. Find the first $d \in A_2$ for which $\phi_i(\bar{a}, d_1, \ldots, d_s, d)$ holds. Extend f_{2k-1} by letting $f_{2k} = f_{2k-1} \bigcup \{(b, d)\}$.

Stage 2k+1. We define f_{2k+1} similarly so as to put the least element of A_2 not yet in the range of f_{2k} into that of f_{2k+1}.

Finally, let $f = \bigcup_{i \in \omega} f_i$. Thus, f is a computable isomorphism.

For the second part of the theorem, we slightly change the original Scott family. Namely, set $\psi_i = \phi_i(\bar{a}, x_1, \ldots, x_{n_i}) \& \exists y_1 \ldots \exists y_n (\&_j (c_j = y_j)$. Then, one can easily check that the sequence $\psi_0, \psi_1, \ldots$ is a Scott family for the expanded structure $(\mathcal{A}, c_1, \ldots, c_n)$. $\Box$

At this point, we would like to make the following two observations about the effect of expanding computably categorical structures by finitely many constants. First, as we mentioned, all the known examples of computably categorical structures have Scott families. Therefore, it is natural to ask whether there exists a computably categorical structure without a Scott family. By the above theorem, one possible way to build a such structure is to provide an example of a computably categorical structure one of whose expansions by finitely many constants is not computably categorical. Second, one of the motivations of computable categoricity comes from the model-theoretic notion of ω–categoricity. It is an easy consequence of the Ryll-Nardzewski Theorem that if a structure $\mathcal{A}$ is countably categorical then so is the structure $(\mathcal{A}, \bar{a})$ expanded by finitely many constants. It is the analogous situation in effective model theory that we wish to consider.

Goncharov [12] proved that a computable structure whose universal-existential theory is decidable is computably categorical if and only if it has a Scott family. Thus any computably categorical structure with a decidable universal-existential theory remains computably categorical when expanded by finitely many constants. Millar proved that the decidability of the structures existential theory is actually sufficient to show that categoricity is preserved under such expansions:

Theorem 1.11 *([35]) If a structure $\mathcal{A}$ is computably categorical and its existential theory is decidable, then any expansion of $\mathcal{A}$ by finitely many constants is also computably categorical.*

Informally this theorem states that if we can effectively solve systems of algebraic equation and inequations over a computably categorical structure, then computable categoricity is preserved under expansions by a finite number of constants. However, without this partial decidability assumption the problem, known as Millar–Goncharov problem, had been open for some time.

Question 1.12 *[9] Is the expansion of every computably categorical structure by finitely many constants computably categorical?*

In [7] a negative solution to this problem is given. Our Theorem 4.2 (a corollary of an extension of our main result) gives a new solution which has a much simpler proof when $k \geq 3$:

Theorem 4.2 *For any natural number k there exists a computably categorical structure $\mathcal{A}$ such that, for each element $a \in A$, the expanded structure $(\mathcal{A}, a)$ has dimension k.*

An immediate consequence of Theorem 4.2 is the following:

Corollary 4.3 *There exists a computably categorical structure without a Scott family.*

Proof. Consider the structure $\mathcal{A}$ whose existence is claimed in Theorem 4.2. This structure is computably categorical. The structure does not have a Scott family as otherwise we would have a contradiction by Theorem 1.10. $\square$

Based on Theorem 1.10, one might suppose that the essential reason that the structure constructed in Theorem 4.2 does not have a Scott family is that its expansion by a finite number of constants is not computably categorical. Therefore one can ask the following question:

Question 1.13 *Does there exist a computably categorical structure without a Scott family whose expansion by any finite number of constants is computably categorical?*

One application of the ideas used in the proof of our basic result (Theorem 2.1) provides a positive answer to Question 1.13. Indeed, the construction used to prove Theorem 2.1 can be simplified to establish the following:

Theorem 4.4 *There exists a computably categorical structure without a Scott family whose expansion by any finite number of constants is computably categorical.*

We note that Kudinov [30] has also recently proven a similar result by quite different methods. He slightly modifies a family of computable enumerations constructed by Selivanov [44] and then codes the family as a unary algebra in such a way as to produce a computably categorical structure with a decidable existential theory but no Scott family. Combining this result with Theorem 1.11 gives our Theorem 4.4. (We thank the referee for directing us to this paper by Kudinov.)

Now we turn to our basic result, **Theorem 2.1**. This theorem is about the degree spectra of relations on structures.

Definition 1.14 *If U is a relation on a computable structure $\mathcal{A}$, we define the partial ordering*

$$DgSp(U) = (\{deg_T(U^{\mathcal{B}}) |\ \mathcal{B} \text{ is a computable presentation of } \mathcal{A}\}, \leq_T),$$

where $\leq_T$ is Turing reducibility on sets of natural numbers and $U_{\mathcal{B}}$ is the image of U in $\mathcal{B}$. We call this partially ordered set **the degree spectrum** *of U. (Recall that a computable presentation $\mathcal{B}$ of $\mathcal{A}$ includes an isomorphism from $\mathcal{A}$ to $\mathcal{B}$ and so determines $U^{\mathcal{B}}$.)*

There has been an extensive study of the degree spectra of relations on computably structures and related results. We refer readers to papers by Ash and Nerode [1], Ash and Knight [3] [4] [24] [25] [26], Harizanov [19] [20] [21], Ash, Cholak and Knight [5], etc.

Harizanov and Millar suggested investigating relations with finite degree spectra. Their motivations came from the fact that Ash-Nerode type decidability conditions on relations (see [1]) usually imply that the degree spectrum is infinite or a singleton. Another motivation is Goncharov's examples of structures with finitely many computable isomorphism types. Here is an open question which we call the **Degree Spectra Problem**:

Question 1.15 *Is every finite partially ordered set isomorphic to $DgSp(U)$ for some relation U on a structure $\mathcal{A}$?*

Modifying Goncharov's construction from [14], Harizanov was the first to provide examples of relations with finite degree spectra.

Theorem 1.16 *([20]) There exists a relation U on a structure $\mathcal{A}$ with exactly two computable presentations $\mathcal{A}_0$ and $\mathcal{A}_1$ such that $U^{\mathcal{A}_0}$ is computable and $U^{\mathcal{A}_1}$ is a noncomputable Δ^0_2–set.*

Our Theorem 2.1 improves this result:

Theorem 2.1 *There exists a relation U in a structure $\mathcal{A}$ which has exactly two computable presentations $\mathcal{A}_0$ and $\mathcal{A}_1$ such that $U^{\mathcal{A}_0}$ is c.e. but not computable while $U^{\mathcal{A}_1}$ is computable. Moreover, the relation $P = \{(x, y)|x \in U^{\mathcal{A}_0} \wedge y \in U^{\mathcal{A}_1} \wedge$ there is an isomorphism from $\mathcal{A}_0$ to $\mathcal{A}_1$ which extends the map $x \mapsto y\}$ is computable.*

Goncharov [18] has announced a construction of a computable structure with dimension two and a relation whose degree spectrum consists of $\mathbf{0}$ and one noncomputable c.e. degree. This work grew out of earlier work with Khoussainov and is based on the construction of a family of c.e. sets with certain properties. However, the construction does not supply any more and Goncharov poses what we have called the Degree Spectra Problem (Question 1.15) as Problem 2 of [18]. Our methods enable us to give a positive solution to this problem:

Theorem 3.1 *For any computable partially ordered set $\mathcal{D}$ there exists a structure $\mathcal{A}$ of dimension the cardinality of $\mathcal{D}$ and a relation U on $\mathcal{A}$ such that $DgSp(U) \cong \mathcal{D}$. Indeed, we can also guarantee that $U^{\mathcal{B}}$ is c.e. for every computable presentation $\mathcal{B}$ of $\mathcal{A}$ and that, if $\mathcal{D}$ has a least element, then the least element in $DgSp(U)$ is $\mathbf{0}$.*

Several natural strengthenings of this result can ruled out by the observation that, for a given relation U on a computable structure $\mathcal{A}$, the set $\{U^{\mathcal{B}}|$ $\mathcal{B}$ is a computable presentation of $\mathcal{A}\}$ is Σ^1_1 in U. Thus, there are countable partial orderings that cannot be realized in the c.e. degrees as the degree spectrum of any relation U on any computable structure $\mathcal{A}$. (Just consider one that is too complicated to be Σ^1_1.) Similarly, such a partial ordering with least element cannot be realized anywhere in the Turing degrees as the degree spectrum of a computable relation U on a computable structure $\mathcal{A}$. Nor can it be true that any finite set of degrees be realized as the degree spectrum of any relation U on a computable structure $\mathcal{A}$. Indeed, any degree spectrum containing both a hyperarithmetic degree and a nonhyperarithmetic degree is uncountable as any Σ^1_1 set with a nonhyperarithmetic member is uncountable. (See, for example, Sacks [43] III.6.) A very natural question is whether every c.e. degree can be realized (with $\mathbf{0}$) as a degree spectrum. Hirschfeldt [23] has just recently answered this question by adapting and extending the methods presented here to show that all n-tuples of c.e. degrees can be realized as degree spectra. Goncharov [18] asks as Problem 1 which pairs or more generally n-tuples of c.e. degrees can be so realized in a structure of dimension precisely 2 (for pairs) or n for n-tuples.

Finally, we note that all the structures considered in this paper are **directed graphs** and we use elementary notions from graph theory, such as for example, edge relation, component, path, connectedness, etc. We denote the edge relation in any graph G by $E(G)$. Definitions of these notions can be found in any basic text on graph theory.

2 The Basic Result and Construction

Our goal in this section is to prove the following:

Theorem 2.1 *There exists a relation U in a structure $\mathcal{A}$ with exactly two computable presentations $\mathcal{A}_0$ and $\mathcal{A}_1$ such that the image $U^{\mathcal{A}_0}$ of U in $\mathcal{A}_0$ is c.e. but not computable while its image $U^{\mathcal{A}_1}$ in $\mathcal{A}_1$ is computable. Moreover, the relation $P = \{(x, y) | x \in U^{\mathcal{A}_0} \wedge y \in U^{\mathcal{A}_1} \wedge$ there is an isomorphism from $\mathcal{A}_0$ to $\mathcal{A}_1$ which extends the map $x \mapsto y\}$ is computable.*

Proof. In order to describe the structure $\mathcal{A}$ we need some notation. For a natural number $n \geq 1$, consider the directed graph

$$(\{0, 1, \ldots, n+1\}, E)$$

such that $E(0, 0)$, $E(n+1, 1)$, $E(1, 0)$, and $E(i, i+1)$ hold for all $i \in \{1, \ldots, n\}$. We denote this graph by $[n]$. Thus, we see that $[n]$ is a cycle of length $n+1$ with a tag. We call 0 the **top of** $[n]$ and $n+1$ the **coding location** of the graph $[n]$.

Let A be any set not containing 0. Consider a sequence $\{\mathcal{B}_n\}_{n\in A}$ of pairwise disjoint graphs such that each $\mathcal{B}_n$ is isomorphic to $[n]$. Define a graph by taking the disjoint union of the graphs $\mathcal{B}_n$ and **identifying** the top elements of these graphs. We denote this graph by $[A]$. Note that

$[n]$ is isomorphic to $[\{n\}]$. (Warning: We abuse notation by confusing the isomorphism type of $[A]$ and individual graphs isomorphic to $[A]$. Similarly, we leave the details of choosing computable representations of such graphs for computable sets A and associated operations on such representations (such as disjoint union) to the reader.)

Let $[A]$ and $[B]$ be graphs with disjoint domains. Suppose that the set $C = B \setminus A$ is nonempty. Let a and c be the top elements of $[A]$ and $[C]$, respectively. Suppose also that the domain of $[A]$ has empty intersection with the domain of $[C]$. We define a new graph $[A] \cdot [B]$ as follows:

1. The domain of $[A] \cdot [B]$ consists of all elements from $[A]$ or $[C]$ except for the top element of $[C]$.

2. The edge relation E is defined as follows. The pair (x, y) belongs to E if and only if

   $$[(x, y) \in E([A]) \bigvee ((x, y) \in E([C]) \,\&\, y \neq c)] \bigvee (y = a \,\&\, (x, c) \in E([C])).$$

For example, if G_1 is isomorphic to $[\{3, 4\}]$ and G_2 is isomorphic to $[\{4, 6\}]$, then $G_1 \cdot G_2$ is isomorphic to $[\{3, 4, 6\}]$. Note that $[A] \cdot [B]$ is isomorphic to $[B] \cdot [A]$.

On the set of all graphs we also consider the operation $+$ which, when applied to two graphs, produces a graph isomorphic to the disjoint union of the graphs. The operation $+$ can be extended as follows. Let $\mathcal{G}_1, \ldots, \mathcal{G}_n, \ldots$ be a sequence of graphs with disjoint domains. Then

$$\mathcal{G}_1 + \cdots + \mathcal{G}_n + \cdots \tag{1}$$

is, by definition, the graph whose domain G is $\bigcup G_i$ and the edge relation $E(G)$ is $\bigcup_i E(G_i)$. Note that the components of a graph of the form

$$[B_1] + \cdots + [B_n] + \cdots$$

are $[B_1], [B_2], \ldots$.

We want to produce two computable presentations $\mathcal{A}_0, \mathcal{A}_1$ of a graph $\mathcal{A}$ of the form

$$[B_1] + [B_2] + \cdots + [B_n] + \cdots$$

with $B_i \setminus B_j \neq \emptyset$ for all $i \neq j$. We will, in fact, also guarantee that this property holds at every stage. More precisely, at stage t the presentations $\mathcal{A}_{k,t}$ (for $k = 0, 1$) will be of the form $[B_{1,t}] + [B_{2,t}] + \cdots + [B_{n,t}] + \cdots$ with $B_{i,t} \setminus B_{j,t} \neq \emptyset$ for all $i \neq j$. Note that since $B_i \setminus B_j \neq \emptyset$ for all $i \neq j$, the graph $\mathcal{A}$ is clearly rigid, that is, it has no nontrivial automorphisms. In addition, we also want to produce a unary relation U on $\mathcal{A}$ with the following properties:

1. Any element of U is a coding location.
2. The image $U^{\mathcal{A}_0}$ of U in $\mathcal{A}_0$ is computably enumerable but not computable.
3. The image $U^{\mathcal{A}_1}$ of U in $\mathcal{A}_1$ is computable.
4. Every computable copy of $\mathcal{A}$ is computably isomorphic to either $\mathcal{A}_0$ or $\mathcal{A}_1$.

Thus, the dimension of $\mathcal{A}$ is two and the degree spectrum, $DgSp(U)$, of U consists of exactly two c.e. degrees one computable and the other not. Since $U^{\mathcal{A}_1}$ is computable and $U^{\mathcal{A}_0}$ is not, the structures $\mathcal{A}_0$ and $\mathcal{A}_1$ are not computably isomorphic.

Another important aspect of our construction will provide a computable binary predicate P such that $P(x, y)$ holds if and only if x belongs to $U^{\mathcal{A}_0}$, y belongs to $U^{\mathcal{A}_1}$, and there exists an isomorphism from $\mathcal{A}_0$ to $\mathcal{A}_1$ extending the map $x \to y$. We will need this property to construct computably categorical structures whose expansions by constants are not computably categorical.

Let $G_0, G_1, G_2, \ldots$ be a standard enumeration of all partial computable directed graphs. We also consider a standard enumeration $\Psi_0, \Psi_1, \Psi_2, \ldots$ of all computable partial functions. In order to build a structure $\mathcal{A}$ and a unary relation U with the properties above we need to satisfy at least the following requirements for $e, j \in \omega, k \in \{0, 1\}$:

$$D_e : \ U^{\mathcal{A}_0} \neq \Psi_e$$

and

$$R_j \quad : \quad \text{If } G_j \text{ is isomorphic to } \mathcal{A} \text{ then } G_j \text{ is computably isomorphic to } \mathcal{A}_0 \text{ or } \mathcal{A}_1$$

As we wish to show that any computable graph G isomorphic to $\mathcal{A}$ is computably isomorphic to $\mathcal{A}_0$ or $\mathcal{A}_1$ and we know about the special form described above that $\mathcal{A}_k$ must have, we can limit ourselves to computable graphs G_j which are of this form: $G_j = [B_1] + [B_2] + \cdots + [B_n] + \cdots$ with $B_i \setminus B_j \neq \emptyset$ for all $i \neq j$. Indeed, we can require that this be true at every stage: $G_{j,t} = [B_{1,t}] + [B_{2,t}] + \cdots + [B_{n,t}] + \cdots$ with $B_{i,t} \setminus B_{j,t} \neq \emptyset$ for all $i \neq j$. Moreover, without loss of generality, we can require that the enumerations $G_{j,t}$ be such that for every component $[Y_t]$ of $G_{j,t}$ there is one $[X_t]$ of each $\mathcal{A}_{k,t}$ such that $Y_t \subseteq X_t$. (Only enumerate components in G_j when they are of the right form, i.e., a top with cycles attached (to which more cycles can be added later) and contained in one of $\mathcal{A}_{k,t}$. If there are components partially enumerated, do not allow any extensions until it is once again possible to make all the components distinct. This can be done so as to add any single desired element to G if it has the required form.) During the construction we will also need to make $U^{\mathcal{A}_1}$ computable. This will be achieved by effectively listing all the elements of $U^{\mathcal{A}_1}$ in strictly increasing order.

The action to meet requirement D_e is based on the operations on graphs that we now define.

