Uniformity in Computable Structure Theory*

Rod G. Downey[1]
Denis R. Hirschfeldt[1]
Bakhadyr Khoussainov[2]

[1] School of Mathematical and Computing Sciences, Victoria University of Wellington

[2] Department of Computer Science, University of Auckland

Abstract

We investigate the effects of adding uniformity requirements to concepts in computable structure theory such as computable categoricity (of a structure) and intrinsic computability (of a relation on a computable structure). We consider and compare two different notions of uniformity, previously studied by Kudinov and by Ventsov. We discuss some of their results and establish new ones, while also exploring the connections with the relative computable structure theory of Ash, Knight, Manasse, and Slaman and Chisholm and with previous work of Ash, Knight, and Slaman on uniformity in a general computable structure-theoretical setting.

1 Introduction

Analyzing the effective content of theorems of classical model theory is part of the work of computable model theory. Another part, which can be termed computable structure theory, consists of studying properties of structures that are particular to the computable setting. One of the important trends in this study has been the discovery of an increasing number of computable structures with "pathological" computable properties that make

*The authors' research was partially supported by the Marsden Fund of New Zealand. Hirschfeldt's current affiliation is: Department of Mathematics, University of Chicago. The authors thank the anonymous referee for providing valuable references to the literature.

them quite different from the more natural examples that initially motivated definitions in the field. As an example, let us consider the number of different computable versions of a given structure.

It has long been known that isomorphic computable structures can behave quite differently from a computability-theoretic point of view. For example, if $\mathcal{L}$ is a linear ordering of type ω and S is the successor relation on $\mathcal{L}$ then there is a computable copy of $\mathcal{L}$ in which the image of S is computable, namely ω with its standard ordering. But there are also computable copies of $\mathcal{L}$ in which the images of S are not computable (see [8] for details). This leads to the following definitions. (We will always assume we are working with computable languages.)

1.1 Definition. A structure $\mathcal{A}$ is *computable* if both its domain $|\mathcal{A}|$ and the atomic diagram of $(\mathcal{A}, a)_{a \in |\mathcal{A}|}$ are computable.

An isomorphism from a structure $\mathcal{A}$ to a computable structure is called a *computable presentation* of $\mathcal{A}$. We often abuse terminology and refer to the image of a computable presentation as a computable presentation.

The *computable dimension* of a computable structure $\mathcal{A}$ is the number of computable presentations of $\mathcal{A}$ up to computable isomorphism.

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

It is easy to give examples of computably categorical structures (for instance, ω with successor) and of structures of infinite computable dimension (for instance, ω as a linear ordering). On the other hand, structures of finite computable dimension greater than 1 are quite hard to come by (they were first shown to exist by Goncharov [10]), and all such structures built so far have required rather elaborate constructions.

Even computably categorical structures are not immune from having surprising properties involving computable dimension. As shown by Cholak, Goncharov, Khoussainov, and Shore [7], there exist computably categorical structures that, when expanded by a single constant, are no longer computably categorical. Such structures can be thought of as having structures of computable dimension greater than 1 hiding in them.

One approach to dealing with pathologies like the ones described above is to find conditions under which they cannot occur. For example, suppose that the existential diagram of the computable structure $\mathcal{A}$ is computable. Then, as shown by Goncharov [9], the computable dimension of $\mathcal{A}$ is either 1 or ω. Furthermore, if $\mathcal{A}$ is computably categorical then, as shown by Millar [19], so is every expansion of $\mathcal{A}$ by finitely many constants.

Another approach is to try to pinpoint the source of the pathologies and eliminate them by altering the relevant definitions. One example of this approach is relative computable structure theory, originated independently by Ash, Knight, Manasse, and Slaman [2] and Chisholm [6]. The idea there is to consider all presentations of a structure, rather than just computable ones.

1.2 Definition. An isomorphism from a structure $\mathcal{M}$ to a structure $\mathcal{A}$ with $|\mathcal{A}| \subseteq \omega$ is called a *presentation* of $\mathcal{M}$. As in the computable case, we often refer to $\mathcal{A}$ as a presentation of $\mathcal{M}$. The *degree* $\deg(\mathcal{A})$ of this presentation is the join of the (Turing) degrees of $|\mathcal{A}|$ and the atomic diagram of $(\mathcal{A}, a)_{a \in |\mathcal{A}|}$.

