COMPUTABLE TREES OF SCOTT RANK ω_1^{CK}, AND COMPUTABLE APPROXIMATION

W. CALVERT, J. F. KNIGHT, AND J. MILLAR

Abstract. Makkai [10] produced an arithmetical structure of Scott rank ω_1^{CK}. In [9], Makkai's example is made computable. Here we show that there are computable trees of Scott rank ω_1^{CK}. We introduce a notion of "rank homogeneity". In rank homogeneous trees, orbits of tuples can be understood relatively easily. By using these trees, we avoid the need to pass to the more complicated "group trees" of [10] and [9]. Using the same kind of trees, we obtain one of rank ω_1^{CK} that is "strongly computably approximable". We also develop some technology that may yield further results of this kind.

1. Introduction

The notion of Scott rank comes from the Scott Isomorphism Theorem [16].

Theorem 1.1 (Scott). *Let $\mathcal{A}$ be a countable structure (for a countable language L). Then there is an $L_{\omega_1\omega}$ sentence σ whose countable models are just the copies of $\mathcal{A}$.*

In the proof, Scott assigned ordinals to tuples in $\mathcal{A}$, and to $\mathcal{A}$ itself. For simplicity, we suppose that the language of $\mathcal{A}$ is finite, and the substructure of $\mathcal{A}$ generated by a finite subset is finite. We begin as Scott did, with a family of equivalence relations on tuples.[1]

Definition 1.
(1) $\overline{a} \equiv^0 \overline{b}$ *if* $\overline{a}$ *and* $\overline{b}$ *satisfy the same quantifier-free formulas,*
(2) *for* $\alpha > 0$, $\overline{a} \equiv^\alpha \overline{b}$ *if for all* $\beta < \alpha$, *for all* $\overline{c}$, *there exists* $\overline{d}$, *and for all* $\overline{d}$, *there exists* $\overline{c}$, *such that* $\overline{a},\overline{c} \equiv^\beta \overline{b},\overline{d}$.

Definition 2. *The* Scott rank *of a tuple* $\overline{a}$ *is the least ordinal* β *such that for all* $\overline{b}$, $\overline{a} \equiv^\beta \overline{b}$ *implies* $(\mathcal{A},\overline{a}) \cong (\mathcal{A},\overline{b})$. *The* Scott rank of the structure $\mathcal{A}$, *denoted by* $SR(\mathcal{A})$, *is the least ordinal* α *greater than the Scott rank of any tuple in* $\mathcal{A}$.

For a countable structure $\mathcal{A}$, $SR(\mathcal{A})$ is a countable ordinal. If $\mathcal{A}$ is computable (or hyperarithmetical), then $SR(\mathcal{A}) \leq \omega_1^{CK}+1$ (see [12]). The following result is well known.

Theorem 1.2 (Folklore). *Let $\mathcal{A}$ be a computable (or hyperarithmetical) structure.*
(1) $SR(\mathcal{A})$ *is a computable ordinal if there is some* β *such that the orbits of all tuples are defined by computable* Π_β *formulas,*

The first author was partially supported by NSF Grants DMS 0139626 and DMS 0353748. The third author was partially supported by NSF Grant DMS 0102052.

[1] Our equivalence relations differ slightly from Scott's. In Clause 2, Scott extended by single elements c and d, where we use tuples. It is easy to check that $\equiv^{\omega\alpha}$ is the same for both definitions.

(2) $SR(\mathcal{A}) = \omega_1^{CK}$ *if the orbits of all tuples are defined by computable infinitary formulas, but there is no bound on the complexity, as in 1,*
(3) $SR(\mathcal{A}) = \omega_1^{CK} + 1$ *if there is a tuple whose orbit is not defined by any computable infinitary formula.*

Proof. Let $\overline{a}$ be a tuple in $\mathcal{A}$. For each computable ordinal α, we have a computable infinitary formula defining the $\equiv^\alpha$-class of $\overline{a}$. These formulas come from Scott, and the complexity can be bounded in terms of α. If $\overline{a}$ has computable Scott rank α, then the formula defining the $\equiv^\alpha$-class of $\overline{a}$ defines the orbit. It is easy to show by induction that $\overline{a} \equiv^\beta \overline{b}$ implies that $\overline{a}$ and $\overline{b}$ satisfy the same Π_β formulas, in particular, the computable Π_β formulas. From this it follows that if the orbit of $\overline{a}$ is defined by a computable Π_β formula, then $\overline{a}$ has Scott rank at most β.

If the orbits are all defined by computable Π_β formulas, then $SR(\mathcal{A})$ is at most $\beta + 1$. If the orbits are defined by computable infinitary formulas, then all tuples have computable Scott rank. If we cannot bound the complexity of the formulas defining the orbits, then we cannot bound the Scott ranks of the tuples, so $SR(\mathcal{A}) = \omega_1^{CK}$. If there is some tuple $\overline{a}$ whose orbit is not defined by any computable infinitary formula, then none of Scott's formulas defines the orbit, so $\overline{a}$ has Scott rank ω_1^{CK}, and then $SR(\mathcal{A}) = \omega_1^{CK} + 1$.

□

There are several definitions of Scott rank in use, some differing from Scott's much more dramatically than ours. While the ordinals assigned may differ, if one definition assigns computable rank to a particular structure $\mathcal{A}$, then so do the other definitions. There are familiar examples of computable structures having various computable ranks. In particular, there are examples of computable well orderings, superatomic Boolean algebras, and reduced Abelian p-groups having arbitrarily large computable Scott ranks. The reader may wish to verify that $SR(\omega) = 2$, and $SR(\omega^n) = n + 1$.

Harrison [8] showed that there is a computable ordering of type $\omega_1^{CK}(1 + \eta)$. The Harrison ordering has Scott rank $\omega_1^{CK} + 1$. There are related examples, also of Scott rank $\omega_1^{CK} + 1$. In particular, there is the *Harrison Boolean algebra*, which is the interval algebra of the Harrison ordering, and there are the *Harrison Abelian p-groups*, which have length ω_1^{CK}, all infinite Ulm invariants, and divisible part of infinite dimension. (See [6] for more about these structures and the orbits witnessing the high rank.)

Makkai [10] produced an arithmetical structure of Scott rank ω_1^{CK}. In [9], it is shown that there is a computable structure of rank ω_1^{CK}. The examples in Makkai's paper and in [9] are quite complicated, involving what Makkai calls "group trees" (Morozov [11] calls them "polygons"). In the present paper, we show that there are computable trees of rank ω_1^{CK}.

1.1. **Approximations.** One natural notion of approximability among computable structures would be that $\mathcal{A}$ is approximable if for any computable infinitary sentence σ true in $\mathcal{A}$, there is some computable $\mathcal{B} \not\cong \mathcal{A}$ such that $\mathcal{B} \models \sigma$. In practice, the following stronger notion seems more useful.

Definition 3. *A structure $\mathcal{A}$ is* strongly computably approximable *if for every Σ_1^1 set S, there is a uniformly computable sequence $(\mathcal{A}_n)_{n\in\omega}$ such that for $n \in S$, $\mathcal{A}_n \cong \mathcal{A}$, and for $n \notin S$, $\mathcal{A}_n$ has computable rank.*

The Harrison ordering is strongly computably approximable, using computable well orderings. The Harrison Boolean algebra, and the Harrison Abelian p-group are also strongly computably approximable. For the Harrison Boolean algebra, the approximating structures are superatomic Boolean algebras. and for the Harrison Abelian p-group, they are reduced Abelian p-groups of computable ordinal length, with all infinite Ulm sequence.