Definition 2.2 *The* $\mathbf{L}$*–operation applied to* $[B_1]+\cdots+[B_n]$ *produces a graph denoted by* $\mathbf{L}([B_1],\ldots,[B_n])$ *which is isomorphic to*

$$[B_1]\cdot[B_2]+\cdots+[B_{n-1}]\cdot[B_n]+[B_n]\cdot[B_1].$$

We also adopt the important convention that the elements of the component $[B_i]$ *are the same ones in the corresponding graph in the component* $[B_i]\cdot[B_{i+1}]$ *of* $\mathbf{L}([B_1],\ldots,[B_n])$ *while those elements in the new graph corresponding to ones in* $[B_{i+1}]$ *of the original graph are new elements in* $[B_i]\cdot[B_{i+1}]$ *(with* 1 *for* $n+1$ *when* $i=n$*).*
The $\mathbf{R}$*–operation applied to* $[B_1]+\cdots+[B_n]$ *produces the graph* $\mathbf{R}([B_1],\ldots,[B_n])$ *isomorphic to*

$$[B_1]\cdot[B_n]+[B_1]\cdot[B_2]+\ldots++[B_{n-1}]\cdot[B_n].$$

We will apply an $\mathbf{R}$ *operation in the construction only when we also apply an* $\mathbf{L}$ *one and we also have the corresponding convention that the elements of the component* $[B_i]$ *are the same ones in the corresponding graph in the component* $[B_{i-1}]\cdot[B_i]$ *of* $\mathbf{R}([B_1],\ldots,[B_n])$ *while those elements in the new* *graph corresponding to ones in* $[B_i]$ *of the original graph are new elements in* $[B_{i-1}] \cdot [B_i]$ *(with* n *for* 0 *when* $i = 1$*).*

From this definition we obtain the following lemma.

Lemma 2.3 *For any graph* $[B_1]+\cdots+[B_n]$ *the graphs* $\mathbf{L}([B_1],\ldots,[B_n])$ *and* $\mathbf{R}([B_1],\ldots,[B_n])$ *are isomorphic and extend* $[B_1]+\cdots+[B_n]$*.* $\Box$

The **L** and **R**–operations can also be applied to some components of a graph G to obtain extensions of G. Indeed, suppose that we have a graph G of the form

$$[C_1]+[C_2]+\cdots+[B_1]+[B_2]+\cdots+[B_n]+\cdots+[C_k].$$

Then we can consider two extensions of G:

$$[C_1]+[C_2]+\cdots+\mathbf{L}([B_1],[B_2],\ldots,[B_n])+\cdots+[C_k]$$

and

$$[C_1]+[C_2]+\cdots+\mathbf{R}([B_1],[B_2],\ldots,[B_n])+\cdots+[C_k].$$

By Lemma 2.1 these two extensions of G are isomorphic.

Now we explain how to meet each of the requirements D_e or R_j separately. Here is a strategy to meet just one D_e. Begin by constructing structures $\mathcal{A}_0$ and $\mathcal{A}_1$ of type

$$[\{1\}]+[\{2\}]+[\{3\}]+\cdots+[\{3e+1\}]+[\{3e+2\}]+[\{3e+3\}]+\cdots$$

As soon as Ψ_e equals 0 on the number x which is the coding location in $[\{3e+2\}]$, the construction acts by extending structure $\mathcal{A}_0$ to the structure:

$$[\{1\}]+[\{2\}]+[\{3\}]+\cdots+\mathbf{L}([\{3e+1\}],[\{3e+2\}],[\{3e+3\}])+\cdots$$

and by extending structure $\mathcal{A}_1$ to the structure:

$$[\{1\}]+[\{2\}]+[\{3\}]+\cdots+\mathbf{R}([\{3e+1\}],[\{3e+2\}],[\{3e+3\}])+\cdots$$

The construction then puts x into $U^{\mathcal{A}_0}$ and puts its image y in $\mathcal{A}_1$ into $U^{\mathcal{A}_1}$. This action meets requirement D_e. Note that the number put into $U^{\mathcal{A}_1}$ is new by our convention on applying the **L** and **R** operations. The construction also puts (x, y) into P.

A strategy to meet just one R_j is as follows. Begin by constructing structures $\mathcal{A}_0$ and $\mathcal{A}_1$ of the type

$$[\{1\}] + [\{2\}] + [\{3\}] + \cdots + [\{3e+1\}] + [\{3e+2\}] + [\{3e+3\}] + \cdots$$

At stage 0 let the partial isomorphism from G_j to $\mathcal{A}_0$ to $\mathcal{A}_1$ be empty. At stage i wait until G_j provides components B_1, B_2 and B_3 isomorphic to $[\{3i+1\}]$, $[\{3i+2\}]$, and $[\{3i+3\}]$. As soon as G_j provides such components, say B_1, B_2, B_3, extend the previous partial isomorphism by adding partial isomorphisms from B_1, B_2, and B_3 to $[\{3i+1\}]$, $[\{3i+2\}]$, and $[\{3i+3\}]$, respectively. If G_j is isomorphic to the structure the construction is building, then the above strategy constructs a computable isomorphism from G_j to the structure

$$[\{1\}] + [\{2\}] + [\{3\}] \ldots [\{3e+1\}] + [\{3e+2\}] + [\{3e+3\}] + \ldots$$

However, a problem can arise even in the case when the construction tries to satisfy all requirements D_e and just one R_j simultaneously. Here is a brief and informal explanation of the problem. Suppose that the construction has acted to meet requirement R_j on the components $[\{3e+1\}]$, $[\{3e+2\}]$, and $[\{3e+3\}]$ and has not yet acted to meet D_e. In other words G_j has provided components, say B_1, B_2, B_3, isomorphic to $[\{3e+1\}]$, $[\{3e+2\}]$, and $[\{3e+3\}]$, respectively, and hence the construction has a partial isomorphism, say r_0, from G_j into $\mathcal{A}_0$ defined on this coding locations; in addition Ψ_e has not yet been defined on the coding location in $[\{3e+2\}]$. Suppose that at a later stage of the construction Ψ_e equals 0 on the coding location in $[\{3e+2\}]$. The construction acts to meet D_e using **R** and **L**–operations. However, the construction now cannot guarantee that r_0 can be extended to an isomorphism from G_j to $\mathcal{A}_0$. Indeed, suppose that at a later stage G_j provides components isomorphic to the components of $\mathcal{A}_0$ containing the coding location and that r_0 cannot be so extended. (Perhaps the component previously isomorphic to $[\{3e+2\}]$ is now isomorphic to $[\{3e+1\}] \cdot [\{3e+2\}]$ and so not to $[\{3e+2\}] \cdot [\{3e+3\}]$ which is the new isomorphism type of the component to which it had previously been isomorphic.) The construction now changes its mind and attempts to begin constructing a new isomorphism from G_j to $\mathcal{A}_1$. But, an action to meet another requirement $D_{e'}$ at some later stage can force the construction to change its mind again and begin constructing yet another isomorphism from G_j to $\mathcal{A}_0$, etc. Thus, one can see that infinitely many D_e's can potentially force the construction to change its mind infinitely many times, and hence, to not satisfy R_j.

Now, we define some of the basic notions of our construction. First, we present a module satisfying all requirements D_e and just one requirement $R = R_j$ for $G = G_j$, the j^{th} partial computable graph of the appropriate form. For $t \in \omega$, let G_t be the approximation to G at stage t. We always use k to be either 0 or 1, and let $k+1 = 1$ if $k = 0$ and $k+1 = 0$ if $k = 1$. This module will construct isomorphic structures $\mathcal{A}_0$ and $\mathcal{A}_1$, unary relations $U^{\mathcal{A}_0}$ and $U^{\mathcal{A}_1}$, and a binary predicate P. If G is isomorphic to $\mathcal{A}_0$, then G will be computably isomorphic to either $\mathcal{A}_0$ or $\mathcal{A}_1$. Our construction proceeds by stages. At stage t of the module, we use the following notions and terminology.

1. Finite Structures $\mathcal{A}_{0,t}$ and $\mathcal{A}_{1,t}$. These are approximations to the isomorphic structures $\mathcal{A}_0$ and $\mathcal{A}_1$ that the construction is building. That is, for $k = 0, 1$, we will have

$$\mathcal{A}_{k,t} \subset \mathcal{A}_{k,t+1} \text{ and } \mathcal{A}_k = \bigcup_t \mathcal{A}_{k,t}.$$

Therefore the structure $\mathcal{A}_k = \bigcup_t \mathcal{A}_{k,t}$ is computable. The structures $\mathcal{A}_{0,t}$ and $\mathcal{A}_{1,t}$ are isomorphic to each other. Each $\mathcal{A}_{k,t}$ is of the form

$$[A^0_{k,t}] + [A^1_{k,t}] + \ldots + [A^t_{k,t}],$$

where, for all $n \neq m$, $A^n_{k,t} \setminus A^m_{k,t} \neq \emptyset$. By construction, we also guarantee that $A^n_{k,t} \subset A^n_{k,t+1}$ and that at the end $\mathcal{A}$ is isomorphic to

$$[A^0_k] + [A^1_k] + \ldots + [A^n_k] + \ldots,$$

where each $A^m_k = \bigcup_t A^m_{k,t}$ and $A^m_k \setminus A^s_k \neq \emptyset$ for $s \neq m$.

2. To each requirement D_e the construction assigns a coding location a_e not in $U^{\mathcal{A}_0}$. One of the goals of the construction is to put a_e into $U^{\mathcal{A}_0}$ if $\Psi_e(a_e) = 0$. (Its image in $\mathcal{A}_1$ will go into $U^{\mathcal{A}_1}$ but a_e itself will not.) Each of these coding locations will be in a different component of $\mathcal{A}_{0,t}$. Success in such an attempt meets requirement D_e on the number put into $U^{\mathcal{A}_0}$.

3. Functions r^0_t and r^1_t. At stage t each r^k_t is a partial isomorphism from G_t into $\mathcal{A}_{k,t}$ one of which is the construction's (designated) isomorphism. The function r^k_t may extend its previous isomorphism r^k_{t-1}. If r^k_t does not extend the previous isomorphism, then we say that the construction **changes its (designated) isomorphism** from r^k_t to r^{k+1}_t.

4. The special G–component. The construction picks a component $S_t(G)$ of the structure G called the **special G–component**. If there exist infinitely many stages at which the construction changes its isomorphism, then the special component $S(G) = \bigcup_t S_t(G)$ becomes infinite, all components in $\mathcal{A}_0$ (and hence in $\mathcal{A}_1$) which can be embedded into $S(G)$ are finite, and therefore G is not isomorphic to $\mathcal{A}_0$. (We say a component $[Y]$ can be **embedded** into one $[X]$ just in case $Y \subseteq X$.) If, after some stage t_0, the construction never changes its (designated) isomorphism from r^k and G is isomorphic to $\mathcal{A}$, then G will be computably isomorphic to $\mathcal{A}_k$ via $r^k = \cup\{r^k_t | t \geq t_0\}$.

5. Special Components $[S]^k_t$, $k = 0, 1$. The construction ensures that $[S]^0_t$ is isomorphic to $[S]^1_t$. The special component $[S]^k_t$ is one of the components of $\mathcal{A}_{k,t}$, that is $[S]^k_t$ is one of the $[A^j_{k,t}]$ for some $j \leq t$. At stage t, the special G–component $S_t(G)$ in the structure G can be embedded into these components. Moreover, if R recovers at stage t (see the definition of recovery below), then these components satisfy the following properties.

1. If the construction does not change its isomorphism from r^k, then $[S]^k_{t-1}$ is a substructure of $[S]^k_t$ and, if it participated in an **L** or **R** operation at t, $[S]^{k+1}_t \bigcap [S]^{k+1}_{t-1} = \emptyset$.

2. If the construction changes its isomorphism from r^k to r^{k+1}, then $[S]^{k+1}_{t-1}$ is a substructure of $[S]^{k+1}_t$ and $[S]^k_{t-1} \bigcap [S]^k_t = \emptyset$.

3. If, after some stage t', the construction never changes its isomorphism from r^k, recovers infinitely many times, and G is isomorphic to $\mathcal{A}$, then the construction guarantees that the special G–component $S(G)$ is isomorphic to $\bigcup_{t>t'} [S]^k_t$.

6. Marking with $\square_w$ and Recovery. If, for a component X in $\mathcal{A}_{k,t}$, there exists a unique component Y in G_t such that Y can be embedded into X but not into any other component of $\mathcal{A}_k$, then we say that Y **is covered by** X, or equivalently, X **covers** Y. During the construction some components in $\mathcal{A}_{k,t}$ will be marked with a special symbol $\square_w$ called a **waiting mark**. We say that R **recovers at stage** t, or equivalently that stage t is a **recovery stage**, if, at the beginning of stage t, for each component X in $\mathcal{A}_{k,t}$ marked with a $\Box_w$, there exists a unique component Y such that X covers Y. We use the notion of recovery to show that if G is isomorphic to $\mathcal{A}$, then G is computably isomorphic to either $\mathcal{A}_0$ or $\mathcal{A}_1$. The idea is the following. Suppose that G is isomorphic to $\mathcal{A}$. By construction, each component X in $\mathcal{A}_{k,t}$ marked with a $\Box_w$ waits to cover a component in G. As soon as R recovers at a stage $t_1 \geq t$ and a unique component Y in G_{t_1} is found such that X covers Y, the construction defines an isomorphic embedding from Y to X and then attempts to guarantee that Y is isomorphic to X. If R does not recover at stage t, then we say that R is in the **waiting state**. If R is always in the waiting state after stage t, then, by construction, G will not be isomorphic to $\mathcal{A}_k$.

In the general construction all the objects, as for example coding locations a_e, functions r_t^k, waiting mark $\Box_w$, etc., defined above will have additional indices which correspond to the nodes of the priority tree. Now we present a module to satisfy R and all requirements D_e.

Stage 0**.** Let $\mathcal{A}_{0,0}$ and $\mathcal{A}_{1,0}$ be isomorphic to the graph [2]. Let the partial isomorphisms r_0^0 and r_0^1 be empty with r_0^0 the designated partial isomorphism. Put R into the waiting state and mark the components of $\mathcal{A}_{0,0}$ and $\mathcal{A}_{1,0}$ with $\Box_w$. When we first have a recovery stage t we will define $S_t(G)$ so that $S_t(G)$, $[S]_t^0$ and $[S]_t^1$ are isomorphic to the graph [2].

Stage $t+1$: Substage 1: If this is not a recovery stage we go on to substage 2. If it is a recovery stage, proceed as follows:

Action. Define the partial isomorphisms r_{t+1}^0 (and r_{t+1}^1) on the components marked with a $\Box_w$: Partial isomorphism r_{t+1}^0 (r_{t+1}^1) maps a component Y in G_t into a component X in $\mathcal{A}_{0,t}$ $(\mathcal{A}_{1,t})$ if and only if Y is covered by X. Suppose r_t^k was the construction's previously designated isomorphism. We now have the following two cases.

Case 1. Suppose that r_{t+1}^k extends r_t^k. In this case, set the special component in $\mathcal{A}_{k,t+1}$ to be $[S]_{t+1}^k$. This special component extends the special component $[S]_t^k$ of the previous stage. [[In this case, $[S]_t^{k+1} \bigcap [S]_{t+1}^{k+1} = \emptyset$ if it participated in an **L** or **R** operation at t.]]

Case 2. Suppose that r^k_{t+1} does not extend r^k_t. In this case, the construction changes its designated partial isomorphism to r^{k+1}_{t+1}. Set the special component $[S]^{k+1}_{t+1}$ in $\mathcal{A}_{k+1,t+1}$ to be the image of the special G–component $S_t(G)$ under the partial isomorphism r^{k+1}_{t+1}. [[In this case, $[S]^k_t \bigcap [S]^k_{t+1} = \emptyset$.]]

Substage 2: Extend both $\mathcal{A}_{0,t}$ and $\mathcal{A}_{1,t}$ by adding a new component isomorphic to the graph $[a]$, where $a \geq 2$ is a new number. This extends $\mathcal{A}_{0,t}$ and $\mathcal{A}_{1,t}$. Let a_{t+1} be the coding location in the component isomorphic to $[a]$ in $\mathcal{A}_{0,t+1}$. Find the least $e \leq t+1$ for which $\Psi_{e,t+1}(a_e) = 0$ such that we have not yet acted for D_e, a_e is not in $U^{\mathcal{A}_0}$ and one of the following conditions is satisfied. (If there is no such e, then go to Substage 3.)

1. There does not exist a z such that $r^k_{t+1}(z) = a_e$.

2. This is a recovery stage and, for some z, $r^k_{t+1}(z) = a_e$.

Action for D_e. Suppose that the first condition holds. Let X be the component in $\mathcal{A}_{0,t}$ containing the coding location a_e. Let X' be the isomorphic image of X in $\mathcal{A}_{1,t}$. Extend $\mathcal{A}_{0,t}$ and $\mathcal{A}_{1,t}$ by performing operations $\mathbf{L}(Y, X, Z)$ in $\mathcal{A}_{0,t+1}$ and $\mathbf{R}(Y', X', Z')$ in $\mathcal{A}_{1,t+1}$, with Y, Z and Y', Z' new components isomorphic to $[y]$, $[z]$, respectively, where y, z are new numbers. Put a_e into $U^{\mathcal{A}_0}$ and its image v in $\mathbf{R}(Y', X', Z')$ (which is a new number and hence greater then all the numbers in $U^{\mathcal{A}_1}$ up to this stage) into $U^{\mathcal{A}_1}$. Put (a_e, v) into P. Note that we have successfully met requirement D_e. Also the structures $\mathcal{A}_{0,t}$ and $\mathcal{A}_{1,t}$ have been extended to new isomorphic structures $\mathcal{A}_{0,t+1}$ and $\mathcal{A}_{1,t+1}$, respectively. Now go to substage 3.