A structure is *relatively computably categorical* if any two of its presentations are isomorphic via a map computable in the join of the degrees of the presentations.

Relatively computably categorical structures are quite well-behaved. In particular, any expansion of such a structure by finitely many constants remains relatively computably categorical. This is a consequence of the fact that relative computable categoricity corresponds to a very natural syntactic notion.

1.3 Definition. Let $\mathcal{A}$ be a structure. A *Scott family* for $\mathcal{A}$ is a computably enumerable (c.e.) set S of existential formulas in the language of $\mathcal{A}$ expanded by finitely many constants from $\mathcal{A}$ such that every tuple of elements of $\mathcal{A}$ satisfies at least one formula in S and any two tuples of elements of $\mathcal{A}$ satisfying the same formula in S are automorphic.

1.4 Theorem (Ash, Knight, Manasse, and Slaman; Chisholm). *A computable structure is relatively computably categorical if and only if it has a Scott family.*

More recently, McCoy [18] has shown that, for a natural notion of relative computable dimension introduced by Goncharov and Ventsov, no computable structure has finite relative computable dimension greater than 1.

It should be noted that all known natural examples of computably categorical structures are also relatively computably categorical, where by natural examples we mean ones that do not require elaborate constructions specifically exploiting the fact that the definition of computable categoricity mentions only computable structures. Such examples include linear orderings with finitely many pairs of adjacent elements, Boolean algebras with finitely many atoms, algebraically closed fields of finite transcendence degree, and many others.

Another possible explanation for the surprising examples that have arisen in the study of computable structures is the lack of uniformity requirements in the relevant definitions. For example, if a computable structure $\mathcal{A}$ is computably categorical then we know that, for every computable structure $\mathcal{B} \cong \mathcal{A}$, there exists a computable isomorphism from $\mathcal{B}$ to $\mathcal{A}$, but producing such an isomorphism given $\mathcal{B}$ might be hard, since it might depend on properties of $\mathcal{B}$ that are not uniformly effective over computable structures isomorphic to $\mathcal{A}$.

On the other hand, the natural examples of computably categorical structures mentioned above are all clearly uniformly computably categorical (in any reasonable sense of uniformity), provided we allow expansion by finitely many constants. (For example, ω with successor is not uniformly computably categorical, but becomes so if we add a constant for 0.)

The purpose of this paper is to study the effects of adding uniformity requirements to some definitions in computable structure theory. As we will see, depending on what we mean by uniformity, these effects can be the same as those produced by relativization.

Previous work in this direction has been done by Kudinov [14, 15, 16] and by Ventsov [23, 24], as we discuss below. Ash, Knight, and Slaman [3] worked in a general context and considered both relativization and one of the notions of uniformity discussed below. We will elaborate on the connections with their work in the next section.

For basic notions of computability theory and model theory, the reader is referred to [22] and [11], respectively. In particular, we will often appeal to Kleene's Recursion Theorem (Theorem II.3.1 in [22]), arguing informally and leaving the details to the reader (although we do provide some of these details in the footnote to the proof of Theorem 2.2). Unless otherwise noted, $\Phi_0, \Phi_1, \ldots$ is a standard list of all unary partial computable functions.

As in the case of computable functions, the computable structures in a given language cannot be effectively listed. This leads us to consider *partial computable structures*. It will not matter for our purposes precisely how these are defined, but only that, for any computable language L, there is a computable list of all partial computable structures in L, and this list includes all the computable structures in L. We do assume, however, that the partial computable structures in L are encoded as subsets of ω in an effective manner, so that when we mention a *partial computable operator* Ψ on partial computable structures in L, we have in mind a Turing functional $\widehat{\Psi}$ acting on the corresponding encodings; that is, for a partial computable structure M in L, if S is the encoding of M as a subset of ω then $\Psi(M)$ denotes $\widehat{\Psi}(S)$. A similar interpretation is intended for *c.e. operators* on partial computable structures.

We should also mention that by a *partial structure* in a language L we mean a structure that does not necessarily contain assignments for all the constant symbols in L and such that function symbols in L may be assigned to partial functions. Given a structure $\mathcal{A}$ with $|\mathcal{A}| \subseteq \omega$ and $n \in \omega$, the partial structure $\mathcal{A} \upharpoonright n$ is obtained by restricting the domain of $\mathcal{A}$ to $|\mathcal{A}| \cap \{0, \ldots, n-1\}$ and restricting its constants, functions, and relations in the obvious way. (This of course means that some constants might become unassigned and some functions might become partial.)