Goncharov and the second author [7] had asked whether all structures of non-computable Scott rank are strongly computably approximable. The problem remains open. In the present paper, we show that there is a computable tree of rank ω_1^{CK} which is strongly computably approximable. This is the first example of a structure of Scott rank ω_1^{CK} for which strong computable approximability is actually known.

We have given some background on Scott rank. Further background material, on computable infinitary sentences, the Barwise-Kreisel Compactness Theorem, Kleene's system of ordinal notations, and Ash's α-systems, may be found in [4]. In the remainder of Section 1, we give some definitions for describing trees, and in particular introduce the notion of *rank homogeneity.* In Section 2, we construct a computable tree with some special properties. In Section 3, we show that any tree with all of these properties has Scott rank ω_1^{CK}. We also describe some back and forth relations on tuples from possibly different trees, which will be useful when we consider approximating families. In Section 4, we construct a tree with the properties in Section 2, together with a family of approximating trees. In Section 5, we give a general result with sufficient conditions for a structure to be strongly computably approximable, and we apply this result to show that the special tree from Section 4 is strongly computably approximable.

1.2. **Describing trees.** We give some definitions useful for describing trees. Our trees are isomorphic to subtrees of $\omega^{<\omega}$. For the language, we use a single unary function symbol, interpreted as the predecessor. Our trees grow down. We write $\emptyset$ for the top node, and we think of $\emptyset$ as its own predecessor. For a tree T, we write T_n for the set of elements at level n.

Definition 4. *Let T be a tree. We define the* tree rank *(sometimes called* foundation rank*) for elements of T, and for T itself, below. We denote the rank of the element a, and the tree T, by $rk(a)$, $rk(T)$.*

- *If a has no successors, then $rk(a) = 0$.*
- *For an ordinal $\alpha > 0$, we say that $rk(a) = \alpha$ if all successors of a have ordinal rank, and α is the least ordinal greater than the ranks of the successors.*
- *If a does not have ordinal rank, then we write $rk(a) = \infty$. We adopt the convention that ∞ is greater than any ordinal rank.*
- *$rk(T)$ is $rk(\emptyset)$.*

Remark: It is easy to see that $rk(T) = \infty$ iff T has a path, and $rk(a) = \infty$ iff there is a path through a.

We define some special properties of trees. We will obtain a tree of Scott rank ω_1^{CK} by showing that there exist trees with these special properties and that every tree with these properties has Scott rank ω_1^{CK}. The first property was defined in [9].

Definition 5. *A tree T is* thin *if for all n, the set of ordinal ranks of elements of T_n has order type at most $\omega \cdot n$.*

As in [9], we use thinness in the following way.

Proposition 1.3. *Suppose T is a computable (or hyperarithmetical) tree. If T is thin, then for each n, there is a computable ordinal α_n such that for all $a \in T_n$, if $rk(a) \geq \alpha_n$ then $rk(a) = \infty$.*

In the earlier constructions [10], [9], the "group tree" was obtained by starting with a tree, putting a group structure on each level, and then using the groups to derive a new, more homogeneous, tree structure, together with a family of unary operations. The main idea in the present paper is the isolation of a homogeneity condition which may be imposed directly on trees to achieve the desired effect. Here is the definition.

Definition 6. *A tree T is* rank homogeneous *if for all n and for all $a \in T_n$,*

(1) *for all ordinals $\beta < rk(a)$ (where $rk(a)$ may be an ordinal or ∞), if some $b \in T_{n+1}$ has rank β, then a has infinitely many successors of rank β,*
(2) *if $rk(a) = \infty$ (so that a has a successor of rank ∞), then a has infinitely many successors a' with $rk(a') = \infty$.*

For a rank homogeneous tree T, we have the following invariants. Let $R_n(T)$ denote the set of ranks (possibly including ∞) of elements of T_n.

Proposition 1.4. *Suppose T and T' are rank homogeneous trees such that for all n, $R_n(T) = R_n(T')$. Then $T \cong T'$.*

Proof. Let $\mathcal{F}$ be the set of rank-preserving isomorphisms f between finite subtrees of T and T'. It is not difficult to see that $\mathcal{F}$ has the back-and-forth property. We give the "forth" part of the argument—the "back" part is the same. If we extend $dom(f)$, adding a new successor x', where $x \in dom(f)$, then there is a corresponding new successor y' for $y = f(x)$ such that $rk(x') = rk(y')$.

□

The proof above lets us describe the orbits of tuples in a rank homogeneous tree T. There is an automorphism of T taking $\overline{a}$ to $\overline{b}$ just in case the function taking $\overline{a}$ to $\overline{b}$ extends to a rank-preserving isomorphism f from the finite subtree generated by $\overline{a}$ onto that generated by $\overline{b}$.

2. Construction of a special tree

In this section, our goal is to prove the following.

Theorem 2.1. *There exists a computable thin rank homogeneous tree T, with a path, but no hyperarithmetical path.*

Proof. There is a Π^1_1 set Γ of computable infinitary sentences describing the tree we want. In [9], there is a Π^1_1 set Γ_0 describing a computable thin tree T such that $rk(T) = \infty$ but T has no hyperarithmetical path. The sentences say that T is a computable tree, $rk(T) \geq \alpha$, for all computable ordinals α, T has no hyperarithmetical path, and T is thin.

To say that T is thin, we add extra symbols, so that we can talk about a map ρ_n from a subset of T_n to an ordering $\mathcal{L}_n$ of type $\omega \cdot n$. We want to say that ρ_n maps the elements of T_n of ordinal rank to $\mathcal{L}_n$ such that $rk(x) < rk(y)$ iff $\rho_n(x) < \rho_n(y)$. We must be careful. There is no computable infinitary formula saying $rk(x) < rk(y)$. We can only say things like $rk(x) = \alpha$, or $rk(x) > \alpha$, for various computable ordinals α.

We put into the Π^1_1 set Γ_0 a computable infinitary sentence for each n and each computable ordinal α, saying that for all $x \in T_n$, if $rk(x) = \alpha$, then $\rho_n(x)$ is defined, with value in $\mathcal{L}_n$, and for all $y \in T_n$, if $\rho_n(y)$ is also defined, then $\rho_n(y) \leq \rho_n(x)$ iff $rk(y) = \beta$ for some $\beta \leq \alpha$. Assuming that T is a computable (or hyperarithmetical) tree, these sentences say what we want. If $x \in T_n$, has ordinal rank, then that rank must be a computable ordinal, and if x and y are two elements of T_n of computable rank, then the sentences say that $rk(x) < rk(y)$ iff $\rho_n(x) < \rho_n(y)$. Therefore, the order type of the set of ordinal ranks is at most $\omega \cdot n$.

To form the set Γ, we add to Γ_0 sentences guaranteeing that the tree T is rank homogeneous. The new sentences say, for computable ordinals β, for all $x \in T_n$ such that $rk(x) > \beta$,

(1) if there exists $y \in T_{n+1}$ of rank β, then x has infinitely many successors of rank β,
(2) if x has some successor of rank $\geq \beta$, then it has infinitely many.