Suppose that the second condition holds. Consider the special components $[S]^0_t$, $[S]^1_t$, $S_t(G)$ in $A_{0,t}$, $\mathcal{A}_{1,t}$, and G_t, respectively. We have two subcases.

Subcase 2.1. There is a sequence of components $B^0_2, X^0, C^0_2, B^0_1, [S]^0_t, C^0_1$ in $\mathcal{A}_{0,t}$ and of their isomorphic images $B^1_2, X^1, C^1_2, B^1_1, [S]^1_t, C^1_1$ in $\mathcal{A}_{1,t}$ such that the following conditions hold:

1. The coding location a_e is in X^0.

2. The components B^k_2 and C^k_2 have never before participated in any operation.

3. For each $k \in \{0,1\}$, the image of r^k_{t+1} has nonempty intersection with every component in the corresponding sequence.

4. If $[S]^0_t$ and $[S]^1_t$ have participated in an **L**– or **R**–operation (respectively) an even number of times (possibly 0) since the construction last changed its isomorphism (e.g., it just changed it now) then the components B^k_1, C^k_1 have never participated in any **L**– or **R**–operations.

5. If $[S]^0_t$ has participated in an **L**-operation an odd number of times since the construction last changed its isomorphism and the designated isomorphism is now r^0_t, then C^0_1 is the component that played the role of B^0_1 the last time a special component participated in an **L**–operation and B^0_1, B^1_1 have never participated in any **L**– or **R**–operations. (C^1_1 is the component in $\mathcal{A}_1$ isomorphic to C^0_1 which was $[S]^1$ the last time an operation was applied.)

6. If $[S]^1_t$has participated in an **L**-operation an odd number of times since the construction last changed its isomorphism and the designated isomorphism is now r^1_t, then B^1_1 is the component that played the role of C^1_1 the last time a special component participated in an **R**–operation and C^1_1, C^0_1 have never participated in any **L**– or **R**–operations. (B^0_1 is the component in $\mathcal{A}_0$ isomorphic to B^1_1 which was $[S]^0$ the last time an operation was applied.)

In this case, extend $\mathcal{A}_{0,t}$ by applying an **L**–operation to the sequence $B^0_2, X^0, C^0_2, B^0_1, [S]^0_t, C^0_1$ and extend $\mathcal{A}_{1,t}$ by applying an **R**–operation to the sequence $B^1_2, X^1, C^1_2, B^1_1, [S]^1_t, C^1_1$. Note that the image of the coding location a_e is a new number v in $\mathcal{A}_{1,t+1}$. Put the pair (a_e, v) into P, the number a_e into $U^{\mathcal{A}_0}$, and the number v into $U^{\mathcal{A}_1}$. Note that we have successfully met requirement D_e and now have extended structures $\mathcal{A}_{k,t+1}$ and $\mathcal{A}_{k+1,t+1}$. Go on to substage 3.

Subcase 2.2. Suppose that the previous subcase does not hold. In this case, add new components to $\mathcal{A}_{0,t+1}$ and $\mathcal{A}_{1,t+1}$ isomorphic to $[b_1]$,$[b_2]$, $[c_1]$, $[c_2]$, where b_1, b_2, c_1, c_2 are new numbers. Go on to substage 3.

Substage 3: If this is not a recovery stage go on to the next stage. If it is, put a mark $\Box_w$ on each component that has participated in an **L**– or **R**–operation, on the new components isomorphic to $[a]$ $[b_i]$ or $[c_i]$ added on to $\mathcal{A}_0$ and $\mathcal{A}_1$ and on the component X in $\mathcal{A}_{0,t+1}$ (and on its image X' in

$\mathcal{A}_{1,t+1}$) which has the least number in any component not yet marked with a $\Box_w$. Go on to the next stage.

This concludes the description of the construction. Now we state several claims about the construction and sketch their proofs. It may be helpful in following the construction above and the proofs below to refer to the diagrams of the results of applying the operations to the special sets and their immediate neighbors in the sequences used supplied later in the proof of Lemma 2.7.

Claim 1. The construction meets all the requirements D_e. Moreover, if at some stage the enumeration of G enters the waiting state and never recovers, then G is not isomorphic to $\mathcal{A}_0$.

Proof of Claim 1. Suppose that at stage t, G enters the waiting state and never recovers. There is now clearly an e_0 such that for every $e > e_0$ for which $\Psi_e(a_e) = 0$ there is a stage t' such that at all stages $t > t'$ the construction satisfies condition 1 of substage 2 via e. Therefore $\Psi_e \neq U^{\mathcal{A}_0}$ by our action at such a stage t. Of course, if $\Phi_e(a_e) \neq 0$, then D_e is met automatically. Note that, in this case, all components of $\mathcal{A}_k$ are finite by construction. Thus it follows from the definition of recovery and the description of stages $t+1$ that G is not isomorphic to $\mathcal{A}_0$. (Of course, satisfying D_e for all $e > e_0$ implies that D_e is satisfied for every e.)

Suppose now that G recovers at infinitely many stages. Let e be the smallest number for which D_e is not met. Let t_1 be such that for all $e' < e$, all action for the requirements $D_{e'}$ are finished before stage t_1. (Clearly we act at most once for each D_e.) Then there exists a stage $t+1 > t_1$ at which $\Psi_{e,t+1}(a_e) = 0$ while condition 2 and Subcase 2.1 of Substage 2 holds. It follows that at stage $t+1$, the construction must meet requirement D_e. This is a contradiction. $\Box$

The next claim gives one of the fundamental properties of recovery.

Claim 2. Consider the sequence

$$Y_1,\ Y_2,\ Y_3, Y_4, S_t(G),\ Y_5$$

in G_t, and the sequences $B_2^k, X^k, C_2^k, B_1^k, [S]_t^k, C_1^k$ corresponding to it via the maps r_t^k. Suppose that at stage t an **L**–operation is applied to B_2^0, X^0, C_2^0, B_1^0, $[S]_t^0, C_1^0$ in $\mathcal{A}_0$ and an **R**–operation is applied to $B_2^1, X^1, C_2^1, B_1^1, [S]_t^1, C_1^1$ in $\mathcal{A}_1$. Let $t' > t$ be the least stage at which G recovers. Then $G_{t'}$ satisfies one of the following mutually exclusive properties:

1. Components Y_1, Y_2, Y_3, Y_4, $S_t(G)$, Y_5 are covered by $B_2^0 \cdot X^0$, $X^0 \cdot C_2^0$, $C_2^0 \cdot B_1^0$, $B_1^0 \cdot [S]_t^k$, $[S]_t^k \cdot C_1^k$, $C_1^k \cdot B_2^0$, respectively, in $\mathcal{A}_0$ and the corresponding isomorphic components in $\mathcal{A}_1$. If r_t^0 was the designated isomorphism at stage t, this covering corresponds to the fact that at stage t' the construction did not change its isomorphism from r^0 to r^1. If r_t^1 was the designated isomorphism at stage t, this covering corresponds to the fact that at stage t' the construction changed its isomorphism from r^1 to r^0.

2. Components Y_1, Y_2, Y_3, Y_4, $S_t(G)$, Y_5 are covered by $C_1^k \cdot B_2^0$, $B_2^0 \cdot X^0$, $X^0 \cdot C_2^0$, $C_2^0 \cdot B_1^0$, $B_1^0 \cdot [S]_t^k$, $[S]_t^k \cdot C_1^k$, respectively, in $\mathcal{A}_0$ and the corresponding isomorphic components in $\mathcal{A}_1$. If r_t^0 was the designated isomorphism at stage t, this covering corresponds to the fact that at stage t' the construction changed its isomorphism from r^0 to r^1. If r_t^1 was the designated isomorphism at stage t, this covering corresponds to the fact that at stage t' the construction did not change its isomorphism from r^1 to r^0.

Proof of Claim 2. Note first that there are no changes in any component of $\mathcal{A}_k$ with elements in the range of r_t^k between t and t' as these happen only at recovery stages by construction. The claim now follows directly from the fact that B_2^k, X^k, C_2^k and one of B_1^k and C_1^k have never participated in any operations before and so are of the form $[\{z\}]$ for distinct numbers z and the definitions of recovery, **L**–operation, and **R**–operation. (Each Y_i in G can grow into one of two components in $\mathcal{A}_k$ but once one is determined so are all the others by the requirements of uniqueness in the definition of recovery.) $\square$

Claim 3. $\mathcal{A}_k$ are of the form $[A_0^k] + \cdots + [A_n^k] + \cdots$; $A_n^k \backslash A_m^k \neq \emptyset$ for $n \neq m$ and $k = 0, 1$; and, for each stage t, $\mathcal{A}_{k,t}$ are of the form $[A_{0,t}^k] + \cdots + [A_{n,t}^k] + \cdots$; $A_{n,t}^k \backslash A_{m,t}^k \neq \emptyset$ for $n \neq m$ and $k = 0, 1$.

Proof of Claim 3. The second version of the claim (for each stage t) follows immediately by induction from the construction. Moreover, the only way that any of the components $[A_n^k]$ can be infinite is for there to be infinitely many recovery stages and for the construction to never change its isomorphism after some stage t_0. In this case, if the isomorphism remains r^k then $[S]_t^k$ becomes the one infinite component of $\mathcal{A}_k$. Thus the disjointness condition for the final components follows from the one at each stage. $\square$

Claim 4. Suppose that G recovers at infinitely many stages. If, after a stage t', the construction never changes its isomorphism from r_t^k, then the following hold:

1. For each $t > t'$, the special component $[S]_{t+1}^k$ extends $[S_t^k]$. Therefore $\bigcup_{t\to\infty}[S]_t^k$ is an infinite component.
2. If $t' < t_1 < t_2 < \dots$ is the sequence of all stages such that during each stage t_i the partial isomorphism $r_{t_i}^k$ properly extends the function $r_{t_{i-1}}^k$, then $[S]_{t_2}^{k+1}$ extends $[S]_{t'}^{k+1}$ and each $[S]_{t_{2(i+1)}}^{k+1}$ extends $[S]_{t_{2i}}^{k+1}$. Therefore the components $\bigcup_{t\geq t'}[S]_t^k$ and $\bigcup_{t\geq t'}[S_{2t}^{k+1}]$ are isomorphic.
3. Each component of $\mathcal{A}_k$ distinct from $\bigcup_{t\to\infty}[S]_t^k$ is finite.
4. If G is isomorphic to $\mathcal{A}$, then G is computably isomorphic to $\mathcal{A}_k$.

Proof of Claim 4. The first part of this claim follows from the definition of $[S]_t^k$ at stage t and the assumption. The second part follows from Claim 2 about the properties of the **L**– and **R**–operations and the description of the construction at stage $t+1$. The third part was already noted in the last Claim as, for any component X distinct from $[S]_t^k$, there exists a stage $t > t'$ such that after this stage the component X will never be used and so it is finite. To prove the last part of the claim consider the function $r^k = \bigcup_{t>t'} r_t^k$. By construction, every component X in $\mathcal{A}_k$ is eventually marked with a $\square_w$ and so there exists a stage $t'' > t'$ such that $r_{t''}^k Y \subset X$ for some component Y in G and so $r^k Y \subseteq X$ (as the construction never changes its isomorphism). As X is the only component of $\mathcal{A}_k$ contained in X and G is isomorphic to

$\mathcal{A}_k$, r^k must restrict to an isomorphism of Y onto X. Finally, if there were some component Y' of G not in the domain of r^k then as G is assumed isomorphic to $\mathcal{A}_k$, Y would be is isomorphic to some component X of $\mathcal{A}_k$ but each such X is isomorphic to some Y in G. Thus G would have two isomorphic components and so not be isomorphic to $\mathcal{A}_k$ by the last Claim.

Claim 5. If the construction changes its isomorphism at infinitely many stages, then the special G–component $S(G)$ is infinite and all components in $\mathcal{A}_0$ are finite. Therefore G is not isomorphic to $\mathcal{A}_0$.

Proof of Claim 5. For each component X, if X is distinct from $[S]^0_t$ for all t, then X is finite. Therefore it is enough to prove that the components in $\mathcal{A}_0$ which contain $[S]^0_t$ for some t are also finite. Let t_1, t_2, $\ldots$ be the sequence of all stages at which the construction changes its isomorphism. We can suppose that at stage t_1 the construction changes its isomorphism from $r^0_{t_1}$ to $r^1_{t_1}$. Consider $[S]^0_{t_1}$ and $[S]^0_{t_1-1}$. We have $[S]^0_{t_1-1} \bigcap [S]^0_{t_1} = \emptyset$. Hence, by the construction $[S]^0_{t_1-1}$ will never be used again. At stage t_2 the construction changes its partial isomorphism from $\mathcal{A}_1$ to $\mathcal{A}_0$. Now $[S]^1_{t_1} \bigcap [S]^1_{t_2} = \emptyset$ but $[S]^0_{t_2}$ extends $[S]^0_{t_1}$. However, after stage t_3 the component $[S]^0_{t_2}$ will never be used. Hence the component containing $[S]^0_{t_1}$ is finite. Continuing this procedure we see that neither $\mathcal{A}_0$ nor $\mathcal{A}_1$ contains an infinite component.

Claim 6. $\mathcal{A}_0$ and $\mathcal{A}_1$ are isomorphic.

Proof of Claim 6. It is immediate from the construction that $\mathcal{A}_{0,t}$ and $\mathcal{A}_{1,t}$ are isomorphic for every t. If there are only finitely many recovery stages, then every component of $\mathcal{A}_k$ is finite and the claim follows. Otherwise, there is exactly one infinite component in each and they are isomorphic by item 2 of Claim 4. Of course, the finite components are again isomorphic by the existence of isomorphisms at each stage. $\Box$

Claim 7. The relation P is computable.

Proof of Claim 7. Each time a pair (x, y) is put into P, y is a new number. Thus, the construction enumerates P in increasing order. $\Box$

The above claims prove the correctness of the construction with respect to one R and all D_e. $\square$

General Construction. We now describe a construction on a **priority tree** $\mathcal{T}$ that satisfies all the requirements. All nodes of a given length will have a fixed set of possible outcomes with a left to right ordering on them. The induced lexicographical ordering $\leq_L$ on the tree $\mathcal{T}$ coincides with the usual priority ordering on $\mathcal{T}$.

For every $\alpha \in \mathcal{T}$ of length $3j+2$, we will define an α–strategy to meet the requirement D_j. At stage t, the construction guarantees that some coding location is attached to each accessible node $\alpha \in \mathcal{T}$ of length $3j+2$. One of the goals of the construction is to satisfy D_j on one of these coding locations. The strategy to meet the requirement D_j employs the **L**– and **R**–operations. These strategies (nodes) have two possible outcomes at a stage t: The outcome is c if the construction now acts or has acted at some previous stage (since α was last initialized) to satisfy the requirement D_j. Otherwise, it is d. The left-to-right ordering on these outcomes is $c <_L d$.

For every $\alpha \in \mathcal{T}$ of length $3j+1$, we will define an α–strategy R_α to meet the requirement R_j. As the structure $\mathcal{A}$ we are building will be of the form $[A^0]+[A^1]+\cdots+[A^n]+\cdots$ with $A^n \nsubseteq A^m$ for $n \neq m$, we can limit ourselves to computable graphs G_j which are of this form. Moreover, without loss of generality, we can require that the enumerations $G_{j,t}$ be such that at no stage are there distinct components $[Y_0]$ and $[Y_1]$ of $G_{j,t}$ with $Y_0 \subseteq Y_1$ and that for every component $[Y]$ of $G_{j,t}$ there is one $[X]$ of $\mathcal{A}$ such that $Y \subseteq X$. (Only enumerate components in G_j when they are of the right form, i.e., a top with cycles attached (to which more cycles can be added later) and contained in one of $\mathcal{A}$. If there are components partially enumerated, do not allow any extensions until it is once again possible to make all the components distinct. This can be done so as to add any single desired element to G if G has the required form.)

The α-strategy to meet the requirement R_j is based on a stagewise definition of potential partial isomorphisms r^k_α which try to define isomorphisms from the structure G_j to $\mathcal{A}_k$ for $k = 0, 1$. These strategies have four possible outcomes at a stage t: The outcome is w if R_α is in the waiting state; the outcome is k if R_α recovers at stage t and the construction does not change its isomorphism from $r^k_{\alpha,t}$ to $r^{k+1}_{\alpha,t}$; the outcome is ∞ if R_α recovers at stage

t and the construction changes its isomorphism from $r^k_{\alpha,t}$ to $r^{k+1}_{\alpha,t}$. (These notions are analogous to the ones in the construction above and are defined precisely in the general construction below.) The outcomes are ordered as follows: $\infty < 0 < 1 < w$.