2 Uniform Computable Categoricity

The first concept we study is computable categoricity. There are at least two natural ways to define the concept of uniform computable categoricity. In computability theory, uniformity is usually defined in terms of indices. From a model-theoretic viewpoint, however, it seems more natural to require the existence of an effective procedure for producing the desired isomorphisms that takes structures, rather than indices for structures, as input. These two different approaches are reflected in the following definitions.

2.1 Definition. Let $\mathcal{A}$ be a computable structure and let $M_0, M_1, \ldots$ be a computable list of all partial computable structures in the language of $\mathcal{A}$.

$\mathcal{A}$ is *weakly uniformly computably categorical* (weakly u.c.c.) if there is a total computable f such that $M_e \cong \mathcal{A} \Rightarrow \Phi_{f(e)} : M_e \cong \mathcal{A}$.

$\mathcal{A}$ is *uniformly computably categorical* (u.c.c.) if there is a partial computable operator Ψ such that $M_e \cong \mathcal{A} \Rightarrow \Psi(M_e) : M_e \cong \mathcal{A}$ (that is, the function $x \mapsto \Psi(M_e; x)$ is an isomorphism from M_e to $\mathcal{A}$).

$\mathcal{A}$ is *(weakly) u.c.c. with parameters* if there are finitely many elements $a_0, \ldots, a_n \in |\mathcal{A}|$ such that $(\mathcal{A}, a_0, \ldots, a_n)$ is (weakly) u.c.c..

Remark. For each partial computable g there exists a total computable f such that $g(e)\downarrow \Rightarrow \Phi_{f(e)} = \Phi_{g(e)}$. Thus for $\mathcal{A}$ to be weakly u.c.c. it is enough that there exist a partial computable g such that $M_e \cong \mathcal{A} \Rightarrow g(e)\downarrow \wedge \Phi_{g(e)} : M_e \cong \mathcal{A}$.

Uniform computable categoricity (with parameters) was studied by Ventsov [23]. Weak uniform computable categoricity (with parameters) was studied by Kudinov [14, 16], who also defined a different notion of index-based uniformity in [15].

The difference between the two notions defined above can be understood from a programming perspective. A computable structure $\mathcal{A}$ is weakly u.c.c. if there is a computable procedure that, given a program P outputting a structure $\mathcal{B}$ isomorphic to $\mathcal{A}$, produces an isomorphism between $\mathcal{B}$ and $\mathcal{A}$ *possibly using information about* P. On the other hand, $\mathcal{A}$ is u.c.c. if there is a computable procedure that, given a computable structure $\mathcal{B}$ isomorphic to $\mathcal{A}$, produces an isomorphism between $\mathcal{B}$ and $\mathcal{A}$ *without knowledge of the particular program that is outputting* $\mathcal{B}$. Thus another way to think of uniform computable categoricity is as *on-line categoricity.*

We will prove several facts relating the notions defined above to each other and to the notions of computable categoricity and relative computable categoricity. We summarize these as follows. Consider the following statements about a computable structure $\mathcal{A}$.

C: $\mathcal{A}$ is computably categorical.

W: $\mathcal{A}$ is weakly uniformly computably categorical.

WP: $\mathcal{A}$ is weakly uniformly computably categorical with parameters.

U: $\mathcal{A}$ is uniformly computably categorical.

UP: $\mathcal{A}$ is uniformly computably categorical with parameters.

R: $\mathcal{A}$ is relatively computably categorical.

The following implications hold, and no other implications except the ones implied by transitivity hold in general.

$$\begin{array}{ccccc} \mathrm{U} & \Longrightarrow & \mathrm{UP} & \Longleftrightarrow & \mathrm{R} \\ \Downarrow & & \Downarrow & & \\ \mathrm{W} & \Longrightarrow & \mathrm{WP} & \Longrightarrow & \mathrm{C} \end{array}$$

All the arrows in this diagram are obvious except for the equivalence of UP and R, which will be discussed below. If we let $\mathcal{A}$ be ω with successor then it is easy to check that UP holds (and hence so does WP), but W does not hold (and hence neither does U). This justifies the lack of arrows from WP to W and from UP to U. The other missing arrows will be justified below.