It may not be immediately clear that these sentences are sufficient to guarantee rank homogeneity. Suppose T satisfies the sentences in Γ, and $x \in T_n$. If $rk(x) = \infty$, then x has a successor of rank ∞. We must show that there are infinitely many. Clause 2 guarantees that for all computable ordinals β, x has infinitely many successors of rank $\geq \beta$. If there is a bound on the computable ordinal ranks of these successors, then x must have infinitely many successors of rank ∞. In a thin tree, there is such a bound. (If there were no such bound, we could still show that x has infinitely many successors of rank ∞, using the fact that T is computable.)

To prove Theorem 2.1, it is enough to produce a model of Γ, for which we use Barwise-Kreisel Compactness. We must show that every Δ^1_1 subset of Γ has a model. For this, we show that for each computable ordinal α, there is a computable thin, rank homogeneous tree T_α of computable rank at least $\omega\alpha$.

The following two lemmas are proved in [9]. The first lemma gives a tree of rank $\omega\alpha$, with a computable function labeling the nodes with their ranks. The labels involve ordinal notations on a fixed path P through $\mathcal{O}$. To indicate that a node has tree rank $\omega\beta + n$, we use the label (b, n), where b is the unique notation for β in P.

Lemma 2.2. *For each computable ordinal α, there is a computable tree of rank $\omega\alpha$, with a computable rank function taking values as above.*

The next lemma says that we can replace the tree from Lemma 2.2 by a thin tree, so that there is still a computable rank function.

Lemma 2.3. *For each computable ordinal α, there is a computable thin tree T of computable rank at least $\omega\alpha$, such that T has a computable rank function.*

The next lemma says that we can modify the tree from Lemma 2.3 to make it rank homogeneous, without changing the ranks represented at each level.

Lemma 2.4. *Suppose T is a computable tree of computable rank, with a computable rank function. There is a computable rank homogeneous tree T', also with a computable rank function, such that for all n, $R_n(T) = R_n(T')$.*

Proof. At level 0 of T', we put $\emptyset$, with the same label as in T. Given $x \in T'_n$ and $y \in T_{n+1}$ such that $rk(y) < rk(x)$, we give x infinitely many successors of the same rank as y. Note that for any possible labels $(b, n), (b', n')$ in $P \times \omega$, we can effectively determine whether $(b, n) < (b', n')$.

□

The three lemmas above clearly give the following.

Lemma 2.5. *For each computable ordinal α, there is a computable thin, rank homogeneous tree T_α of computable rank at least $\omega\alpha$.*

We saw above that this lemma is enough to complete the proof of Theorem 2.1.

□

3. Orbits and back-and-forth relations

In this section, our first goal is to show that if T is a computable thin rank homogeneous tree, with a path, but with no hyperarithmetical path, then T has Scott rank ω_1^{CK}. By Theorem 1.2, to show that a computable structure has Scott rank ω_1^{CK}, it is enough to show that all tuples have computable Scott rank, but there is no computable bound on the Scott ranks of the tuples—or equivalently, that the orbits of all tuples are defined by computable infinitary formulas, but there is no computable bound on the complexities of these formulas. A second goal of the section is to prepare for the results on computable approximation by considering tuples in a pair of rank homogeneous trees, and defining a family of "back-and-forth" relations which can be calculated in terms of tree ranks.

The lemma below, generalizing Proposition 1.4, describes the orbits in a rank homogeneous tree T in terms of tree ranks. The proof is identical to that for Proposition 1.4.

Lemma 3.1. *Let T be a rank homogeneous tree. Then $\overline{a}$ and $\overline{b}$ are in the same orbit if the function taking $\overline{a}$ to $\overline{b}$ extends to an isomorphism f between the subtrees generated by $\overline{a}$ and $\overline{b}$ such that for all $x \in dom(f)$, $rk(x) = rk(f(x))$.*

Using Lemma 3.1, together with Proposition 1.3, we get the following.

Lemma 3.2. *Let T be a computable thin rank homogeneous tree. Then the orbits of tuples in T are all definable by computable infinitary formulas.*

Proof. To define the orbit of a tuple $\overline{a}$, we describe the subtree generated by $\overline{a}$, and give the tree ranks for all elements of this subtree. We must think how to express $rk(x) = \infty$ using computable infinitary formulas. By thinness, for each n there is a computable ordinal α_n bounding the ordinal tree ranks of elements at level n. Thus for x at level n, we can say $rk(x) = \infty$ just by saying $rk(x) \geq \alpha_n$. We do *not* have a computable infinitary formula saying for arbitrary x (at unspecified level) that $rk(x) = \infty$.

□

We have shown that if T is a computable thin rank homogeneous tree, then all tuples have computable Scott rank, so $SR(T) \leq \omega_1^{CK}$. We must show that if, in addition, T has a path, but no hyperarithmetical path, then $SR(T) \geq \omega_1^{CK}$; i.e., there is no computable bound on the Scott ranks of tuples. Note that for every computable ordinal α, there exist elements $x \in T$ such that $rk(x) > \alpha$. Since T has no hyperarithmetical path, this must happen below any x with $rk(x) = \infty$.

We define a family of relations $\sim^\alpha$ on tuples in T (of the same length).

Definition 7. *Let T be a rank homogeneous tree. For an ordinal α, we say that $(T,\overline{a}) \sim^\alpha (T,\overline{b})$ if the function taking $\overline{a}$ to $\overline{b}$ extends to an isomorphism f between the substructures generated by $\overline{a}$ and $\overline{b}$ such that if $f(x) = y$, then either $rk(x) = rk(y)$ or else $rk(x), rk(y) \geq \omega\alpha$.*

We will later extend this definition to tuples taken from *different* rank homogeneous trees. The next lemma says that the relations $\sim^\alpha$ (as defined above) have the "back-and-forth" property.

Lemma 3.3. *Suppose T is a rank homogeneous tree.*

1. *If $(T,\overline{a}) \sim^0 (T,\overline{b})$, then $\overline{a}$ and $\overline{b}$ satisfy the same quantifier-free formulas.*
2. *If $(T,\overline{a}) \sim^\alpha (T,\overline{b})$, then for all $\beta < \alpha$ and all $\overline{c}$, there exists $\overline{d}$ such that $(T,\overline{a},\overline{c}) \sim^\beta (T,\overline{b},\overline{d})$.*

Proof. For 1, suppose $(T,\overline{a}) \sim^0 (T,\overline{b})$. According to the definition, the function taking $\overline{a}$ to $\overline{b}$ extends to an isomorphism between the subtrees generated by the tuples. This implies that $\overline{a}$ and $\overline{b}$ satisfy the same quantifier-free formulas.

For 2, take f witnessing that $(T,\overline{a}) \sim^0 (T,\overline{b})$. We must extend f to a function g on the subtree generated by $\overline{a},\overline{c}$. For $x \in dom(f)$, let τ_x be the finite set of elements below x that are in the subtree generated by $\overline{a},\overline{c}$, and not in that generated by $\overline{a}$. We shall define g to map τ_x to a set τ_y of new elements below y, preserving tree structure, and and preserving ranks to the following extent: for $x' \in \tau_x$ and $y' = g(x')$, if $rk(x') < \omega\alpha$, then $rk(y') = rk(x')$, and if $rk(x') \geq \omega\alpha$, then $rk(y') \geq \omega\beta$.