The nodes $\alpha \in \mathcal{T}$ of length $3j$ are devoted to guessing the components of G_j that are isomorphic to the infinite special components of $\mathcal{A}_0$ and $\mathcal{A}_1$ constructed (one each) by nodes $\beta \subset \alpha$ of length $3n+1$ such that $\beta\hat{\ }i \subset \alpha$ for $i = 0, 1$. This information is needed by the nodes γ of length $3j+1$ to successfully define the required isomorphism r^k_γ. The possible outcomes of a node of length $3j$ in order are $\infty < 0 < 1 < \ldots < n < \ldots$ for $n \in \omega$. The intention is that each n corresponds to a different guess as to the required components in G_j together with the correspondence to the desired components in $\mathcal{A}_k$ and ∞ is the outcome that there is no "eligible" guess or each such guess is eventually proven false.

We will define the **accessible** nodes of the priority tree at stage t by induction on their length. The empty sequence $\emptyset$ of length 0 is the root of the priority tree and is **accessible** at every stage t. If β is accessible at stage t and o is the outcome of β at t, then, $\beta\hat{\ }o$ is **accessible** at t. If $|\beta| < t$, we deal with $\beta\hat{\ }o$ if not instructed to go to the end of the stage. If $|\beta| = t$ we go to the end of the stage.

To **cancel** a component means to guarantee to never use it again in any **L**– or **R**–operation. By construction, any component X in $\mathcal{A}_{k,t}$ (and therefore its isomorphic image in $\mathcal{A}_{k+1,t}$) cancelled at stage t, will never be used at later stages of the construction. Therefore the component containing X in $\mathcal{A}_k$ will be equal to X itself. Hence X will be finite. To **initialize** a node β of length $3j+2$ at stage t means to **cancel** all the components associated with β. To **initialize** a node β of length $3j+1$ at stage t means to cancel the previous isomorphism $r^k_{\beta,t-1}$ and β-special components $[S]^k_{\beta,t-1}$ in $\mathcal{A}_k$ and the choice of $S(G_{\beta,t-1})$ in G_j and all other components associated with β that have participated in an **L**- or **R-**operation. If a special component $[S]^k_{\beta,t-1}$ is not cancelled or changed to another component at stage t, the construction keeps $[S]^k_{\beta,t}$ an extension of $[S]^k_{\beta,t-1}$. We will use this convention (of not changing $[S]^k_{\beta,t}$ without explicit mention) for all other parameters as well.

Now we describe the general construction. Remarks enclosed in double brackets [[like this one]] are explanatory only and not part of the formal construction.

Stage 0. Initialize all requirements β. For each $\alpha \in \mathcal{T}$ take distinct numbers $b_{\alpha,t}, c_{\alpha,t}, b_{\beta,\alpha,t}, c_{\beta,\alpha,t}, p_{\alpha,t}, q_{\alpha,t} \geq t+2$ such that the sets $\{b_{\alpha,t}| t \in \omega\}, \{c_{\alpha,t}| t \in \omega\}, \{b_{\beta,\alpha,t}| t \in \omega\}, \{c_{\beta,\alpha,t}| t \in \omega\}, \{p_{\alpha,t}| t \in \omega\}, \{q_{\alpha,t}| t \in \omega\}$ form a uniformly computable collection of disjoint sets for $\alpha \subset \beta \in \mathcal{T}$ with infinitely many numbers not in any of them. We say that the numbers $b_{\alpha,t}, c_{\alpha,t}, b_{\beta,\alpha,t}, c_{\beta,\alpha,t}, p_{\alpha,t}, q_{\alpha,t}$ are **associated** with α.

Stage t+1. We proceed to act for each accessible node β in turn until the stage is terminated. Let u be the stage at which β was last initialized and s be the last stage after u at which β was accessible (u if there is no such stage). As a node β is declared accessible we initialize all nodes γ to the right of β, i.e., $\beta <_L \gamma$ but $\beta \not\subseteq \gamma$.

Case 0: $|\beta| = 3j$. If the outcome of β at s was ∞ or $s = u$, the outcome of β is now the least n that has never been the outcome of β since u and is a code for a sequence of components of G_j that is eligible to be isomorphic to the corresponding special components $[S]^k_\gamma$ of $\mathcal{A}_k$ for $\gamma\hat{\ }k \subseteq \beta$. (If there are no γ of length $3i+1$ with $\gamma\hat{\ }k \subseteq \beta$ for $k = 0$ or 1, no guessing is needed and the outcome of β is always 0, the code for the empty correspondence.) We say that a component $[Y]$ of G_j (which at stage t is $[Y_t]$) is **eligible** to be isomorphic to $[S]^k_\gamma$ at stage t, if $q_{\gamma,v} \in Y_t$ where v is the stage at which $[S]^k_\gamma$ was last defined after u [[necessarily as $[q_{\gamma,v}]$]] and $Y_t \subseteq X_t$ or $Y_t \subseteq Z_t$where $[S]^k_{\gamma,t} = [X_t]$ and $[Z_t]$ is isomorphic to a component that participated in an **L**- or **R**-operation with $[S]^k_\gamma$ the last time it participated in one. [[This last option is included because such a *set* may grow into the desired special component.]] If there is no such n, the outcome of β is ∞. If the outcome of β at s was some $n \in \omega$, we see if we have irrefutable evidence that one element of the sequence of components of G_j coded by n is not isomorphic to the corresponding special components $[S]^k_\gamma$ of $\mathcal{A}_k$ for $\gamma\hat{\ }k \subseteq \beta$. If so, the outcome of β is ∞; if not, it is n. We say that we have **irrefutable evidence** at stage t that a component $[Y]$ of G_j (which at stage t is $[Y_t]$) is not isomorphic to $[S]^k_\gamma$ if there is a $z \in Y_t$ which is not associated with γ.

Case 1: $|\beta| = 3j+1$. If $\beta = \delta\hat{\ }\infty$, we think that G_j is not isomorphic to $\mathcal{A}$ and so do not need to do anything for R_j. In this case, the outcome of β is w. Suppose then that $\beta = \delta\hat{\ }n$ for some $n \in \omega$. We say that R_β **recovers at stage** $t+1$, or equivalently that stage $t+1$ is a β**-recovery stage**, if the following conditions hold:

1. Ignoring the components of G_j picked out by the outcome n of δ and the associated special components in $\mathcal{A}_0$ and $\mathcal{A}_1$ in the definition of covering, there exists (for each of $k = 0, 1$), for each component X in $\mathcal{A}_{k,t+1}$ marked with a $\square^\beta_w$, a unique component Y such that X covers Y.

2. If there has already been a β-recovery stage since u, we also require that, for each component $[Z_t]$ in the sequence coded by n that is supposed to correspond to $[S]^k_\gamma$ for $\gamma\hat{\ }k \subseteq \beta$, $Z_t \subseteq W$ where $[W] = [S]^k_{\gamma,v}$ for the last β-recovery stage v.

Otherwise, we say that R_β is in the **waiting state** at $t + 1$. In the latter case, the outcome of β is w. If at $t + 1$ there is no special component for β, we add components isomorphic to $[q_{\beta,t+1}]$ to $\mathcal{A}_{0,t+1}$ and $\mathcal{A}_{1,t+1}$ which we say are **associated** with β. (Note that when we perform an **L**- or **R**-operation the components maintain their association with a node α the same way they maintain their original elements, that is if a component originally associated with α contains various numbers when first added to the graph then the component containing those numbers is associated with α at every later stage.) We declare the components isomorphic to $[q_{\beta,t+1}]$ to be the special components $[S]^k_{\beta,t+1}$ for β. We also let $r^0_{\beta,t+1}$ and $r^1_{\beta,t+1}$ be empty and designate $r^0_{\beta,t+1}$ as the construction's isomorphism for β at $t+1$. In this case, too we let the outcome of β be w.

If $t+1$ is a recovery stage for R_β and there is a special component for β we define the partial isomorphisms $r^k_{\beta,t+1}$ (for $k = 0, 1$) from some components Y of $G_{j,t+1}$ into components X of $\mathcal{A}_k$ marked with $\square^\beta_w$ by making $r^k_{\beta,t+1}Y \subset X$ if the correspondence coded by n matches Y and X . If the correspondence does not mention Y or X, we make $r^k_{\beta,t+1}Y \subset X$ if and only if Y is covered by X when we ignore the components mentioned by the code n and their corresponding components in $\mathcal{A}_0$ and $\mathcal{A}_1$. If $S(G_{\beta,s})$ was not defined, we set it to be the component of G_j mapped into $[S]^0_\beta$ by $r^0_{\beta,t+1}$. We now define the outcome of R_β as follows:

Subcase 1. Suppose that $r^k_{\beta,t+1}$ extends the previous designated isomorphism $r^k_{\beta,s}[[= r^k_{\beta,t}$ and component $[S]^k_{\beta,t+1}$ extends $[S]^k_{\beta,t}$ (and $[S]^k_{\beta,t} = [S]^k_{\beta,s})]]$. In this case, the outcome of β is k.

Subcase 2. Suppose that $r^k_{\beta,t+1}$ does not extend the previous designated isomorphism $r^k_{\beta,t}$. In this case, the construction changes its (designated) isomorphism from $r^k_{\beta,t}$ to $r^{k+1}_{\beta,t+1}$ and the outcome of β is ∞. [[Note that, in this case, the component $[S]^{k+1}_{\beta,t+1}$ extends $[S]^{k+1}_{\beta,t}$ (and $[S]^{k+1}_{\beta,t} = [S]^{k+1}_{\beta,s}$) and $[S]^k_{\beta,t+1} \bigcap [S]^k_{\beta,t} = \emptyset$.]]

Case 2: $|\beta| = 3j+2$. If $\beta\hat{\ }c$ was the outcome of β at s, it is so again. If not, and there is no coding location attached to β, we add components isomorphic to $[b_{\beta,t+1}]$, $[c_{\beta,t+1}]$ and $[p_{\beta,t+1}]$ to each of $\mathcal{A}_{0,t+1}$ and $\mathcal{A}_{1,t+1}$ which we **associate** with β and attach the coding location in the copy of $[p_{\beta,t+1}]$ to β. Let a be the coding location attached to the node β. Let X^0 be the component in $\mathcal{A}_{0,t+1}$ containing the coding location a and X^1 be the isomorphic image of X^0 in $\mathcal{A}_{1,t+1}$. Let B^k_{n+1} and C^k_{n+1} be the components that we associated with β at the stage at which a was attached to it.

Let $k_1 < \ldots < k_n \leq t+1$ be the sequence of numbers i such that $\beta(3i+1) \neq w$. We let β_i denote $\beta \restriction (3k_i + 1)$. If there are no such numbers, $n = 0$ and the corresponding conditions in the description below are vacuous. If it is not the case that $\Psi_{j,v}(a) = 0$, the outcome of β is d. If this is the first stage v since u at which β is accessible such that $\Psi_{j,v}(a) = 0$, we add new components to $\mathcal{A}_{0,t+1}$ and $\mathcal{A}_{1,t+1}$ isomorphic to $[b_{\beta,\beta_i,,t+1}]$ and $[c_{\beta,\beta_i,t+1}]$ associated with β_i for each $i \leq n$. In this case, the outcome of β is also d. Otherwise, we have two subcases.

Subcase 2.1. There exists a sequence

$$B^0_{n+1}, X^0, C^0_{n+1}, B^0_n, [S]^0_{\beta_n,t+1}, C^0_n, \ldots, B^0_1, [S]^0_{\beta_1,t+1}, C^0_1$$

of components in $\mathcal{A}_{0,t+1}$ and the corresponding isomorphic image

$$B^1_{n+1}, X^1, C^1_{n+1}, B^1_n, [S]^1_{\beta_n,t+1}, C^1_n, \ldots, B^1_1, [S]^1_{\beta_1,t+1}, C^1_1$$

of the sequence in $\mathcal{A}_{1,t+1}$ such that B^k_i and C^k_i are associated with β_i with the following properties:

1. For each $i \in \{1, \ldots, n\}$ and $k \in \{0, 1\}$, the image of $r^k_{\beta_i,t+1}$ has non-empty intersection with every component in the sequence $B^k_{n+1}, X^k, C^k_{n+1}, B^k_n$, $[S]^k_{\beta_n,t+1}, C^k_n, \ldots, B^k_1, [S]^k_{\beta_{i_1},t+1}, C^k_1$ except for either B^k_i if the designated

isomorphism for β_i is $r^0_{\beta_i,t+1}$ or C^k_i if the designated isomorphism for β_i is $r^1_{\beta_i,t+1}$.

2. If $[S]^0_{\beta_i,t+1}$ and $[S]^1_{\beta_i,t+1}$ have participated in an **L**– or **R**–operation (respectively) an even number of times (possibly 0) since the construction last changed its isomorphism for β_i after u (e.g., it just changed it now and so $\beta_i\hat{\ }\infty \subseteq \beta$) then the components B^k_i, C^k_i have never participated in any **L**– or **R**–operations and are isomorphic to $\{b_{\beta,\beta_i,u}\}$ and $\{c_{\beta,\beta_i,u}\}$, respectively, for some $u < t+1$.

3. If $[S]^0_{\beta_i,t+1}$ has participated in an **L**-operation an odd number of times since the construction last changed its isomorphism for β_i and the designated isomorphism is now $r^0_{\beta_i,t+1}$, then C^0_i is the component that played the role of B^0_i the last time a special component for β_i [[necessarily $[S]^0_{\beta,i,u}$ for some $u < t+1$]] participated in an **L**–operation and B^0_i, B^1_i have never participated in any **L**– or **R**–operations and are isomorphic to $\{b_{\beta,\beta_i,u}\}$ for some $u < t+1$. [[C^1_i is the component in $\mathcal{A}_1$ isomorphic to C^0_i which was $[S]^1_{\beta_i}$ the last time an operation was applied.]]

4. If $[S]^1_{\beta_n,t+1}$has participated in an **L**-operation an odd number of times since the construction last changed its isomorphism for β_i and the designated isomorphism is now $r^1_{\beta_i,t+1}$, then B^1_i is the component that played the role of C^1_i the last time a special component for β_i [[necessarily $[S]^1_{\beta_i,,u}$ for some $u < t+1$]] participated in an **R**–operation and C^1_i, C^0_i have never participated in any **L**– or **R**–operations and are isomorphic to $\{c_{\beta,\beta_i,u}\}$ for some $u < t+1$. [[B^0_i is the component in $\mathcal{A}_0$ isomorphic to B^1_i which was $[S]^0_{\beta_i}$ the last time an operation was applied.]]

Action for D_j at β: In this subcase, extend $\mathcal{A}_{0,t+1}$ by applying an **L**–operation to the β-transform of the sequence

$$B^0_{n+1}, X^0, C^0_{n+1}, B^0_n, [S]^0_{\beta_n,t+1}, C^0_n, \dots D^0_1, B^0_1, [S]^0_{\beta_1,t+1}, C^0_1$$

and extend $\mathcal{A}_{1,t+1}$ by applying an **R**–operation to the β-transform of the sequence

$$B^1_{n+1}, X^1, C^1_{n+1}, B^1_n, [S]^1_{\beta_n,t+1}, C^1_n, \dots D^1_1, B^1_1, [S]^1_{\beta_1,t+1}, C^1_1.$$

Here we define the β-*transform* of the sequence $B^k_{n+1}, X^k, C^k_{n+1}, B^k_n, [S]^k_{\beta_{n,t+1}}, C^k_n, \ldots D^k_1, B^k_1, [S]^k_{\beta_1,t+1}, C^k_1$ as the sequence $B^k_{i_1}, [S]^k_{\beta_{i_1},t+1}, C^k_{i_1}, B^k_{i_2}, [S]^k_{\beta_{i_2},t+1}, C^k_{i_2}, \ldots, B^k_{i_m}, [S]^k_{\beta_{i_m},t+1}, C^k_{i_m}, B^k_{n+1}, X^k, C^k_{n+1}, B^k_{j_1}, [S]^k_{\beta_{j_1},t+1}, C^k_{j_1}, B^k_{j_2}, [S]^k_{\beta_{j_2},t+1}, C^k_{j_2}, \ldots, B^k_{j_r}, [S]^k_{\beta_{j_r},t+1}, C^k_{j_r}$ where $i_1, i_2, \ldots, i_m$ list, in order, the i such that the designated isomorphism for β_i at $t+1$ is $r^0_{\beta_i,t+1}$ and $j_1, j_2, \ldots, j_r$ list, in order, the i such that the designated isomorphism for β_i is $r^1_{\beta_{i,t+1}}$.

Note that the image of the coding location a is a new number v in $\mathcal{A}_{1,t+1}$. Put the pair (a, v) into P, the number a into $U^{\mathcal{A}_0}$, and the number v into $U^{\mathcal{A}_1}$.

In this subcase, the outcome of β is c and we go to the end of stage $t+1$.

Subcase 2.2. Otherwise, the outcome of β is d.

At the end of stage $t+1$ we do some cancelation (in addition to the initializations of nodes to the right of accessible ones) and marking. Suppose $t+1$ is a γ recovery stage (and so γ is accessible). If there are any uncancelled components isomorphic to $[b_{\beta,\gamma,u}]$ or $[c_{\beta,\gamma,u}]$ for $u \leq t$ (and $\beta \supseteq \gamma$) which, necessarily, have not participated in any operation, we cancel them and appoint new ones $[b_{\beta,\gamma,t+1}]$ or $[c_{\beta,\gamma,t+1}]$, respectively. [[Thus each will eventually get marked as necessary but whenever β is accessible there will be components available as needed to act for β.]] Now to describe the marking procedure for $\square^\gamma_w$ when $t+1$ is a γ-recovery stage.

If not already marked with $\square^\gamma_w$, we mark all of the following with $\square^\gamma_w$:

1. Any cancelled component.

2. Any component associated with a node β to the left of γ.

3. Any component of the form $[q_{\beta,t+1}]$ with $\beta \supseteq \gamma$.

4. Any components of the form $[b_{\beta,t+1}]$, $[c_{\beta,t+1}]$ or $[p_{\beta,t+1}]$ for $\beta \supseteq \gamma$.

5. Any components of the form $[b_{\beta,\beta_i,t+1}]$ or $[c_{\beta,\beta_i,t+1}]$ for $\beta \supseteq \beta_i \supset \gamma$.

6. Any components of the form $[b_{\beta,\gamma,t+1}]$ if $\beta \supseteq \gamma$ and the designated isomorphism for γ is $r^1_{\gamma,t+1}$or of the form $[c_{\beta,t+1}]$ if $\beta \supseteq \gamma$ and the designated isomorphism for γ is $r^0_{\gamma,t+1}$.

7. If, at $t+1$, we performed an operation on the β-transform of a sequence $B^0_{n+1}, X^0, C^0_{n+1}, B^0_n, [S]^0_{\beta_n}, C^0_n, \ldots D^0_1, B^0_1, [S]^0_{\beta_1,t+1}, C^0_1$ (and so $\gamma \subseteq \beta$),

then, for $k = 0, 1$, we mark B_i^k or C_i^k if $\beta_i \subseteq \gamma$ and it has previously participated in an operation; we mark B_i^k if $\beta_i \subseteq \gamma$ and the designated isomorphism for β_i is $r^1_{\beta_i,t+1}$; we mark C_i^k if $\beta_i \subseteq \gamma$ and the designated isomorphism for β_i is $r^0_{\beta_i,t+1}$; and we mark both B_i^k and C_i^k for $\beta_i = \gamma$. [[We are marking components that have not been marked before but will now never be used again in an operation with $[S]^0_\gamma$ in a position to its left.]]

This concludes the description of the construction.

Thus, we have constructed two structures $\mathcal{A}_0$ and $\mathcal{A}_1$, where

$$\mathcal{A}_0 = \bigcup_t \mathcal{A}_{0,t} \quad \text{and} \quad \mathcal{A}_1 = \bigcup_t \mathcal{A}_{1,t}.$$

The following two lemmas state several basic obvious facts about the construction.

Lemma 2.4 *The following properties hold of the construction:*

1. *For any component X^k, if X^k is cancelled at stage t, then the construction never uses the component X^k or X^{k+1} in any* **L***– or* **R***–operation after stage t. Therefore X^k and X^{k+1} are finite isomorphic components.*

2. *Component X is infinite if and only if the set* $\{t|$ at stage t the set X participated in an **L**– or **R**–operation$\}$ *is infinite.* □

Lemma 2.5 *Suppose that at a stage t an* **L***–operation (***R***–operation) is applied to the β-transform of the sequence*

$$B^k_{n+1}, X^k, C^k_{n+1}, B^k_n, [S]^k_{\beta_n,t}, C^k_n, \ldots, B^k_1, [S]^k_{\beta_1,t}, C^k_1.$$

Consider the node β_i. If there exists a stage $t' > t$ at which a node to the left of β_i is accessible, then the components

$$B^k_{n+1}, X^k, C^k_{n+1}, B^k_n, [S]^k_{\beta_n}, C^k_n, \ldots, B^k_i, [S]^k_{\beta_i,t}, C^k_i$$

never participate in any **L***– or* **R***–operation at any $t_1 > t'$, and therefore the components in $\mathcal{A}_k$ containing*

$$B^k_{n+1}, X^k, C^k_{n+1}, B^k_n, [S]^k_{\beta_{n,t}}, C^k_n, \ldots, B^k_i, [S]^k_{\beta_i,t}, C^k_i$$

are finite. □

Now in order to prove the correctness of the construction, we need to consider **the true path** $\mathcal{P}$ on the tree $\mathcal{T}$, that is the leftmost path on $\mathcal{T}$ whose nodes are accessible infinitely often. Thus, η is on the true path if an only if there are infinitely many stages at which η is accessible and there exists a stage t after which no β to the left of η is accessible. It is clear that there is a unique true path $\mathcal{P}$ on $\mathcal{T}$.

Lemma 2.6 *The relation* $U^{\mathcal{A}_0}$ *is computably enumerable but not computable. The relation* $U^{\mathcal{A}_1}$ *is computable. In particular, the structures* $\mathcal{A}_0$ *and* $\mathcal{A}_1$ *are not computably isomorphic.*

Proof. Note that at stage t if we put an element v into $U^{\mathcal{A}_1}$, then v is new and hence is greater than all elements appearing in $U^{\mathcal{A}_1}$ before stage t. Therefore the construction effectively lists all the elements in $U^{\mathcal{A}_1}$ in a strictly increasing order. Hence $U^{\mathcal{A}_1}$ is computable.

Suppose that $U^{\mathcal{A}_0}$ is also computable and Ψ_j is a characteristic function for $U^{\mathcal{A}_0}$. Consider the requirement D_j and the node $\beta = a_0 \dots a_{3j+1}$ on the true path corresponding to D_j. Let t be the first stage such that β is accessible at t but no η to the left of β is ever accessible after stage t. Let a be the coding location which is attached to the node β at stage t and B^k_{n+1}, C^k_{n+1} be the components associated with β when a was attached to it. If there is no $t_1 > t$ such that $\Psi_{j,t_1}(a) = 0$, the requirement is clearly satisfied, so suppose there is such a stage and let t_1 be the first such at which β is accessible. We add new components isomorphic to $[b_{\beta,\beta_i,t_1}]$ and $[c_{\beta,\beta_i,t_1}]$ at stage t_1 by construction that can be used in an operation only when we act to satisfy D_j at β. Moreover, by the cancelation and appointment procedures at the end of each later β_i-stage and the defintion of recovery, there will always be unused components of the form $[b_{\beta,\beta_i,v}]$ and $[c_{\beta,\beta_i,v}]$ marked and available as needed in the definition of acting for β at every β-stage. Thus at the next β-stage, t_2, Subcase 2.1 of Case 2 of the construction must hold if we have not already satisfied D_j by acting for β. (None of the new components needed can have been used in other operations by construction.) It follows that the construction must meet the requirement D_j at stage t_2 and so Ψ_j can not be the characteristic function of $U^{\mathcal{A}_0}$. $\square$

Lemma 2.7 *1. Suppose that at a stage t an* $\mathbf{L}$*–operation is applied to the β-transform of the sequence*

$$B^0_{n+1}, X^0, C^0_{n+1}, B^0_n, [S]^0_{\beta_n,t}, C^0_n, \dots, B^0_1, [S]^0_{\beta_1,t}, C^0_1.$$

(And so each β_i in the sequence is accessible at t.) Consider the node β_i and suppose that the designated partial isomorphism from G_{k_i} to $\mathcal{A}_{0,t}$ is $r^0_{\beta_i,t}$ at t. If t_1 is the first recovery stage for β_i after t (and so, in particular, β_i is accessible at t_1) and, between stage t_1 and stage t, no nodes to the left of β_i are accessible, then the following hold:

(a) *If the construction changes its isomorphism from $r^0_{\beta_i}$ to $r^1_{\beta_i}$ at stage t_1, then $[S]^0_{\beta_{i,t}}$ and so the G_{k_i}–special component $S(\beta_i, t_1)$ is embedded into $B^0_i \cdot [S]^0_{\beta_{i,t}}$ (the component corresponding to B^0_i) which becomes $[S]^0_{\beta_i,t_1}$ and $[S]^0_{\beta_i,t_1}$ has empty intersection with component $[S]^0_{\beta i,t}$.*

(b) *If the construction does not change its isomorphism from $r^0_{\beta_i}$ to $r^1_{\beta_i}$ at stage t_1, then $[S]^0_{\beta_{i,t}}$ and so the G_{k_i}–special component $S(\beta_i, t_1)$ is embedded into $[S]^0_{\beta_i,t}{\cdot}C^0_i$ (the component corresponding to $[S]^0_{\beta_i,t}$) which becomes $[S]^0_{\beta_i,t_1}$ and $[S]^0_{\beta_i,t_1}$ extends component $[S]^0_{\beta_i,t}$. Moreover, the component B^0_i becomes $B^0_i \cdot [S]^0_{\beta_i,t}$.*

2. *Suppose that at a stage t an $\mathbf{R}$–operation is applied to the β-transform of the sequence*

$$B^1_{n+1}, X^1, C^1_{n+1}, B^1_n, [S]^1_{\beta_n,t}, C^1_s, \ldots, B^1_1, [S]^1_{\beta_1,t}, C^1_1.$$

(And so each β_i in the sequence is accessible at t.) Consider the node β_i and suppose that the designated partial isomorphism from G_{k_i} to $\mathcal{A}_{1,t}$ is $r^1_{\beta_i,t}$ at t. If t_1 is the first recovery stage for β after t (and so in particular β_i is accessible at t_1) and no nodes to the left of β_i are accessible between stage t_1 and stage t then the following hold:

(a) *If the construction changes its isomorphism from $r^1_{\beta_i}$ to $r^0_{\beta_i}$ at stage t_1, then $[S]^1_{\beta_{i,t}}$ and so the G_{k_i}–special component $S(\beta_i, t_1)$ is embedded into $C^1_i \cdot [S]^1_{\beta_{i,t}}$ (the component corresponding to C^1_i) which becomes $[S]^1_{\beta_i,t_1}$ and $[S]^1_{\beta_i,t_1}$ has empty intersection with component $[S]^1_{\beta_i,t}$.*

(b) *If the construction does not change its isomorphism from $r^1_{\beta_i}$ to $r^0_{\beta_i}$ at stage t_1, then $[S]^1_{\beta_{i,t}}$ and so the G_{k_i}–special component $S(\beta_i, t_1)$ is embedded into $B^1_i{\cdot}[S]^1_{\beta_i,t}$ (the component corresponding to $[S]^1_{\beta_i,t}$)*

which becomes $[S]^1_{\beta_i,t_1}$ and $[S]^1_{\beta_i,t_1}$ extends component $[S]^1_{\beta_i,t}$. Moreover, the component C_i^1 becomes $C_i^1 \cdot [S]^0_{\beta_i,t}$.

Proof. This lemma follows directly from the definitions of recovery, **L**–operation, **R**–operation, and the construction. The crucial point is that if, say, the designated isomorphism is $r^0_{\beta_i,t}$ then all the components to the left of $[S]^0_{\beta_i,t}$ in the β-transform to which the operation is applied which are associated with β_i or nodes of lower priority are marked with $\Box^{\beta_i}_w$ and none of the others. Thus when we apply an **L**-operation to $\mathcal{A}_0$ none of the marked components can change their place in the isomorphism $r^0_{\beta_i,t_1}$ if $[S]^0_{\beta_i,t}$ does not change its place. The situation is analogous to that in **Claim 2** of the construction above with only one R_j. To make it easier to see this and later points we schematically display the results of applying the operations in various circumstances. For the sake of simplicity, we omit the subscripts for the node β and the stages t as well as the duplications in the product notation $(\cdot)$ for the results of applying the operations. We also omit the superscripts designating which component is in $\mathcal{A}_0$ and which in $\mathcal{A}_1$ and just display first the results in $\mathcal{A}_0$ and then those in $\mathcal{A}_1$.

Here is the result of a sequence of operations when the designated isomorphism remains r^0. Note that the B_i are new components each time. First on $\mathcal{A}_0$:

$$\begin{array}{ccccc} B_0 & \leftarrow & [S]^0 = S_0 & \leftarrow & C_0 \\ B_1 & \leftarrow & S_0 \cdot C_0 & \leftarrow & B_0 \cdot S_0 \\ B_2 & \leftarrow & S_0 \cdot C_0 \cdot B_0 & \leftarrow & C_2 \\ B_3 & \leftarrow & S_0 \cdot C_0 \cdot B_0 \cdot B_1 & \leftarrow & B_2 \cdot S_0 \cdot C_0 \cdot B_0 \end{array} \tag{1a}$$

Next on $\mathcal{A}_1$:

$$\begin{array}{ccccc} B_0 & \rightarrow & [S]^1 = S_0 & \rightarrow & C_0 \\ B_1 & \rightarrow & C_0 \cdot S_0 & \rightarrow & S_0 \cdot B_0 \\ B_2 & \rightarrow & S_0 \cdot B_0 \cdot C_0 & \rightarrow & C_2 \\ B_3 & \rightarrow & C_2 \cdot S_0 \cdot B_0 \cdot C_0 & \rightarrow & S_0 \cdot B_0 \cdot C_0 \cdot B_2 \end{array} \tag{1b}$$

Here are the results of applying the operations when the designated isomorphism remains r^1 with the same conventions. Note here that the C_i are new components each time. First on $\mathcal{A}_0$:

$$\begin{array}{ccccc} B_0 & \leftarrow & [S]^0 = S_0 & \leftarrow & C_0 \\ S_0 \cdot C_0 & \leftarrow & B_0 \cdot S_0 & \leftarrow & C_1 \\ B_2 & \leftarrow & S_0 \cdot C_0 \cdot B_0 & \leftarrow & C_2 \\ S_0 \cdot C_0 \cdot B_0 \cdot C_2 & \leftarrow & B_2 \cdot S_0 \cdot C_0 \cdot B_0 & \leftarrow & C_3 \end{array} \tag{2a}$$

Next on $\mathcal{A}_1$:

$$\begin{array}{ccccc} B_0 & \rightarrow & [S]^1 = S_0 & \rightarrow & C_0 \\ C_0 \cdot S_0 & \rightarrow & S_0 \cdot B_0 & \rightarrow & C_1 \\ B_2 & \rightarrow & S_0 \cdot B_0 \cdot C_0 & \rightarrow & C_2 \\ C_2 \cdot S_0 \cdot B_0 \cdot C_0 & \rightarrow & S_0 \cdot B_0 \cdot C_0 \cdot B_2 & \rightarrow & C_3 \end{array} \tag{2b}$$

And finally, here are the results if the construction starts with r^0 and then changes its isomorphism at each stage so that all the B_i and C_i are new. First on $\mathcal{A}_0$:

$$\begin{array}{ccccc} B_0 & \leftarrow & [S]^0 = S_0 & \leftarrow & C_0 \\ B_1 & \leftarrow & B_0 \cdot S_0 & \leftarrow & C_1 \\ B_2 & \leftarrow & B_0 \cdot S_0 \cdot C_1 & \leftarrow & C_2 \\ B_3 & \leftarrow & B_2 \cdot B_0 \cdot S_0 \cdot C_1 & \leftarrow & C_3 \end{array} \tag{3a}$$

Next on $\mathcal{A}_1$:

$$\begin{array}{ccccc} B_0 & \rightarrow & [S]^1 = S_0 & \rightarrow & C_0 \\ B_1 & \rightarrow & S_0 \cdot B_0 & \rightarrow & C_1 \\ B_2 & \rightarrow & C_1 \cdot S_0 \cdot B_0 & \rightarrow & C_2 \\ B_3 & \rightarrow & C_1 \cdot S_0 \cdot B_0 \cdot B_2 & \rightarrow & C_3 \end{array} \tag{3b}$$

$\square$

Before we prove the next lemma we again note that each component in $\mathcal{A}_0$ $(\mathcal{A}_1)$ is of the form $[\bigcup_t A^n_{0,t}]$ $([\bigcup_t A^m_{1,t}])$ for some n (m). We denote this component by $\mathcal{A}^n_0$ $(\mathcal{A}^m_1)$.

Lemma 2.8 *For any component $\mathcal{A}^n_0$ in $\mathcal{A}_0$ there is a component $\mathcal{A}^m_1$ in $\mathcal{A}_1$ such that $\mathcal{A}^n_0$ and $\mathcal{A}^m_1$ are isomorphic. Similarly, for any component $\mathcal{A}^n_1$ in $\mathcal{A}_1$ there is a component $\mathcal{A}^m_0$ in $\mathcal{A}_0$ such that $\mathcal{A}^n_1$ and $\mathcal{A}^m_0$ are isomorphic.*

Proof. Suppose that $\mathcal{A}_0^n$ is finite. There exists a stage t such that $\mathcal{A}_0^n = \mathcal{A}_{0,t}^n$. This component cannot participate in an **L**–operation at any stage $t' > t$. Hence its image, say $\mathcal{A}_{1,t'}^m$ does not participate in an **R**–operation. Therefore component $\mathcal{A}_0^n$ is isomorphic to $\mathcal{A}_1^m$. (Only components in $\mathcal{A}_0$ participate in **L**-operations and ones in $\mathcal{A}_1$ in **R**-operations.)

Suppose now that the component $\mathcal{A}_0^n$ is infinite. Then the set

$$\{t| \text{ at stage } t \text{ the set } \mathcal{A}_{0,t}^n \text{ participates in an } \mathbf{L}\text{-operation } \}$$

is infinite. Let $t_1 < t_2 < \ldots < t_n < \ldots$ be the list of all elements of this set. Consider stage t_j for any j. At this stage $\mathcal{A}_{0,t_j}^n$ participates in an **L**-operation. Let

$$B_{v+1}^0, X^0, C_{v+1}^0, B_v^0, [S]_{\beta_v,t_j}^0, C_v^0, \ldots, B_1^0, [S]_{\beta_1,t_j}^0, C_1^0$$

be all the components participating in this **L**–operation including $\mathcal{A}_{0,t_j}^n$. As the components $B_{v+1}^0, X^0, C_{v+1}^0$ can participate in an **L**-operation at most once, $\mathcal{A}_{0,t_j}^n$ belongs to the set $\{B_i^0, [S]_{\beta_i,t_j}^0, C_i^0\}$ and is associated with an accessible β_i for some $i \leq v$ (and remains so associated). For any stage t between stages t_j and t_{j+1}, no node to the left of β_i can be accessible since otherwise, by Lemma 2.3, $\mathcal{A}_{0,t}^n$ would be cancelled and hence finite.