We will first outline the required properties of τ_y, and will then show that they can be satisfied. We know the desired tree structure for τ_y, and the target ranks less than $\omega\alpha$ which we plan to match exactly. There may be some elements of τ_y whose ranks are not determined in this way. This implies that $rk(x)$ and $rk(y)$ must both be at least $\omega\alpha$. We calculate lower bounds for ranks of the remaining nodes, working our way up. Suppose for a node y', we have assigned to all successors of y' in τ_y either a precise ordinal rank or a lower bound. We assign y' a lower bound. If there is at least one successor assigned a lower bound, then we take the maximum γ and assign y' the bound $\gamma + 1$. In case y' has no successors, or the successors all have precise ranks $< \omega\beta$, then we assign γ the lower bound $\omega\beta$.

So far, we have said what τ_y should look like. We will now show that the elements actually exist in the subtree below y. Having started at the bottom of τ_y in calculating the target ranks or lower bounds, we now start at the top and work our way down. We let $g(x) = y$. Having found an appropriate $y' = g(x')$, with a target rank γ, we consider a successor x'' of x', and look for a new (not already used) successor y'' of y' to serve as $g(x'')$. If y'' is supposed to have rank exactly $\gamma' < \gamma$, it is because x'' has rank γ', and y' has infinitely many successors of this rank, so we can take a new one for $g(x'')$. If we have decided only that y'' should have rank at least γ', where $\gamma' < \gamma$, then we must show that y' has a successor of some appropriate rank. If $\gamma = \delta + 1$, then $\delta \geq \gamma'$, and y' has infinitely many successors of rank δ. On the other hand, if γ is a limit, then it is the supremum of the ranks of the successors of y'. So, there is some $\delta \geq \gamma'$ such that y' has infinitely many successors of rank δ. We let $g(x'')$ be a new one of these.

□

The next lemma connects the relations $\sim^\alpha$, defined in terms of tree rank, to the relations $\equiv^\alpha$ used in our definition of Scott rank.

Lemma 3.4. *Suppose T is a rank homogeneous tree. If $(T,\overline{a}) \sim^\alpha (T,\overline{b})$, then $(T,\overline{a}) \equiv^\alpha (T,\overline{b})$.*

Proof. We proceed by induction on α. For $\alpha = 0$, this follows from the definition, or from Part 1 of Lemma 3.3. Let $\alpha > 0$, and assume that the statement holds for $\beta < \alpha$. Let $(T,\overline{a}) \sim^\alpha (T,\overline{b})$. By Part 2 of Lemma 3.3, for each $\overline{c}$, there exists $\overline{d}$ and for each $\overline{d}$, there exists $\overline{c}$, such that $(T,\overline{a},\overline{c}) \sim^\beta (T,\overline{b},\overline{d})$, and by the induction hypothesis, this implies $(T,\overline{a},\overline{c}) \equiv^\beta (T,\overline{b},\overline{d})$. Then by definition, $(T,\overline{a}) \sim^\alpha (T,\overline{b})$. □

Here is the main theorem of the section.

Theorem 3.5. *Suppose T is a computable thin rank homogeneous tree with a path but no hyperarithmetical path. Then $SR(T) = \omega_1^{CK}$.*

Proof. By Lemma 3.2, every orbit is definable by a computable infinitary formula, so $SR(T) \leq \omega_1^{CK}$. Let α be a computable ordinal. There is some element $b \in T$ at level n with $rk(b) \geq \omega\alpha$. Let a be an element at level n such that $rk(a) = \infty$. By definition, $(T,a) \sim^\alpha (T,b)$. Then by Lemma 3.4, $(T,a) \equiv^\alpha (T,b)$. Therefore, $SR(a) > \alpha$. It follows that $SR(T) \geq \omega_1^{CK}$. □

In fact, we can give the following more complete description of the Scott ranks of computable rank homogeneous trees.

Theorem 3.6. *Let T be a computable rank homogeneous tree.*

(1) *If there is a computable bound on the ordinal tree ranks (T may or may not have paths), then $SR(T)$ is computable.*
(2) *If for each n, there is a computable bound on the ordinal tree ranks in T_n, but no computable bound over-all, then $SR(T) = \omega_1^{CK}$.*
(3) *If for some n, there is no computable bound on the ordinal tree ranks in T_n, then $SR(T) = \omega_1^{CK} + 1$.*

Proof. For 1, if there is a computable bound on the ordinal tree ranks, then we can define the orbits of all tuples using computable infinitary formulas of bounded complexity. For 2, if for each n, there is a computable bound on the ordinal tree ranks in T_n, then we have computable infinitary formulas defining the orbits. Therefore, $SR(T) \leq \omega_1^{CK}$. If there is no bound on the tree ranks over-all, then $rk(\emptyset) = \infty$, and T has a path. For each computable ordinal $\omega\alpha$, there is an element x of rank $\geq \alpha$. Say $x \in T_n$, and let $y \in T_n$, where $rk(y) = \infty$. Then $(T,x) \sim^\alpha (T,y)$, so $SR(x) > \alpha$. Therefore, $SR(T) \geq \omega_1^{CK}$. For 3, if there is no computable bound on the ordinal tree ranks in T_n, then there is some $x \in T_n$ such that $rk(x) = \infty$. For each computable ordinal α, there exists $y \in T_n$ such that $rk(y) \geq \omega\alpha$. Then $(T,x) \sim^\alpha (T,y)$, so $SR(x) > \alpha$. Therefore, $SR(x) = \omega_1^{CK}$, and $SR(T) = \omega_1^{CK} + 1$. □

We turn to pairs of rank homogeneous trees. Below, we extend the relations $\sim^\alpha$ to pairs of tuples (of the same length) from different trees.

Definition 8. *Let T and T' be rank homogeneous trees.*

(1) *$T \sim^\alpha T'$ if for all n, $R_n(T)$ and $R_n(T')$ contain the same ordinals less than $\omega\alpha$, and $R_n(T)$ has elements greater than or equal to $\omega\alpha$ iff $R_n(T')$ does.*
(2) *for $\overline{a}$ in T and $\overline{b}$ in T', $(T,\overline{a}) \sim^\alpha (T',\overline{b})$ if $T \sim^\alpha T'$ and the function taking $\overline{a}$ to $\overline{b}$ extends to an isomorphism f between the subtrees generated by the tuples, such that f preserves tree rank up to $\omega\alpha$; i.e., for any $x \in T$, we have $rk(x) = rk(f(x))$, or else $rk(x)$ and $rk(f(x))$ are both at least $\omega\alpha$.*

The next lemma generalizes Lemma 3.3, saying that the relations $\sim^\alpha$, for tuples from possibly different trees, have the "back-and-forth" property.

Lemma 3.7. *Suppose T and T' are rank homogeneous trees.*
(1) *If $(T,\overline{a}) \sim^0 (T',\overline{b})$, then $\overline{a}$ and $\overline{b}$ satisfy the same quantifier-free formulas in their respective trees.*
(2) *If $(T,\overline{a}) \sim^\alpha (T',\overline{b})$, then for all $\beta < \alpha$ and all $\overline{c}$, there exists $\overline{d}$ such that $(T,\overline{a},\overline{c}) \sim^\beta (T,\overline{b},\overline{d})$.*

Proof. The proof is exactly as for Lemma 3.3. The condition that $R_n(T)$ and $R_n(T')$ have the same ordinals $< \omega\alpha$ guarantees the existence of the necessary elements.

□

The relations $\sim^\alpha$ are defined for tuples $\overline{a}$, $\overline{b}$ of the same length. Below, we extend the definition to pairs of tuples in which the second may have greater length.

Definition 9. *Suppose T and T' are rank homogeneous trees, and let $\overline{a}$ and $\overline{b}$ be tuples in T, T', respectively, where $length(\overline{a}) \leq length(\overline{b})$. If $\overline{c}$ is the restriction of $\overline{b}$ to the elements corresponding to $\overline{a}$, then $(T,\overline{a}) \preceq^\alpha (T',\overline{b})$ if $(T,\overline{a}) \sim^\alpha (T',\overline{c})$.*

We have referred to the "back-and-forth" property for a family of relations on a pair of structures. Now, we give the definition (not just for trees).

Definition 10. *Let $\mathcal{A}$ and $\mathcal{B}$ be structures for the same language. For simplicity, suppose the language is finite, and that finitely generated substructures are finite. Let $(\leq_\beta)_{\beta<\alpha}$ be a family of binary relations on tuples. We say that the relations have the* back-and-forth *property if*
(1) *$(\mathcal{A},\overline{a}) \leq_0 (\mathcal{A}',\overline{b})$ implies that all quantifier-free formulas true of $\overline{a}$ in $\mathcal{A}$ are true of $\overline{b}$ in $\mathcal{A}'$,*
(2) *for $\beta > 0$, if $(\mathcal{A},\overline{a}) \leq_\beta (\mathcal{A}',\overline{b})$, then for all $\gamma < \beta$ and all $\overline{d}$, there exists $\overline{c}$ such that $(\mathcal{A}',\overline{b},\overline{d}) \leq_\gamma (\mathcal{A},\overline{a},\overline{c})$.*

The following is clear from the definitions together with Lemma 3.7.

Proposition 3.8. *Let T and T' be rank homogeneous trees such that $T \sim^\alpha T'$. Then the relations $\preceq^\beta$ defined above, for $\beta < \alpha$, have the back-and-forth property.*

There are other families of relations with the back-and-forth property. In particular, there are the "standard" back and forth relations (see [4]). What makes the particular relations that we have defined above useful is the fact that they are uniformly c.e. Let P be a path through $\mathcal{O}$. Let T and T' be computable rank homogeneous trees, each with a computable rank function which assigns to each node of rank $\omega\beta + n$ the label (b,n), where b is the notation for β in P.

For a fixed α, with notation $a \in P$, given a node with label (b,n), we can apply an effective procedure to see if the node has tree rank at least $\omega\alpha$—we check whether $a = b$, and if not, we watch simultaneously for a to appear among the $<_{\mathcal{O}}$-predecessors of b, and for b to appear among the $<_{\mathcal{O}}$-predecessors of a. It is easy to tell whether two nodes, both of tree rank less than $\omega\alpha$, have the same rank—the labels must match.

Proposition 3.9. *Let T and T' be computable rank homogeneous trees, with computable rank functions. If $T \sim^\alpha T'$, then the relations $(T,\overline{a}) \preceq^\beta (T',\overline{b})$, for $\beta < \alpha$, are c.e. Moreover, we can effectively find a c.e. index for $\preceq^\beta$, given the notation $b \in P$ for β and computable indices for T and T', with their computable rank functions.*

4. A special tree and family of approximations

In this section, we will produce a tree $T^{h,*}$ of Scott rank ω_1^{CK}, along with a family of trees that will be used in Section 5 to show that $T^{h,*}$ is strongly computably approximable. We begin with a Harrison ordering $\mathcal{H}$. We may suppose that there is no hyperarithmetical decreasing sequence in $\mathcal{H}$ (see [8], [15], or [4]). Kleene showed that the class of computable indices for well orderings is m-reducible to $\mathcal{O}$. We use ideas from Kleene's proof, as described in [14], [4]. First, we suppose that in $\mathcal{H}$, we can effectively determine which elements are successors and which are limits, and there is an effective procedure which, when applied to a successor element a, yields the predecessor, and when applied to a limit element a, yields a computable index for an increasing sequence $(a_n)_{n\in\omega}$ with limit a. (If the original $\mathcal{H}$ did not have this feature, we would replace it by $\mathcal{H} \times \omega$, with the lexicographic ordering.) We write $pred(a)$ for the set $\{b : \mathcal{H} \models b < a\}$.

Still following Kleene, we will get a partial computable function f that maps the well ordered initial segment of $\mathcal{H}$ onto a path P in $\mathcal{O}$, such that for a computable ordinal α, the set $pred(a)$ has order type α if and only if $f(a)$ is a notation for α. If a is the first element of $\mathcal{H}$, we let $f(a) = 1$. If a is the successor of b in $\mathcal{H}$, then $f(a) = 2^{f(b)}$. If a is a limit element, and $(a_n)_{n\in\omega}$ is our effectively determined increasing sequence with limit a, then we can effectively find an index e for the corresponding sequence $(f(a_n))_{n\in\omega}$. We let $f(a) = 3 \cdot 5^e$.

The function f will be defined on more than the well ordered initial segment of $\mathcal{H}$. However, if $pred(a)$ is not well ordered, then $f(a)$ will not be an element of $\mathcal{O}$. Given an element a of the well ordered initial segment of $\mathcal{H}$, we can find the corresponding element $f(a)$ in P. Conversely, given an element of P, we can effectively find the corresponding element of $\mathcal{H}$. We may therefore identify a with $f(a)$ and think of P as an initial segment of $\mathcal{H}$.

Next, starting with $\mathcal{H}$, we form $\mathcal{H}^* = \mathcal{H} \times \omega$, with the lexicographic ordering. Again, the order type is $\omega_1^{CK}(1 + \eta)$, we can effectively determine successor and limit elements, and we can effectively find witnesses as above. If $pred(b)$ has order type β in $\mathcal{H}$, and $n \in \omega$, then in $\mathcal{H}^*$, the set of predecessors of (b, n) has order type $\omega\beta + n$. Now, (b, n) is exactly the kind of value we want for our computable rank function, to represent rank $\omega\beta + n$. In what follows, we shall describe some computable trees, with computable rank functions taking values in $\mathcal{H}^*$.

Lemma 4.1. *Given $a \in \mathcal{H}$, we can pass effectively to a computable tree T^a, with a computable function labeling the nodes by elements of $\mathcal{H}^*$, such that $\emptyset$ has label $(a, 0)$, and if x has label (b, k), then*

(1) *if $pred(b)$ has type β, then $rk(x) = \omega\beta + k$, and*

(2) *if pred*(b) *is not well ordered, then* $rk(x) = \infty$.

Proof. We give $\emptyset$ label $(a, 0)$, as required. If $x \in T^a$ has label $(b, k+1)$, then we give x just one successor, with label (b, k). If $x \in T^a$ has label $(b, 0)$, where $b = c + 1$, then for each n, we give x one successor with label (c, n). Finally, if $x \in T^a$ has label $(b, 0)$, where b is a limit element of $\mathcal{H}$, and $(b_n)_{n\in\omega}$ is the effectively determined increasing sequence with limit b, then for each n, x has one successor with label $(b_n, 0)$. This completes the description of T^a and its labels.

$\square$

In any tree, for any node x with ordinal tree rank γ, all ordinals $\gamma' < \gamma$ occur as tree ranks of nodes y below x. (The proof of this is an easy induction on γ.) Therefore, we have the following.

Lemma 4.2. *If pred*(a) *is well ordered, then for any* $b < a$ *and* $n \in \omega$, (b, n) *occurs as a label in* T^a.