If the construction changes its isomorphism at β_i infinitely often then no component associated with β_i can participate in an operation infinitely often: The ones playing the roles of B_i^k and C_i^k are new each time the isomorphism changes and the $[S]_{\beta_i}^k$ change every other time by Lemma 2.7 and once changed can never be used again. (See (3a) and (3b) above.) Thus there is a least j such that the construction never changes the isomorphism for β_i after t_j. There are two cases to consider:

Case 1. Suppose that the designated isomorphism is always $r_{t_k}^0$ for $k \geq j$. In this case, the only component of $\mathcal{A}_0$ that participates infinitely often is the special component $[S]_{t_k}^0$ which is increasing in $k \geq j$. (See (1a) and (1b) above.) Moreover, the components $[S]_{t_{j+2k+2}}^1$ are isomorphic to $[S]_{t_{j+2k+2}}^0$ and also form an increasing sequence. Thus the two limits $\cup\{[S]_{t_{j+2k+2}}^0\}$, which is the given $\mathcal{A}_0^n$, and $\cup\{[S]_{t_{j+2k+2}}^1\}$, which is some $\mathcal{A}_1^m$, are isomorphic as required.

Case 2. Suppose that the designated isomorphism is always $r_{t_k}^1$ for $k \geq j$. In this case, the only component of $\mathcal{A}_0$ that participates infinitely often is the one that alternates between being $[S]_{t_{j+2k}}^0$ and playing the role of B_i^0 in the operations at stages t_{j+2k+1}. (See (2a) and (2b) above.) In this case, the two increasing limits $\cup\{[S]^0_{t_{j+2k}}\}$ which is the given $\mathcal{A}^n_0$ and $\cup\{[S]^1_{t_{j+2k}}\}$ which is some $\mathcal{A}^m_1$ are isomorphic as required.

The proof going from $\mathcal{A}^n_1$ to $\mathcal{A}^m_0$ is similar. $\square$

Lemma 2.9 *For each $k = 0, 1$ and $n \neq m$, $A^n_k \not\subseteq A^m_k$, Moreover, for each $k = 0, 1$ and $n \neq m$, $A^n_{k,t} \not\subseteq A^m_{k,t}$ for every t. In addition, each infinite special component $[S]^k_\alpha = [X]$ has only numbers associated with α in the set X.*

Proof. The stage by stage distinctions follow immediately from the construction by induction. The only additional concern is the infinite components. As there is at most one for each α and they are all clearly differentiated by having only cycles of length some $q_{\alpha,t}, b_{\alpha,t}, c_{\alpha,t}, b_{\beta,\alpha,t}$ or $c_{\beta,\alpha,t}$ which are all distinct for different α. $\square$

Lemma 2.10 *The structures $\mathcal{A}_0$ and $\mathcal{A}_1$ are isomorphic.*

Proof. A back and forth argument constructs the desired isomorphism by the previous two lemmas.$\square$

We now wish to prove that all the requirements R_j are satisfied. We first analyze the outcomes of the appropriate nodes on the true path under the assumption that G_j is isomorphic to $\mathcal{A}$.

Lemma 2.11 *If G_j is isomorphic to $\mathcal{A}$, α is the node of length $3j$ on $\mathcal{P}$ and n is the code of the sequence of components of G_j isomorphic to the infinite special sets of higher priority (the $[S]^k_\gamma$ for $\gamma\hat{\ }k \subseteq \alpha$ and $|\gamma| = 3i+1$ for $i < j$) and the correct correspondence between them and the isomorphic components of $\mathcal{A}_0$ and $\mathcal{A}_1$, then $\alpha\hat{\ }n$ is on $\mathcal{P}$.*

Proof. Let u be the last stage at which α is initialized. It is clear that n is eventually eligible to be the outcome of α as G_j is isomorphic to $\mathcal{A}$. As n represents the correct isomorphism on the relevant components, once eligible it can never later fail to be eligible. As all these components $[X] = [S]^k_\gamma$ only have numbers z associated γ in the set X, with we can never have irrefutable evidence that n is not the correct outcome. Thus, we only need to show that we get irrefutable evidence that each code $m < n$ is not correct. Each one must have some correspondence between a component $[Y]$ of G_j and one $[X]$ of $\mathcal{A}$ that is eligible but not correct. For it to be eligible, Y must contain the number $q_{\gamma,v}$ for the stage v at which we correctly defined $[S]^k_\gamma$. Thus, if Y is not the component of G_j isomorphic to $[S]^k_\gamma$, it must be isomorphic to one of the ones that play the role of B_i or C_i with $[S]^k_\gamma$ when some operation is applied. It is obvious from the definition of the sequence of operations (as displayed above) that each such component eventually gets an addition which includes a number not associated with γ for the desired contradiction. $\square$

Lemma 2.12 *If G_j is isomorphic to $\mathcal{A}$ and $\beta = \alpha\hat{\ }n$ is the node of length $3j+1$ on $\mathcal{P}$, then $\beta\hat{\ }k \in \mathcal{P}$ for $k = 0$ or 1.*

Proof. Let u be the last stage at which α is initialized. Suppose first that $\alpha\hat{\ }\infty \in \mathcal{P}$. Consider the components $[S]^k_{\alpha,t}$ and $k \in \{0,1\}$. Let $t_0 < t_1 < t_2 < \ldots$ be the sequence of stages after u at which $\alpha\hat{\ }\infty$ is accessible so the construction changes its isomorphism from $r^0_{\alpha,t_{2n}}$ to $r^1_{\alpha,t_{2n}}$ at stage t_{2n} and from $r^1_{\alpha,t_{2n+1}}$ to $r^0_{\alpha,t_{2n+1}}$ at stage t_{2n+1} . Then, by construction, after stage t_{2n+k+3} no extension of $[S]^k_{\alpha,t_{2n+k}}$ will ever participate in any **L**– or **R**-operations. (See (3a) and (3b) above.) Therefore, by the construction, all components extending $[S]^k_{\alpha,t_i}$ are finite. Consider the corresponding special G_j–component $\alpha(S)$ in the structure G_j. Note that this special component is infinite but as we have just argued, all of the components of $\mathcal{A}_0$ which can be embedded in $\alpha(S)$ are finite. (Only components associated with α can be embedded in $\alpha(S)$ by construction.) Thus, we have contradicted the assumption that $\mathcal{A}$ and G_j are isomorphic.

Next suppose that $\alpha\hat{\ }w \in \mathcal{P}$ and let t_0 be that least stage at which $\alpha\hat{\ }w$ is accessible but after which no node to its left is accessible. So for all $t \geq t_0$, R_α is in the waiting state. Since n is the correct correspondence between the infinite special sets of higher priority and the appropriate components of G_j, the enumeration of G_j must eventually put into these components all the members in the ones in those of $\mathcal{A}$ at t_0 and so we cannot be waiting for this part of the desired recovery.

Thus we need only consider the finitely many components X in $\mathcal{A}_{k,t_o}$ $(k = 0, 1)$ marked with $\square^\alpha_w$ other than those coded by n. (No new marks are put down while R_α is in the waiting state.) Moreover, each such component X in this set is finite and so eventually constant. (Any components associated with a node to the right of $\alpha\hat{\ }w$ were canceled when α was last accessible before t_0 by our choice of t_0 and so are finite. None introduced after that stage can get a $\square_w^\alpha$ mark since these are assigned only at α-recovery stages. Components associated with nodes either extending α or to its left can also never participate in an operation again and so are finite. This leaves only components associated with $\gamma\hat{\ }k \subset \alpha$. Of these, the only ones that can possibly become infinite are the special components $[S]_\gamma^k$ and their isomorphic images in $\mathcal{A}_{k+1}$but these are excluded from consideration as being coded for by n by the previous Lemma.) As G_j is isomorphic to $\mathcal{A}$, there is eventually be a stage $t > t_0$ at which $G_{j,t}$ has an isomorphic copy of each such finite marked component X. The only way we would not now get a recovery stage (for the desired contradiction) is for there to be an additional component of $G_{j,t}$ which is also isomorphically embeddable into one of these X's. However, we explicitly prohibited such a situation from ever occurring in our enumeration of G_j. $\square$

Lemma 2.13 *If G_j is isomorphic to $\mathcal{A}$, $\beta = \alpha\hat{\ }n$ is the node of length $3j+1$ on $\mathcal{P}$ and $\beta\hat{\ }k \in \mathcal{P}$ for $k = 0$ or 1, then G_j is computably isomorphic to $\mathcal{A}_k$.*

Proof. Consider the effective sequence

$$t_1 < t_2 < t_3 < \ldots$$

such that no γ to the left of $\beta\hat{\ }k$ is accessible after t_1 and $\beta\hat{\ }k$ is accessible at each stage t_i. Since the outcome of the node β at each of these stages is k, it follows that the sequence

$$r^k_{\beta,t_1}, r^k_{\beta,t_2}, r^k_{\beta,t_3}, \ldots$$

is a sequence of partial isomorphisms such that $r^k_{\beta,t_i} \subseteq r^k_{\beta,t_{i+1}}$ for all $i \geq 1$. (The argument is as in the case of only one R_j with a few additional remarks: All components associated with nodes to the right of $\beta\hat{\ }k$ are cancelled and so undergo no further changes. None associated with nodes to the left of $\beta\hat{\ }k$ can participate in an operation (and so change) by our choice of t_1. None associated with a node extending $\beta\hat{\ }k$ can participate in an operation until $\beta\hat{\ }k$ is again accessible. Finally, the only ones associated with nodes $\gamma \subseteq \beta\hat{\ }k$ (other than the special components) that can participate in operations are marked with $\square_w^\beta$ (and so in the domain of $r^k_{\beta,t}$) only after they can no longer participate in any further operations and so cannot change either.)

Moreover, by the assignment procedure for marks $\square_w^\beta$, every component, except for finitely many, eventually gets a $\square_w^\beta$ mark. (The only possible exceptions are the special components for $\gamma \subset \beta$ and the ones initially of the form $[b_{\delta,t}]$, $[c_{\delta,t}]$ or $[p_{\delta,t}]$ for $\delta \subset \beta$ and $t \leq t_1$.) Now consider the function $r^k = \bigcup_{i\geq 1}(r^k_{\beta,t_i})$ and any component $[Y]$ of G such that $[Y] \bigcap dom(r^k) \neq \emptyset$. It is clear that r^k is an embedding of $[Y]$ into some component $[X]$ of $\mathcal{A}_k$. As G_j is isomorphic to $\mathcal{A}_k$ and there are no two components of $\mathcal{A}_k$ such that one can be embedded in the other by Lemma 2.9 nor can there be any proper embedding of any component into itself by construction, r^k must restrict to an isomorphism of $[Y]$ into $[X]$. As almost every component of $\mathcal{A}_k$ eventually gets a $\Box^{\beta}_{w}$, r^k maps onto $\mathcal{A}_k$ except for finitely many components. If some component $[Y]$ of G_j is not in the domain of r^k then it is still isomorphic to some $[X]$ in $\mathcal{A}_k$. As there cannot be two isomorphic components in G_j or in $\mathcal{A}_k$ by Lemma 2.9, the finitely many components left out by r^k must match up. Thus r^k extended by the unique possible matching of the finitely many components omitted is the desired computable isomorphism. $\Box$

Lemma 2.14 *The relation P is computable.*

Proof: Each time a pair (x, y) is put into P, y is a new number. Thus, the construction enumerates P in increasing order. $\Box$

The above lemmas prove the correctness of the construction. Thus we have proved Theorem 2.1 $\Box$

3 The Degree Spectra Problem

This section gives a positive solution to the **The Degree Spectra Problem** (Question 1.15). Our proof is based on the ideas of the proof of Theorem 2.1 from the previous section. We give the basic ideas of our proof. However, we do not intend to give the full proof of this result simply because we do not want to repeat the arguments and construction which are similar to the ones from Theorem 2.1.

Theorem 3.1 *For any computable partially ordered set $\mathcal{D}$ there exists a structure $\mathcal{A}$ of dimension the cardinality of $\mathcal{D}$ and a relation U on $\mathcal{A}$ whose degree spectrum, $DgSp(U)$, is isomorphic to $\mathcal{D}$. Indeed, we can guarantee that $U^{\mathcal{B}}$ is computably enumerable for every computable presentation $\mathcal{B}$ of $\mathcal{A}$ and that if $\mathcal{D}$ contains a least element, then the least element of $DgSp(U)$*

is $\mathbf{0}$*. Moreover, we can choose representatives* $\mathcal{A}_i$ *of the degree spectrum of* $\mathcal{A}$ *such that the relation* $P = \{(x, y)|$ *there are* i *and* j *such that* $d_i \not\leq d_j \wedge x \in U^{\mathcal{A}_i} \wedge y \in U^{\mathcal{A}_j} \wedge$ *there is an isomorphism from* $\mathcal{A}_i$ *to* $\mathcal{A}_j$ *which extends the map* $x \mapsto y\}$ *is computable where the isomorphism between* $DgSp(U)$ *and* $\mathcal{D}$ *takes* d_i *to* $U^{\mathcal{A}_i}$*.*

Sketch of the Proof. Let $d_0, d_1, \ldots$ be an effective list, without repetitions, of all elements of the partially ordered set $\mathcal{D}$. The goal is to construct a computable sequence $\mathcal{A}_0$, $\mathcal{A}_1$, $\ldots$ of computable structures (actually digraphs) each one isomorphic to a single rigid structure $\mathcal{A}$ which has a relation U such that the following properties hold:

1. Any computable presentation $\mathcal{B}$ of $\mathcal{A}$ is computably isomorphic to one of $\mathcal{A}_0$, $\mathcal{A}_1$, ...
2. The isomorphic images $U^{\mathcal{A}_0}, U^{\mathcal{A}_1}, \ldots$ of the relation U in $\mathcal{A}_0$, $\mathcal{A}_1$, ..., respectively, are such that for all $i, j \in \omega$, $d_i \leq d_j$ in $\mathcal{D}$ if and only if $U^{\mathcal{A}_i} \leq_T U^{\mathcal{A}_j}$.
3. The predicate P, such that for all (x, y) the pair $(x, y) \in P$ if and only if there are i, j such that $d_i \not\leq d_j$ with $x \in U^{\mathcal{A}_i}$ and $y \in U^{\mathcal{A}_j}$ for which the mapping $x \to y$ can be extended to an isomorphism from $\mathcal{A}_i$ to $\mathcal{A}_j$, is computable.

Let $G_0, G_1, G_2, \ldots$ be a standard enumeration of all computably enumerable digraphs. In order to build a structure $\mathcal{A}$ and a unary relations U with the properties above we need to satisfy at least the following requirements:

$$D_{n,m} : d_m \leq d_n \text{ if and only if } U^{\mathcal{A}_m} \leq_T U^{\mathcal{A}_n}$$

and

$$R_j \quad : \quad \text{If } G_j \text{ is isomorphic to } \mathcal{A} \text{ then } G_j \text{ is computably isomorphic to one of } \mathcal{A}_0, \mathcal{A}_1, \ldots$$

where $n, m, j \in \omega$, and $n \neq m$. Let $\Psi_0^X, \Psi_1^X, \ldots$ be an effective enumeration of all computable partial functions with oracle X. For all $n, m \in \omega$, in order to satisfy requirement $D_{n,m}$, our construction needs to divide the requirement into infinitely many (sub)requirements:

1. If $d_m \leq d_n$, then $D_{e,n,m} : U^{\mathcal{A}_n} \leq_T U^{\mathcal{A}_m}$.

2. If $d_m \not\leq d_n$, then $D_{e,n,m}$: $U^{\mathcal{A}_m} \neq \Psi_e^{U^{\mathcal{A}_n}}$ where $e, n, m \in \omega$.

Thus, our construction must satisfy requirements R_j and $D_{e,n,m}$ for all j, n, m, $e \in \omega$ with $n \neq m$.

A construction of the sequence $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ and unary predicates $U^{\mathcal{A}_0}$, $U^{\mathcal{A}_1}$, $U^{\mathcal{A}_2}, \ldots$ can be carried out in a priority construction on a tree $\mathcal{T}$ defined as follows. For each requirement R_j there is level $l(R_j) \geq j$ on the tree $\mathcal{T}$ such that each node β of this level is devoted to satisfying R_j and has exactly $j+2$ immediate successors $\beta\hat{}\infty$, $\beta\hat{}0$, $\beta\hat{}1$, ..., $\beta\hat{}j$, $\beta\hat{}w$. For each requirement $D_{e,n,m}$ with $d_m \not\leq d_n$ there is a level $l(D_{e,n,m})$ on the tree $\mathcal{T}$ such that each node β of this level is devoted to satisfy $D_{e,n,m}$ and has exactly 2 immediate successors $\beta\hat{}c$ and $\beta\hat{}d$. Moreover, we also assume that $l(D_{e,n,m}) > l(R_n)$ and $l(D_{e,n,m}) > l(R_m)$ for all $e, n, m \in \omega$. Here are the notions and ideas which are used in the construction.

At stage t each accessible node α of length $l(R_j)$ has a **special** component $S_{\alpha,t}$. The node α also has a potential finite partial isomorphisms $r^0_{\alpha,t}$, ..., $r^j_{\alpha,t}$ from G_j to $\mathcal{A}_{0,t}$, ..., $\mathcal{A}_{j,t}$ (one of which is the designated one for the construction at t). (As before these partial isomorphisms are based on a guess as to the correct correspondence between the infinite special components of higher priority and the appropriate components of G_j.) The idea is that if G_j is going to be isomorphic to the graph the construction is building, then the construction attempts to force the graph G_j to be computably isomorphic to one of $\mathcal{A}_0$, ..., $\mathcal{A}_j$ as follows. Suppose that α is on the true path. Then α can have $j+2$ many outcomes. If the outcome is $\alpha\hat{}\infty$, then this corresponds to the fact that the construction changes its mind about its potential isomorphism infinitely often. Hence the special component S_α becomes infinite, all the components in $\mathcal{A}$ which can be embedded into S_α are finite, and therefore G_j is not isomorphic to $\mathcal{A}$. If the outcome of α is $\alpha\hat{}w$, then at node α G_j is the waiting state and hence can not be isomorphic to $\mathcal{A}$. If the outcome is $\alpha\hat{}i$ for some $i \leq j$, then G_j is computably isomorphic to $\mathcal{A}_i$.

At stage t for each node α of length $l(D_{e,n,m})$ the construction picks a coding location $a_{\alpha,t}$ and attempts to meet requirement $D_{e,n,m}$ of the form $U^{\mathcal{A}_m} \neq \Psi_e^{U^{\mathcal{A}_n}}$ at this location. The idea is the following. First of all the action to meet any requirement of type $D_{e,n,m}$ is based on performing **L** as well as **R**–operations in the structures the construction is building. Suppose that α on the true path. There is a stage t' such that no β to the left of α is ever accessible after t'. We can assume that the coding location $a_{\alpha,t}$ is attached to the node after this stage. If there exists a stage $t_1 > t'$ such that $\Psi_{e,t_1}^{U^{\mathcal{A}_{n,t}}}(a_{\alpha,t}) = 0$, then the construction puts $a_{\alpha,t}$ into $U^{\mathcal{A}_m}$ at some α-stage. Moreover, the construction puts the images of $a_{\alpha,t}$ into $U^{\mathcal{A}_i}$ for $i \neq m$. These images of the coding location $a_{\alpha,t}$ are new numbers if $d_m \not\leq d_i$ and are the same number $a_{\alpha,t}$ put into $U^{\mathcal{A}_m}$ if $d_m \leq d_i$. This is accomplished by performing an **L**–operation in $\mathcal{A}_{i,t}$ when $d_m \leq d_i$ and an **R**–operation in $\mathcal{A}_i$ when $d_m \not\leq d_i$. This clearly makes $U^{\mathcal{A}_m} \leq_T U^{\mathcal{A}_i}$ if $d_m \leq d_i$. It is also basically why the predicate P is computable: whenever a pair is put into P at t, one of its elements and so the pair itself is larger than t. Note, that the requirement $D_{e,n,m}$ is met as all lower priority requirements are initialized and so cannot put a number less than the use of the computation into $U^{\mathcal{A}_n}$. No higher priority requirement can act to put in any numbers by our choice of t'. Hence $\Psi_e^{U^{\mathcal{A}_n}}$ can not be the characteristic function for $U^{\mathcal{A}_m}$. Of course, all the $U^{\mathcal{A}_i}$ are c.e. by construction. If d_i is the least element of $\mathcal{D}$, then there are no diagonalization requirements of the form $D_{e,n,i}$ and so the only new numbers are put into $U^{\mathcal{A}_i}$ which is therefore computable as desired.

The argument that the requirements R_j are all satisfied by the node α on the true path associated with R_j is the same as in the construction of the previous section except that there are possible outcomes $0, 1, \ldots, i, \ldots, j$ corresponding to claim that the map $r^i_\alpha = \cup\{r^i_{\alpha,t} | t > t_0\}$ is a computable isomorphism from G_j to $\mathcal{A}_i$ where $\alpha\hat{\ }i$ is on the true path and t_0 is the least stage at which $\alpha\hat{\ }i$ is accessible but no node to its left is ever accessible again. (As before, if the $\alpha\hat{\ }w$ or $\alpha\hat{\ }\infty$ is on the true path then G_j is not isomorphic to $\mathcal{A}$.) $\square$

4 Applications

In this section we provide several applications of the techniques developed in the previous sections. The first application answers Question 1.8 posed by Goncharov. The second application answers Question 1.12 and thus giving a new solution to the Ash–Goncharov problem. The third application answers Question 1.13 about the connections between Scott families and categoricity. We begin by answering Question 1.8:

Theorem 4.1 *There exists a structure $\mathcal{B}$ which has exactly two computably enumerable presentations $\mathcal{B}_1$ and $\mathcal{B}_2$ such that the following properties hold:*

1. *$\mathcal{B}_1$ is a computably enumerable but not computable structure.*
2. *$\mathcal{B}_2$ is a computable structure.*
3. *Any computably enumerable presentation $\mathcal{C}$ of $\mathcal{B}$ is computably isomorphic to either $\mathcal{B}_1$ or $\mathcal{B}_2$.*

Proof. Consider the structure $\mathcal{A}$ constructed in the proof of Theorem 2.1. Let U be the unary relation in $\mathcal{A}$ for which $U^{\mathcal{A}_0}$ is computably enumerable but not computable and $U^{\mathcal{A}_1}$ is computable. Expand the structure $\mathcal{A}$ by adding to the language of $\mathcal{A}$ two predicate symbols **U** and **E**. Set the interpretation of **U** to be the unary predicate U constructed in Theorem 2.1. Set the interpretation of **E** to be a binary predicate E such that for all $x, y,\ (x, y) \in E$ if and only if $x \neq y$. Define $\mathcal{B}$ to be the expanded structure $(\mathcal{A}, U, E)$. Obviously, the structure $\mathcal{B}_0$ defined as $(\mathcal{A}_0, U^{\mathcal{A}_0}, E)$ is a computably enumerable but not computable presentation of $\mathcal{B}$. The structure $\mathcal{B}_1$ defined as $(\mathcal{A}_1, U^{\mathcal{A}_1}, E)$ is a computable presentation of $\mathcal{B}$. The structures $\mathcal{B}_0$ and $\mathcal{B}_1$ are isomorphic and are computably enumerable presentations of $\mathcal{B}$. These two presentations are not computably isomorphic. Let $\mathcal{C}$ be any other computably enumerable presentation of $\mathcal{B}$. The relation E is a computably enumerable relation in $\mathcal{C}$. The equality relation in $\mathcal{C}$ is also computably enumerable. Hence, the equality relation $\{(x, y)|x = y\}$ of the structure $\mathcal{C}$ is computable. Therefore, if we omit the predicate symbols U and E from the language of $\mathcal{C}$, then the structure $\mathcal{C}'$ with the predicates U and E omitted is computably isomorphic to either $\mathcal{A}_0$ or $\mathcal{A}_1$. Hence $\mathcal{C}$ is computably isomorphic to either $\mathcal{B}_1$ or $\mathcal{B}_2$. $\square$

Our next result provides a new solution to the Ash–Goncharov problem originally solved in [7]:

Theorem 4.2 *For each natural number $k \geq 2$ there exists a computably categorical structure $\mathcal{B}$ whose expansion by finitely many constants has exactly k many computable isomorphism types.*

Proof. By Theorem 3.1 there is a computable structure $\mathcal{A}$ with a unary relation U such that the following properties hold:

1. The structure $\mathcal{A}$ has exactly k many computable presentations $\mathcal{A}_1$, ..., $\mathcal{A}_k$ whose domains are pairwise disjoint.
2. For all distinct $i, j \in \{1, \ldots, k\}$, $U^{\mathcal{A}_i}$ and $U^{\mathcal{A}_j}$ are Turing incomparable.
3. The binary predicate P, such that for all (x, y), $(x, y) \in P$ if and only if there are distinct i, j with $x \in \mathcal{A}_i$ and $y \in \mathcal{A}_j$ for which the mapping $x \to y$ can be extended to an isomorphism from $\mathcal{A}_i$ to $\mathcal{A}_j$, is computable.

To see this, simply let $\mathcal{D}$ be a set of k pairwise incomparable elements. (So $d_i \neq d_j$ is the same as $d_i \not\leq d_j$.)

Now, define the desired structure $\mathcal{B}$ to be the graph

$$\mathcal{A}_1 + \ldots + \mathcal{A}_k$$

expanded by the predicate P and the equivalence relation E whose equivalence classes are $\mathcal{A}_1$, ..., $\mathcal{A}_k$. Clearly $\mathcal{B}$ is a computable structure. Now let $\mathcal{B}'$ be a computable presentation of $\mathcal{B}$. Let $\mathcal{A}'_1$ and $\mathcal{A}'_2$ be two equivalence classes in $\mathcal{B}'$. These two substructures of $\mathcal{B}'$ considered as graphs are isomorphic to $\mathcal{A}$. Hence $\mathcal{A}'_1$ is computably isomorphic to one of $\mathcal{A}_1$, ..., $\mathcal{A}_k$. Without lost of generality suppose that $\mathcal{A}'_1$ is computably isomorphic to $\mathcal{A}_1$ via a computable function $f_1 : \mathcal{A}_1 \to \mathcal{A}'_1$. If $\mathcal{A}'_2$ were computably isomorphic to $\mathcal{A}_1$ via computable function $f_2 : \mathcal{A}_1 \to \mathcal{A}'_2$, then we would be able to decide $U^{\mathcal{A}_1}$ in $\mathcal{A}_1$ as follows: x in $\mathcal{A}_1$ belongs to $U^{\mathcal{A}_1}$ if and only if $(f_1(x), f_2(x)) \in P$. Hence all $\mathcal{A}'_1$, ..., $\mathcal{A}'_k$ are pairwise noncomputably isomorphic. Hence $\mathcal{B}'$ is computably isomorphic to $\mathcal{B}$. Thus, $\mathcal{B}$ is computably categorical.

Let a be any designated element from $\mathcal{A}_1$. Consider the expanded structure $(\mathcal{B}, a)$ with a new constant for a. Let a_i and a_j be the image of a in $\mathcal{A}_i$ and $\mathcal{A}_j$, respectively. It follows that the structures

$$\mathcal{A}_1 + \ldots + (\mathcal{A}_i, a_i) + \ldots + \mathcal{A}_k$$

and

$$\mathcal{A}_1 + \ldots + (\mathcal{A}_j, a_j) + \ldots + \mathcal{A}_k$$

expanded by the predicate P and the equivalence relation E are isomorphic but not computably isomorphic. Thus, $(\mathcal{B}, a)$ has exactly k many computable isomorphism types. $\square$

An immediate consequence is

Corollary 4.3 *There exists a computably categorical structure without a Scott family.* $\square$

Now our goal is to strengthen this corollary so as to answer Question 1.13.

Theorem 4.4 *There exists a structure without a Scott family such that every expansion of the structure by a finite number of constants is computably categorical.*

Proof. The structure required to establish the theorem is constructed by coding certain (uniformly) computably enumerable families of sets of natural numbers.

Definition 4.5 *A family S of sets of natural numbers has a* **one-to-one computable enumeration** *if there is a bijection $f : \omega \to S$ such that $\{(i, x) | x \in f(i)\}$ is computably enumerable. We then call f a* **(computable) one-to-one enumeration** *of S.*

We wish to consider a standard preordering on the one-to-one computable enumerations of S that naturally induces an equivalence relation corresponding to computable isomorphism:

Definition 4.6 *A computable enumeration f of S is* **reducible to** *g, $f \leq g$, if there is a computable Φ such that $f = g\Phi$. If $f \leq g$ and $g \leq f$, then we say that f and g are* **equivalent**.

Note that if f is a one-to-one enumeration of S and $f = g\Phi$, then Φ is a permutation of ω and so $f \leq g$. Thus the equivalence classes of one-to-one enumerations are minimal elements in the induced partial ordering. These are the enumerations that we need to consider to define the family that supplies the structure required for the theorem. Informally, computable categoricity corresponds to there being a single such equivalence class and the dimension of a structure corresponds to the number of such classes.

Definition 4.7 *Two computable structures $\mathcal{A}$ and $\mathcal{B}$ are of the same* **computable isomorphism type** *if there is computable isomorphism taking $\mathcal{A}$ to $\mathcal{B}$. The* **dimension** *of a computable structure $\mathcal{A}$ is the number of its computable isomorphism types.*

Definition 4.8 *A computable sequence $D_0, D_1, \ldots$ of (canonical indices for) finite sets is a* **Scott sequence** *for a family S if the following properties hold:*

1. *For each D_i there exists exactly one $M \in S$, denoted by M_i, such that $D_i \subset M$.*

2. *The set $S \setminus \{M_0, M_1, \ldots\}$ is finite.*

From this definition the next lemma follows easily:

Lemma 4.9 *If S has a Scott sequence, then any two computable enumerations of S are equivalent.* $\square$

For any given family S, we want to construct a structure $\mathcal{A}_S$ such that $\mathcal{A}_S$ has a Scott family if and only if S has a Scott sequence. Thus, let S be a family of sets and let f be a one-to-one computable enumeration of S. We assume that each set in S has at least two elements and does not contain 0 or 1. Consider the following structure $\mathcal{A}_f$:

$$[f(0)] + [f(1)] + [f(2)] + [f(3)] + \ldots$$

We can assume that $\mathcal{A}_f$ is a computable structure constructed uniformly in f. Note that the set of all top elements of $\mathcal{A}_f$ is computable in every computable presentation of $\mathcal{A}_f$ (they are the elements with outdegree at least 2 and also the ones not part of a cycle.) The following lemma describes the relationship between S and $\mathcal{A}_f$.

Lemma 4.10 *The structure $\mathcal{A}_f$ satisfies the following conditions.*

1. *If g is a one-to-one computable enumeration of S, then $\mathcal{A}_f$ is isomorphic to $\mathcal{A}_g$.*

2. *The structure $\mathcal{A}_f$ is rigid, that is it does not have any nontrivial automorphisms.*

3. *If g is a one-to-one computable enumeration of S, then $\mathcal{A}_f$ is computably isomorphic to $\mathcal{A}_g$ if and only if f and g are equivalent.*

4. *The dimension of the structure $\mathcal{A}_f$ is equal to the maximal number of nonequivalent one-to-one computable enumerations of S.*

5. *The structure $\mathcal{A}_f$ has a Scott family if and only if S has a Scott sequence.*

Proof. To prove 1, first, note that for any pair $i, j \in \omega$ the graphs $\mathcal{A}_{f,i}$ and $\mathcal{A}_{g,j}$ are isomorphic if and only if $f(i) = g(j)$. Hence, since f and g are one-to-one enumerations of S, we can conclude that $\mathcal{A}_f$ is isomorphic to $\mathcal{A}_g$.

Any automorphism α of $\mathcal{A}_f$ must be the identity by the construction of $\mathcal{A}_f$ and the fact that f is a one-to-one mapping. This proves 2.

Suppose that f and g are equivalent. There exists a computable function Φ such that $f = g\Phi$. Hence $\mathcal{A}_f$ and $\mathcal{A}_g$ are computably isomorphic. Let $\mathcal{B}$ be a computable presentation of $\mathcal{A}_f$. Consider an effective sequence $e_0, e_1, e_2, \ldots$ without repetition of all top elements in $\mathcal{B}$. We define a one-to-one computable enumeration $f_{\mathcal{B}}$ of S as follows:

$$f_{\mathcal{B}}(i) = \{n|\ e_i \text{ is connected to a cycle of length } n\}.$$

It follows that $\mathcal{B}$ is computably isomorphic to $\mathcal{A}_g$ if and only if g is equivalent to $f_{\mathcal{B}}$

Part 4 follows from the proof of 3.

We are left to prove the last part of the lemma. Suppose that S has a Scott sequence $D_0, D_1, D_2, \ldots$. Without lost of generality we suppose that $D_i \subset f(i)$. The case $S \setminus \{M_0, M_1, \ldots\} \neq \emptyset$ can be derived easily from our considerations below simply using the fact that $S \setminus \{M_0, M_1, \ldots\} \neq \emptyset$ is finite. We have to prove that $\mathcal{A}_f$ has a Scott family. Take an $x \in A_f$. Find a top element d_i which is connected to x. Suppose that the length of the path which connects x with d_i is n. Define the following formula: $\psi(x) =$ [there exists a path of length n which connects x with a top element y such that for each $m \in D_i$ the element y is connected to a cycle of length m]. Now for every s–tuple $(x_1, \ldots, x_s)$ let $\phi_{(x_1,\ldots,x_s)}$ be $\psi(x_1)\&\ldots\&\psi(x_s)$. It is not hard to check that the sequence $\{\phi_{(x_1,\ldots,x_s)}\}$ is a Scott family for $\mathcal{A}_f$.