Now, we take $h \in \mathcal{H}$ such that *pred*(h) is not well-ordered, but the property above still holds.

Lemma 4.3. *There is some* $h \in \mathcal{H}$ *such that pred*(h) *is not well-ordered, and for all* $x \in T^h$, *if* x *has label* $r = (b, n)$, *then all* $r' < r$ *occur as labels on nodes* y *below* x.

Proof. Let I consist of those $a \in \mathcal{H}$ such that for all $x \in T^a$, if x has label $r = (b, n)$, then all predecessors of r (in the lexicographic ordering on $\mathcal{H}^*$) occur as labels on nodes of T^a below x. Now, I is a hyperarithmetical subset of $\mathcal{H}$ that includes all of the well ordered initial segment. Since there is no hyperarithmetical well ordering of type ω_1^{CK}, I must contain some h such that *pred*(h) is not well ordered.

$\square$

We fix h as in Lemma 4.3, and consider T^h. In T^h, the top node $\emptyset$ has label $(h, 0)$, indicating that the rank is not a computable ordinal. In fact, $rk(\emptyset) = \infty$, so T^h has a path. There is no hyperarithmetical path, since $\mathcal{H}$ has no hyperarithmetical decreasing sequence.

We proceed as in Section 2, transforming T^h first to a thin tree $T^{h,1}$, with a computable rank function, and then to a rank homogeneous tree T^*, where $R_n(T^*) = R_n(T^{h,1})$ for all n.

Lemma 4.4. *There is a computable thin tree* $T^{h,1}$, *with a computable rank function such that* $\emptyset$ *has label* $(h, 0)$, *and* $T^{h,1}$ *shares with* T^h *the feature that if* x *has label* r, *then each* $r' < r$ *occurs as the label on some node below* x.

Proof. The proof is the same as for Lemma 2.3, which was given in [9]. We keep the top node and level 1 as in T^h. At each level $n > 1$, we "expand" at most one node, corresponding to an element x of $T^{h,1}$ that has a limit label, always choosing the first x. The result is a computable embedding f of T^h in $T^{h,1}$ that preserves labels. Suppose x is a successor of z. If x has successor label $(b, n+1)$, then $f(x)$ is a successor of $f(z)$, with the same label. If x has a limit label $(b, 0)$, then $f(x)$ may not be a successor of $f(z)$ (because the expansion is delayed). We can determine the level by looking at the number of $x' < x$ waiting to be expanded. It follows that we can effectively determine the appropriate labels on the chain leading from $f(z)$

down to $f(x)$. Note that if x has label (b, n), where $pred(b)$ is not well ordered, then x has at least one successor with label (b', n'), where $pred(b')$ is not well ordered.

□

Lemma 4.5. *Let $T^{h,1}$ be as in Lemma 4.4. There is a computable rank homogeneous tree T^*, with a computable rank function, such that $R_n(T^*) = R_n(T^{h,1})$ for all n. (Such a T^* is necessarily thin.)*

Proof. The proof is the same as for Lemma 2.4. We give the top node $\emptyset$ the same label as in $T^{h,1}$. For all x at level n in T^*, if x has been given label (b, m), then for all $(b', m') < (b, m)$ such that some element at level $n + 1$ in $T^{h,1}$ has label (b', n'), we give x infinitely many successors with label (b', m'). According to this scheme, if x has label (b, m), where $pred(b)$ is not well ordered, then for at least one $(b', m') < (b, m)$, where $pred(b')$ is not well ordered, x has infinitely many successors with label (b', m').

□

We are ready to define a new family of trees $T^{a,*}$, for $a < h$, such that if a is in the well-ordered initial segment P, where $pred(a)$ has type α, then $T^{a,*}$ will have computable rank at least $\omega\alpha$, and $T^{a,*}$ will be a good approximation for T^*.

Lemma 4.6. *Given $a \in P$ such that $pred(a)$ has order α, we can apply a uniform effective procedure to obtain a computable tree $T^{a,1}$, of computable rank $\omega(\alpha + 1)$, with a computable rank function, such that for each n, the labels $r < (a, 0)$ on nodes at level n of T^* (or $T^{h,1}$) match those on nodes at level n in $T^{a,1}$. Furthermore, if T^* has nodes of rank at least $\omega\alpha$ at level n, then $T^{a,1}$ will have them, as well.*

Proof. We first remove from $T^{h,1}$ all nodes with labels $r > (a, 0)$. Now, there are "orphans" at various levels—that is, nodes x with no predecessor. Suppose x is such a node at level $n > 0$. If x has label $(a, 0)$, then we add a chain of length n connecting x to the top node, and we put the obvious labels, all greater than $(a, 0)$, on the nodes in the chain.

Suppose that x is an orphan at level $n > 0$ with label strictly less than $(a, 0)$. Since the predecessor z of x in $T^{h,1}$ had a label greater than or equal to $(a, 0)$, there exists y below z, at some level $M \geq n$, such that y has label $(a, 0)$. We may suppose that y has been given a chain of ancestors leading to the top node, as in the previous paragraph. Say y' is the one at level $n - 1$. We attach x as a successor of this y'. The result of all this is a computable tree $T^{a,1}$ of rank $\omega(\alpha + 1)$, with a computable rank function, such that for all n, and all $\gamma < \omega\alpha$, $R_n(T^{a,1})$ and $R_n(T^*)$ include the same ordinals $\gamma < \omega\alpha$.

□

The tree $T^{1,a}$ may not be thin. Next, we replace $T^{a,1}$ by a tree that is rank homogeneous.

Lemma 4.7. *Given $a \in P$, we can effectively find a computable tree $T^{a,*}$, with computable rank function, such that $T^{a,*}$ is rank homogeneous, and for all n, $R_n(T^{a,*}) = R_n(T^{a,1})$.*

Proof. The proof is the same as for Lemmas 2.4 and 4.5.

□

We have designed the approximations $T^{a,*}$ such that if $pred(a)$ has type α, then $T^* \sim^\alpha T^{a,*}$. We shall also need the fact that $(T^{b,*})_{b<a}$, with the back-and-forth relations from Section 3, is "α-friendly". We give the definition below. (For more information, see [4].)

Definition 11. *Let $(\mathcal{A}_i)_{i\in I}$ be a family of structures, and let $(\leq_\gamma)_{\gamma<\alpha}$ be a family of binary relations on the pairs $(i,\overline{a})$, where i is in I and $\overline{a}$ is a tuple from $\mathcal{A}_i$. (We are letting $(i,\overline{a})$ represent $(\mathcal{A}_i,\overline{a})$.) We identify the ordinals $< \alpha$ with their unique notations on the fixed path P through $\mathcal{O}$. We say that the family $(\mathcal{A}_i)_{i\in I}$ is α-friendly, under the relations $\leq_\gamma$, for $\gamma < \alpha$, if*

1. *the relations $\leq_\gamma$ are reflexive and transitive, and for any $i,j \in I$, the restrictions to pairs $(i,\overline{a}),(j,\overline{b})$ for a given i and j have the back-and-forth property,*
2. *the structures $\mathcal{A}_i$ are computable, uniformly in i, and*
3. *the relations $\leq_\gamma$ are c.e., uniformly in the notation for γ.*

Recall the relations $\preceq^\gamma$ from Section 3. We defined these relations for tuples from a pair of trees. By Proposition 3.9, if the trees T and T' are computable, with computable rank functions, then the relations are c.e., uniformly in indices for T and T' and their rank functions. Now, for a given $a \in P$ such that a is a notation for α, we consider tuples from the family of trees $T^{b,*}$, for $b < a$. For each $\gamma < \alpha$, we form a single relation $\preceq^\gamma$ such that $(b,\overline{a}) \preceq^\gamma (b',\overline{a}')$ if $(T_b,\overline{a}) \preceq^\gamma (T_{b'},\overline{a}')$. We have the following.