Now suppose for simplicity that $\mathcal{A}_f$ has a Scott family

$$\phi_0(x_1, \ldots, x_{n_0}), \phi_1(x_1, \ldots, x_{n_1}), \ldots$$

without parameters. The proof below will show that we do not lose any generality by making this assumption. Let $d_0, d_1, d_2 \ldots$ be an effective sequence of all top elements from $\mathcal{A}_f$. Let

$$\phi_{i_0}(x_0), \phi_{i_1}(x_1), \ldots$$

be an effective subsequence of the original sequence such that $\phi_{i_k}(d_k)$ holds for each $k \in \omega$. Since the formulas are all existential and the structure is computable, we can effectively find a finite substructure $\mathcal{B}_i$ of $\mathcal{A}_f$ such that $d_i \in B_i$ and $\phi_{i_k}(d_k)$ holds in $\mathcal{B}_i$. Define

$$D_i = \{n|\ d_i \text{ is connected to a cricle of length } n \text{ in substructure } \mathcal{B}_i\}.$$

Since we have a Scott family for $\mathcal{A}_f$ and since the structure $\mathcal{A}_f$ is rigid, we can see that the sequence $D_0, D_1, \ldots$ is a Scott sequence for family S. $\square$

Corollary 4.11 *Any two one-to-one computable enumerations of S are equivalent if and only if $\mathcal{A}_f$ is computably categorical.* $\square$

Now, to prove the Theorem 4.4 it suffices, by the lemma, to build a computably enumerable family S of sets without a Scott sequence any two computable one-to-one enumerations of which are equivalent.

Lemma 4.12 *There is a computably enumerable family S of sets with no Scott sequence any two computable one-to-one enumerations of which are equivalent.*

Proof. In order to build a such family S and its one-to-one enumeration f, we need to satisfy the following requirements:

$$D_e: \quad F_e \text{ is not a Scott sequence for } S,$$

$$R_j: \quad g_j \equiv f \text{ or } g_j \text{ is not a one-to-one enumeration of } S,$$

where each $\{g_j\}_j$ is a computable sequence of all potential one-to-one enumerations of a family of sets and $\{F_e\}_e$ is a computable sequence of all potential Scott sequences for S. Now one can see that these requirements are similar to the requirements for constructing a computable structure in Theorem 2.1. We briefly explain how to meet one R_j and all D_e. The verifications for this much and the modifications needed to prove the full theorem are similar to those for Theorem 2.1 and are left to the reader.

We set $g = g_i$. For $t, n \in \omega$ and $n \leq t$, let $g_t(n)$ be

$$\{x| \ \ x \leq t \text{ and } x \text{ appears in } g(n) \text{ in fewer than } t+1 \text{ steps of a fixed computation procedure for } g\}.$$

Our construction proceeds by stages. At stage t, we use the following notions and terminology similar to those in the proof of Theorem 2.1.

1. Enumeration f_t. This is an approximation to the enumeration f that the construction is building. That is, for each $i \in \omega$, we will have

$$f(i) = \bigcup_t f_t(i).$$

2. The family S_t. The function f_t enumerates a family denoted by S_t.

3. To each $\mathcal{F}_e$, we assign a set $\{c_e\}$ and a number p_e, called **witnesses**, such that $D = \{c_e, p_e | e \in \omega\}$ is a coinfinite computable set and we set $f_0(c_e) = \{p_e\}$. One of the goals of the construction is to meet D_e on one of these numbers.

4. Potential Reduction Function r_t. The map r_t, is the function which potentially reduces g_t to f_t at stage t. The function r_t can extend the previous potential reduction r_{t-1}. If r_t does not extend the previous potential reduction, then we say that the construction **changes its potential reduction**.

5. Let $i \in dom(f_t)$. The construction will guarantee that

$$f_t(i) \setminus \bigcup \{ \ f_t(j) \mid j \neq i \,\&\, j \in dom(f_t)\} \neq \emptyset.$$

Thus at stage t, each $f_t(i)$ possesses an element which does not belong to $f_t(j)$ for $i \neq j$. The purpose of this property is to ensure that f will be a one-to-one enumeration. We will also guarantee that at the end $f(i) \backslash f(j) \neq \emptyset$ for $i \neq j$.

6. A special g**–set**. The construction needs to pick a set $g(s_g)$ in the enumeration g which is called a **special** g**–set**. If there exist infinitely many stages at which the construction changes its reduction, then the set $g(s_g)$ becomes infinite, all sets in f contained in $g(s_g)$ are finite, and therefore g is not a one-to-one enumeration of S. On the other hand, if after some stage the construction never changes its reduction and g is a one-to-one enumeration of the family S, then g will be equivalent to f.

7. Special Numbers s^f_t. The construction uses these numbers so that $r_t(s_g) = s^f_t$. Thus $f_t(s^f_t)$ is the set in f_t which, at stage t, corresponds to $g(s_g)$. Moreover, if g recovers at stage t (as defined below), then these numbers satisfy the following properties.

1. If the construction does not change its previous reduction at stage t, then $s^f_{t+1} = s^f_t$.

2. If the construction changes its reduction at stage t, then $s^f_{t+1} \neq s^f_t$.

3. If, after some stage, the construction never changes its reduction, recovers infinitely often, and g is a one-to-one enumeration of the family S, then the construction guarantees that the set $f(lim_{n\to\infty} s^f_n)$ becomes infinite.

8. Marking with $\Box_w$ and Recovery. If, for a f_t–index x, there exists a $y \leq t$ such that $g_t(y) \subset f_t(x)$ and for all $z \neq x$ the pair $g_t(y)$ is not contained in $f_t(z)$, then we say that $g_t(y)$ **is covered by** $f_t(x)$, or equivalently, $f_t(x)$ **covers** $g_t(y)$. During the construction some f_t–indices will be marked with a special symbol $\Box_w$ called a **mark**. We say that the enumeration g **recovers at stage** t, or equivalently that stage t is a **recovery stage**, if for each f_t–index x marked with a $\Box_w$, there exists a unique y such that $f_t(x)$ covers $g_t(y)$. We use the notion of recovery to show that if g is a one-to-one enumeration of the family S, then g is equivalent to f. The idea is the following. Suppose that g is a one-to-one enumeration of S. By construction, each $f_t(x)$ marked with a $\Box_w$ waits to cover a set in g. As soon as g recovers at a stage $t_1 \geq t$ and a unique g_{t_1}–index y is found such that $f_t(x)$ covers $g_{t_1}(y)$, the construction defines $r_{t_1}(x) = y$ and then attempts to guarantee that $g(y) = f(x)$. If the enumeration does not recover at stage t, then we say that g is in the **waiting state**. If g is always in the waiting state after t, then, by construction, g will not be a one-to-one enumeration of S.

Now we need a definition for an operation which corresponds to the **L**–operation in the proof of Theorem 2.1. This operation will be needed to meet requirements D_e.

Definition 4.13 *Let $X_k, \ldots, X_1$ be a sequence of sets. The* **L***–operation applied to this sequence gives, by definition, the sequence*

$$X_k \bigcup X_{k-1}, \ldots, X_1 \bigcup X_k.$$

We also say that the sets $X_k, \ldots, X_1$ **participated** *in the* **L***–operation.*

Now we will describe the construction for satisfying all D_e and one R.

Stage 0. Let $dom(f_0) = D \bigcup \{a\}$, where a is a new number not in D. Let p also be a new number. We set $f_0(a) = \{p\}$. Put a mark $\Box_w$ on a. Let the reduction r_0 be the empty function. Put g into the waiting state. When we first have a recovery stage we will define s_g so that $s_0^f = a$.

Stage $t+1$: Substage 1: If this is not a recovery stage we go on to substage 2. If it is a recovery stage, we proceed as follows:

Action. Define the reductions r_{t+1}^f on the indices marked with a $\Box_w$ as follows: Put $r_{t+1}^f(x) = y$ if and only if $g_{t+1}(y)$ is covered by $f_{t+1}(x)$. We now have the following two cases.

Case 1. Suppose that r_{t+1}^f extends the previous reduction r_t^f. In this case, set $s_{t+1}^f = s_t^f$.

Case 2. Suppose that r_{t+1}^f does not extend the previous reduction r_{t+1}^f. In this case, the construction changes its reduction. Note that $s_{t+1}^f \neq s_t^f$.

Substage 2: Compute $F_{e,t+1}$ for all $e \leq t+1$. Let $f_{t+1}(a) = \{b\}$ where a, b are new numbers. Find the least $e \leq t+1$ for which $F_{e,t+1}(q_e) = f_t(c_e)$ for some $q_2 \leq t+1$ such that we have not yet acted for D_e and one of the following cases is satisfied (If there is no such e, then go on to the next stage).

1. There does not exist j_1 such that $r_{t+1}(c_e) = j_1$.

2. This is a recovery stage and, for some j_1, $r_{t+1}(c_e) = j_1$.

Action for D_e. Suppose that the first case holds. Let $f_{t+1}(c_e) = \{p_e, p_e'\}$, $f_{t+1}(u) = \{p_e, p_e''\}\}$ where p_e', p_e'', and u, are new numbers. Note that we have successfully met the potential reduction D_e. Also the family S_t has been changed to a new family S_{t+1} in such a way that f_t can naturally be extended to an enumeration f_{t+1} of the family S_{t+1}.

Suppose that the second case holds. In this case, we have two subcases.

Subcase 2.1. There exist distinct numbers a_1, a_2, a_3 not in D such that for some x_1, x_2, x_3, $r_t(a_i) = x_i$, and none of the numbers a_i have previously participated in any **L**–operation. In this case, consider the sequence

$$a_1, c_e, a_2, s_t^f, a_3.$$

and the corresponding finite sequence

$$f_t(a_1), f_t(c_e), f_t(a_2), f_t(s_t^f), f_t(a_3).$$

Perform **L**–operations on the sequence. Put into each of these changed sets new elements. Thus the family S_t has been changed to a new family S_{t+1} in such a way that f_t can naturally be extended to an enumeration f_{t+1} of the family S_{t+1}. Put a mark $\Box_w$ on each f_{t+1}–index that participated in the **L**-operation, on a, on the least number x which has not yet been marked with a $\Box_w$. Go on to the next stage.

Subcase 2.2. Suppose that the previous subcase does not hold.

In this case, take distinct new elements a_1, a_2, a_3. Set $f_t(a_j)$ to be $\{u_j\}$, where u_j are new distinct numbers, $j = 1, 2, 3$. Add these sets to S_t. Thus, the family S_t has been changed to a new family S_{t+1} in such a way that f_t can naturally be extended to an enumeration f_{t+1} of the family S_{t+1}.

Substage 3: If this is a recovery stage, put a mark $\Box_w$ on each new f_{t+1}–index and on the least number x which has not yet been marked with a $\Box_w$. In any case, we now go on to the next stage.

This concludes the description of the construction.

For each $i \in \omega$, define $f(i) = \bigcup_t f_t(i)$. Define the family S by $S = \{f(i) | i \in \omega\}$. We leave to the reader the verification of the correctness of this construction as well as the details of the general construction. $\Box$

References

[1] C. J. Ash, and A. Nerode, Intrinsically Recursive Relations, in [8], 26-41.

[2] C. J. Ash, Categoricity in the Hyperarithemetical Degrees, *Ann. Pure and Appl. Logic* **34** (1987), 1-14.

[3] C. J. Ash and J. Knight, Pairs of Recursive Structures, *Ann. Pure and Appl. Logic* **46** (1990), 211-234.

[4] C. J. Ash, J. Knight, M. Manasse and T. Slaman, Generic Copies of Countable Structures, *Ann. Pure and Appl. Logic* **42** (1989) 195-205.

[5] C. J. Ash, P. Cholak and J. Knight, Permitting, Forcing, and Copies of a Given Recursive Relation, to appear.

[6] D. Cenzer and J. Remmel, Polynomial Time versus Recursive Models, *Ann. Pure and Appl. Logic* **54** (1991), 17-58.

[7] P. Cholak, S. S. Goncharov, B. Khoussainov and R. A. Shore, Computably Categorical Structures and Expansions by Constants, *Journal of Symbolic Logic*, to appear.

[8] J. N. Crossley (ed.), *Aspects of Effective Algebra, Proceedings of a Conference at Monash University, 1979*, Upside Down A Book, Yarra Glen, Australia, 1981.

[9] Yu. Ershov and S. S. Goncharov (eds.), *Logic Notebook*, Novosibirsk University, 1986.

[10] Yu. Ershov, S. S. Goncharov, A. Nerode and J. Remmel (eds.), *Handbook of Computable Algebra*, to appear.

[11] A. Frölich and J. Shepherdson, Effective Procedures in Field Theory, Philos. *Trans. Roy. Soc., London*, ser. A **248** (1955), 432-487.

[12] S. S. Goncharov, Autostability and Computable Families of Constructivizations, *Algebra and Logic* **14** (1975), 392-409.

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

[14] S. S. Goncharov, The Problem of the Number Of Non-Self-Equivalent Constructivizations, *Algebra and Logic* **19 (**1980), 411-414.

[15] S. S. Goncharov, Effectively Infinite Classes of Weak Constructivizations of Models, *Algebra and Logic* **32** (1993), 342-360.

[16] S. S. Goncharov, Limiting Equivalent Constructivizations (Russian), *Tr. Inst. Mat. Sib. Akad. Nauk SSSR* **2**, 1982, 4-12.

[17] S. S. Goncharov and V. D. Dzgoev, Autostability of Models, *Algebra and Logic* **19** (1980), 28 -38.

[18] S. S. Goncharov and B. Khoussainov, On the Spectrum of Degrees of Decidable Relations, *Doklady Mathematics* **55** (1997), 55-57.

[19] V. Harizanov, Some Effects of Ash–Nerode and Other Decidability Conditions on Degree Spectra, *Ann. Pure and Appl. Logic* **55** (1991), 51-65.

[20] V. Harizanov, The Possible Turing Degree of Nonzero Member in Two Element Degree Spectrum, *Ann. Pure and Appl. Logic* **60** (1993), 1-30.

[21] V. Harizanov, Pure Recursive Model Theory, in *Recursive Mathematics*, Yu. Ershov, S. Goncharov, A. Nerode, and J. Remmel (eds.), to appear.

[22] G. Hird, Recursive Properties of Relations on Recursive Models, Ph.D. Thesis, Monash University, 1983.

[23] D. Hirschfeldt, On the Degree Spectrum Problem, in preparation.

[24] J. Knight, A Metatheorem for Construction by Finitely Many Workers, *Journal of Symbolic Logic* **55** (1990), 787-804.

[25] J. Knight, Constructions for Transfinitely Many Workers, *Ann. Pure and Appl. Logic* **46** (1990), 211-234.

[26] J. Knight, Degrees Coded in Jumps Orderings, *Journal of Symbolic Logic* **51** (1986), 1034-1042.

[27] B. M. Khoussainov, Spectrum of Algorithmic Dimensions of Homomorphisms of Models, *Algebra and Logic* **?? (**1992), ??.

[28] B. Khoussainov, Recursive Unary Algebras and Trees, *Ann. Pure and Appl. Logic* **67** (1994), 213-268.

[29] B. Khoussainov and A. Nerode, Automatic Presentations of Structure, in *Logic and Computational Complexity, Proceedings of the International Conference on LCC*, D. Leivant (ed.), LNCS **960**, Springer- Verlag, Berlin, 367-393.

[30] O. V. Kudinov, An Autostable 1-Decidable Model without a Computable Scott Family of $\exists$-Formulas, *Algebra and Logic* **35** (1996), 255-260.

[31] A. I. Mal'cev, Recursive Abelian Groups, *Dokl. Akad. Nauk SSSR* (Russian) **146** (1962), 1009-1012.

[32] A. I. Mal‘cev A. I., Constructive Algebras, *Uspekhi Matem. Nauk.* **16** (1961), 3-60, tr. in *Russian Math. Surveys* **16** (1961), 77-129.

[33] G. Metakides and A. Nerode, Recursion Theory and Algebra, in *Algebra and Logic*, ed. J. Crossley, Lecture Notes in Mathematics **450**, Springer-Verlag, Berlin, 1975, 209-219.

[34] T. Millar, The theory of Recursively Presentable Models, Ph.D. Dissertation, Cornell University, 1976.

[35] T. Millar, Recursive Categoricity and Persistence, *Journal of Symbolic Logic* **51** (1986), 430-434.

[36] A. Nurtazin, Strong and Weak Constructivizations and Computable Families, Algebra and Logic **13** (1974), 177-184.

[37] M. Rabin, Computable Algebra: General theory and Theory of Computable Fields, Trans. Am. Math. Soc. **95**, 1960, 341-360

[38] J. B. Remmel, Recursive Isomorphisms Types of Recursive Boolean Algebras, *Journal of Symbolic Logic* **46**, 1981, 572-593.

[39] J. Remmel, Recursive Boolean Algebras, in *Handbook of Boolean Algebras*, J. D. Monk (ed.), v.3, North Holland, Amsterdam, 1990, 1097-1066 .

[40] J. B. Remmel and J. N. Crossley, The Work of Anil Nerode: A Retrospective, in *Logical Methods*, J. N. Crossley, J. B. Remmel, R. A. Shore and M. E. Sweedler (eds.), Birkhäuser, Boston, 1993, 1-91.

[41] H. Rogers Jr., *Theory of Recursive Function and Effective Computability*, New York, 1967.

[42] G. Sacks, *Saturated Model Theory*, W. A. Benjamin, Reading, Mass. 1972.

[43] G. Sacks, *Higher Recursion Theory*, Springer-Verlag, Berlin, 1990.

[44] V. L. Selivanov, Numerations of Families of General Recursive Functions, *Algebra and Logic* **15** (1976), 205-226.