Lemma 4.8. *Let a be the notation for α in P. Then the family $(T^{b,*})_{b<a}$ is α-friendly under the relations $\preceq^\gamma$, for $\gamma < \alpha$.*

5. Strong computable approximability

Let $T^{h,*}$ be the computable tree of Scott rank ω_1^{CK} constructed in the previous section. Recall that P is a path through $\mathcal{O}$. Whenever it is helpful, we identify the ordinals with their notations in P. We write $|a|$ for the ordinal with notation a. Let $(T^{a,*})_{a\in P}$ be the family of approximating trees constructed in the previous section. Recall that these all have computable Scott rank. Our goal in this section is to show that $T^{h,*}$ is strongly computably approximable, using the trees $T^{a,*}$, for $a \in P$. We shall apply the following technical result on approximations.

Theorem 5.1. *Let $\mathcal{A}$ be a computable structure. Let $(\mathcal{A}_a)_{a\in P}$ be a family of further computable structures for the same language, indexed by elements of a fixed path P through $\mathcal{O}$. Let $\leq_\beta$, for computable ordinals β, be a family of binary relations on tuples from the structures $\mathcal{A}_a$. Suppose that*

- *the computable $\Sigma_{|a|}$ sentences true in $\mathcal{A}$ are all true in $\mathcal{A}_a$,*
- *$(\mathcal{A}_b)_{b<a}$ is $|a|$-friendly under the back-and-forth relations $\leq_\beta$, for $\beta < |a|$.*

Then for any Σ^1_1 set S, there is a uniformly computable sequence $(\mathcal{C}_n)_{n\in\omega}$ such that if $n \in S$, then $\mathcal{C}_n \cong \mathcal{A}$, and if $n \notin S$, then $\mathcal{C}_n \cong \mathcal{A}_b$, for some $b \in P$.

Before proving Theorem 5.1, let us see how it applies to the tree T^* and the approximations $T^{a,*}$, for $a \in P$. Suppose $|a| = \alpha$. We have the following:

1. $T^* \sim^\alpha T^{a,*}$, and
2. $(T^{b,*})_{b<a}$ is an α-friendly family under the back-and-forth relations $\preceq^\gamma$, for $\gamma < \alpha$.

It follows from 1 that the computable Σ_α sentences true in T^* are true in $T^{a,*}$. Then 2 puts us in a position to apply Theorem 5.1. The conclusion says that T^* is strongly computably approximable, using the trees $T^{a,*}$, for $a \in P$.

5.1. **Proof of Theorem 5.1.** In this section, we describe the proof of Theorem 5.1. We use Δ^0_β to denote the particular complete Δ^0_β oracle associated with the notation $b \in P$ for β. We may suppose that there is a uniform effective procedure which, when applied with the oracle Δ^0_β, yields the notation b.

Let S be a Σ^1_1 set. We write $n \notin S_\alpha$ if Δ^0_α can determine that $n \notin S$, and we write $n \in S_\alpha$ otherwise. We suppose that if α is first such that $n \notin S_\alpha$, then for all $\gamma > \alpha$, the oracle for Δ^0_γ can determine that $n \notin S$.

To prove Theorem 5.1, we use Barwise-Kreisel Compactness. There is a Π^1_1 set Γ of computable infinitary sentences, describing a uniformly computable sequence of structures $(\mathcal{C}_n)_{n\in\omega}$ such that if $n \in S_\alpha$, then $\mathcal{A}_n$ satisfies all of the computable Σ_α sentences true of $\mathcal{A}$, and if α is first such that $n \notin S_\alpha$, then $\mathcal{C}_n$ satisfies all of the computable infinitary sentences true of $\mathcal{A}_a$, where $|a| = \alpha$.

We must show that every Δ^1_1 subset of Γ is satisfiable. For this, we need the following.

Lemma 5.2. *For each computable ordinal $\alpha \geq 1$, there is a uniformly computable sequence $(\mathcal{C}_n)_{n\in\omega}$ such that if $\beta \leq \alpha$ is first such that $n \notin S_\beta$, then $\mathcal{C}_n \cong \mathcal{A}_\beta$, and otherwise $\mathcal{C}_n \cong \mathcal{A}_\alpha$.*

Ash developed some quite general machinery for transfinitely nested priority constructions. He defined the notion of an "α-system", with a set of four conditions taking care of the combinatorics. To use Ash's machinery, it is sufficient to define an appropriate α-system (satisfying the four conditions), together with a Δ^0_α "instruction function", giving the high-level information needed. The success of the construction is guaranteed by a "Metatheorem". For a discussion of α-systems, with a number of examples, see [4].

If we could, we would produce $(\mathcal{C}_n)_{n\in\omega}$ by applying Ash's Metatheorem, in a uniform way, using a single α-system $\mathcal{S} = (L, U, \hat{\ell}, Q, E, (\leq_\gamma)_{\gamma<\alpha})$, and a family of instruction functions $(p_n)_{n\in\omega}$ which are uniformly Δ^0_α. For each n, the instruction function p_n has just one piece of information to give; namely the least β, if any, such that $n \notin S_\beta$. We can define most of the ingredients of an α-system in a natural way. However, there is no good candidate for the "anchor label" $\hat{\ell}$, and we cannot verify Ash's fourth condition.

Although we do not have an α-system, we can leave $\hat{\ell}$ indefinite and get what we want anyway. As in earlier papers such as [2], [3], having carried out a nested priority construction to which we cannot apply Ash's original Metatheorem, we record the abstract properties that make the construction work, so that when we see these same properties again, we can use what we have just done.

5.2. **Indefinite α-systems.** In what follows, whenever it is convenient, we identify computable ordinals with their notations on a fixed path through $\mathcal{O}$. We define the following variant of an α-system.

Definition 12. *An* indefinite *α-system is a structure of the form*

$$\mathcal{S} = (L, Q, E, (\leq_\gamma)_{\gamma<\alpha}) \ ,$$

where L is a c.e. set, and Q is a c.e. tree of finite sequences $\beta\ell_\beta\ell_1\ell_2\ldots$, starting with an ordinal $\beta \leq \alpha$, and with later terms in L (we may suppose that there is a unique ℓ_β which can follow a given β), E is a c.e. enumeration relation assigning to each $\ell \in L$ a finite set $E(\ell)$, and the $\leq_\gamma$ are binary relations on L, uniformly c.e. We suppose that if $\beta > \gamma$, and $\beta\ell, \gamma\ell'$ are both in Q, then $\ell \leq_\gamma \ell'$. In addition, we have the following conditions:

1. *If $\ell \leq_0 \ell'$, then $E(\ell) \subseteq E(\ell')$.*
2. *The relations $\leq_\gamma$ are transitive and reflexive.*
3. *If $\beta < \gamma$, then $\ell \leq_\gamma \ell'$ implies $\ell \leq_\beta \ell'$.*
4. *Every "picture" has a "completion", where these notions are defined below.*

A picture *is a pair $(\sigma;\tau)$, where σ is a sequence in Q of length at least 2, and τ is either empty or of the form $\gamma_0\ell^1\gamma_1\ell^2\ldots\gamma_{k-1}\ell^k$. We suppose that $\gamma_0 > \gamma_1 \ldots > \gamma_{k-1}$, and $\ell^1 \leq_{\gamma_1} \ldots \leq_{\gamma_{k-1}} \ell^k$. If σ starts with β, then $\beta > \gamma_0$, and if σ ends with $\ell^0 \in L$, then $\ell^0 \leq_{\gamma_0} \ell^1$. A* completion *of the picture $(\sigma;\tau)$ is some ℓ such that $\sigma\ell \in Q$, $\ell^i \leq_{\gamma_i} \ell$, for all $i < k$, and $\ell^k \leq_0 \ell$.*

Next, we define a variant of an instruction function. For an indefinite α-system, there is just one piece of information, an ordinal $\beta \leq \alpha$. The idea behind the definition below is that for each γ, $1 \leq \gamma \leq \alpha$, Δ^0_γ knows that the value is β if $\beta \leq \gamma$, and guesses that the value is α, otherwise.

Definition 13. *Let Q be the tree in an indefinite α-system.*

1. *An* indefinite Δ^0_α *instruction function, with* value *$\beta \leq \alpha$, is a function q on ordinals $1 \leq \gamma \leq \alpha$, where $q(\gamma)$ is Δ^0_γ, uniformly in γ, and*

$$q(\gamma) = \left\{ \begin{array}{ll} \beta & \text{if } \gamma \geq \beta \\ \alpha & \text{otherwise} \end{array} \right\}$$

2. *A* run *of (Q, q) is a path through Q starting with the true value $\beta = q(\alpha)$.*

Here is our variant of Ash's Metatheorem.

Theorem 5.3 (Metatheorem)**.** *Let $\mathcal{S} = (L, Q, E, (\leq_\gamma)_{\gamma<\alpha})$ be an indefinite α-system, and let q be an indefinite Δ^0_α instruction function for Q, with value β. Then there is a Δ^0_β run $\pi = \beta\ell_0u_1\ell_1u_2\ell_2\ldots$ of (Q, q) such that $E(\pi) = \cup_k E(\ell_k)$ is c.e. Moreover, from indices for q and $\mathcal{S}$, we can find a c.e. index for $E(\pi)$.*

The proof is almost exactly the same as for Ash's original Metatheorem and the variants in [4], [1], [2]. We assume that the reader can look at the proof, say in [4], and make the minor changes that are needed. If β is the value of the indefinite instruction function, then everything proceeds as though we were dealing with a β-system, with a trivial Δ^0_β instruction function. At levels above β, the steps are the same as at level β. At levels below β, we have the usual steps of approximation. At levels $\gamma \geq \beta$ and above, the first label is ℓ_β, and all of the pictures are trivial, reflecting the fact that there is no approximation. At levels $\gamma < \beta$, the first label is ℓ_α, and all of the ordinals that occur in the pictures are $\geq \gamma$.

5.3. **Return to Lemma 5.2.** We apply the metatheorem above to prove Lemma 5.2. Let C be an infinite computable set of constants, for the universe of $\mathcal{C}_n$. Let L be the set of pairs (β, f), where $\beta \leq \alpha$, and f is a finite partial $1-1$ function from C to $\mathcal{A}_\gamma$. Let Q be the set of finite sequences of the form $\beta\ell_0\ell_1\ell_2\ldots$, where $\beta \leq \alpha$, ℓ_i has the form (β, f_i) (for the same β), the first i elements of C are in $dom(f_i)$ and the first i elements of $\mathcal{B}_\beta$ are in $ran(f_i)$. Let E be the usual enumeration relation on L; that is, for $\ell = (\beta, f)$, where n is the cardinality of f, we let $E(\ell)$ consist of the atomic sentences and negations of atomic sentences with constants in $dom(f)$ with Gödel number is at most n and such that f makes the sentence true in $\mathcal{B}_\beta$. Let $\leq_\gamma$ be the given back-and-forth relations.

We have described the ingredients of our indefinite α-system. The first three conditions are clearly satisfied. To verify Condition 4, consider a picture $(\sigma;\tau)$, where σ begins with β and ends in $\ell^0 = (\beta, f)$. If $\tau = \emptyset$, then there is clearly a next ℓ such that $\sigma\ell \in Q$—we just extend the function f from ℓ^0. Suppose $\tau = \gamma_0\ell^1\gamma_1\ell^2\ldots\gamma_{k-1}\ell^k$. Then $\ell_0 \leq_{\gamma_0} \ell^1$. To complete the picture, we first form extensions $(\ell^i)'$ of ℓ^i such that $(\ell^i)' \leq_{\gamma_i} \ell^{i-1}$, starting with $i = k$ and working back to $i = 0$. Then we extend the function in $(\ell^0)'$ to get the desired completion ℓ.

References

[1] Ash, C. J., and J. F. Knight, "Pairs of recursive structures", *Annals of Pure and Appl. Logic*, vol. 46(1990), pp. 211–234.

[2] Ash, C. J., and J. F. Knight, "Ramified systems", *Annals of Pure and Appl. Logic*, vol. 70(1994), pp. 205–221.

[3] Ash, C. J., and J. F. Knight, "Mixed systems", *J. Symb. Logic*, vol. 59(1994), pp. 1383–1399.

[4] Ash, C. J., and J. F. Knight, *Computable Structures and the Hyperarithmetical Hierarchy*, Elsevier, 2000.

[5] Barwise, J., "Infinitary logic and admissible sets", *J. Symb. Logic*, vol. 34(1969), pp. 226–252.

[6] Goncharov, S. S., V. S. Harizanov, J. F. Knight, and R. Shore, "Π^1_1 relations and paths through O", *J. Symb. Logic*, vol. 69(2004), pp. 585–611.

[7] Goncharov, S. S., and J. F. Knight, "Computable structure and non-structure theorems", *Algebra and Logic*, vol. 41(2002) (English), pp. 351–373.

[8] Harrison, J., "Recursive pseudo well-orderings",*Trans. of the Amer. Math. Soc.*, vol. 131(1968), pp. 526–543

[9] Knight, J. F., and J. M. Young, "Computable structures of rank ω_1^{CK}", submitted to *J. of Math. Logic*.

[10] Makkai, M., "An example concerning Scott heights", *J. Symb. Logic*, vol. 46(1981), pp. 301–318.

[11] Morozov, A. S., "Groups of computable automorphisms", in *Handbook of Recursive Mathematics*, ed. by Yu. L. Ershov, et. al., 1998, Elsevier, pp. 311–345.

[12] Nadel, M., "Scott sentences and admissible sets", *Annals of Mathematical Logic*, vol. 7 (1974), pp. 267–294.

[13] Nadel, M., "$L_{\omega_1\omega}$ and admissible fragments", in *Model-Theoretic Logics*, vol. 45(1980), pp. 612–622.

[14] Rogers, H., Jr., *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, New York, 1967.

[15] Sacks, G. E., *Higher Recursion Theory*, Springer-Verlag, 1990.

[16] Scott, D., "Logic with denumerably long formulas and finite strings of quantifiers", in *The Theory of Models*, ed. by J. Addison, L. Henkin, and A. Tarski, North-Holland, 1965, pp. 329–341.