If $\mathcal{A}$ is rigid (that is, has no nontrivial automorphisms), or even if it has only finitely many automorphisms, then we will see that the picture is the following.

$$\begin{array}{ccccc} \mathrm{U} & \Longrightarrow & \mathrm{UP} & \Longleftrightarrow & \mathrm{R} \\ \Updownarrow & & \Updownarrow & & \\ \mathrm{W} & \Longrightarrow & \mathrm{WP} & \Longrightarrow & \mathrm{C} \end{array}$$

(Notice that the example of ω with successor given above still works in this case.)

It might seem that relativizing uniform computable categoricity could yield a stronger notion, but we will see below that this is not the case. It might also seem that there could be other natural notions of uniformity strictly between weak uniform computable categoricity and uniform computable categoricity, but we will argue below that this is also not the case. We will also discuss the issue of preservation of these notions under expansion by finitely many constants.

Uniform computable categoricity is a special case of a notion introduced by Ash, Knight, and Slaman in [3], as we now explain.

In [1], Ash and Knight considered the following question. Given a structure $\mathcal{A}$ and an infinitary sentence ψ in the language of $\mathcal{A}$ expanded by a new relation symbol R, when is it the case that for every $\mathcal{B} \cong \mathcal{A}$ there is a $\deg(\mathcal{B})$-computable relation $R^{\mathcal{B}}$ on $\mathcal{B}$ such that $(\mathcal{B}, R^{\mathcal{B}}) \vDash \psi$? As they pointed out, the problem of determining when a structure is relatively computably categorical is a special case of this question.

Indeed, given a structure $\mathcal{A}$, we can form the *cardinal sum* $\widehat{\mathcal{A}}$ of $\mathcal{A}$ with itself (essentially two disjoint copies of $\mathcal{A}$ in a language expanded by two new unary predicate symbols to distinguish the copies). Then there exists a computable infinitary Π^0_2 sentence ψ in the language of $\widehat{\mathcal{A}}$ expanded by a new binary relation symbol R such that $\mathcal{A}$ is relatively computably categorical if and only if for every $\mathcal{B} \cong \widehat{\mathcal{A}}$ there is a $\deg(\mathcal{B})$-computable relation $R^{\mathcal{B}}$ on $\mathcal{B}$ such that $(\mathcal{B}, R^{\mathcal{B}}) \vDash \psi$. (Basically, ψ says of R that it is an isomorphism between the two halves of its underlying structure viewed as a cardinal sum; ψ needs to be infinitary only if the language of $\mathcal{A}$ is infinite. See [1] for details.)

This line of research was continued by Ash, Knight, and Slaman in [3]. As they pointed out, instead of requiring the existence of an appropriate $R^{\mathcal{B}}$ for every $\mathcal{B} \cong \mathcal{A}$, we could restrict our attention to computable $\mathcal{B}$; in the special case of the sentence ψ discussed in the previous paragraph, this corresponds to studying computable categoricity. Another possibility considered in [3] is to require that $R^{\mathcal{B}}$ be given effectively (in the operator sense) from $\mathcal{B}$, and this is the notion that, with the same ψ as above, corresponds to the study of uniform computable categoricity.

We will comment further on what we can conclude about uniform computable categoricity from the results in [3] below, but first we show that uniform computable categoricity can still be viewed as uniformity with respect to indices, but a particularly strong, extensional form of uniformity, in which we are not allowed to cheat by treating two programs with identical outputs in different ways.

There is a similarity between certain aspects of the subject matter of this paper and the study of type 2 computability (that is, computability of operations of type $(\omega \to \omega) \to \omega)$, as noted by Ventsov [24]. In particular, the following result can be seen as an analog of the fundamental theorems of Myhill and Shepherdson [20] and Kreisel, Lacombe, and Shoenfield [13] on the relationship between effective operations and computable functionals. (See Section 15.3 of Rogers [21] for the statements of these theorems as well as a discussion of related concepts and results.)

2.2 Theorem. *Let $\mathcal{A}$ be a computable structure and let $M_0, M_1, \ldots$ be a computable list of all partial computable structures in the language of $\mathcal{A}$. If there is a total computable f such that $M_e \cong \mathcal{A} \Rightarrow \Phi_{f(e)} : M_e \cong \mathcal{A}$ and $M_e = M_i \cong \mathcal{A} \Rightarrow \Phi_{f(e)} = \Phi_{f(i)}$ then $\mathcal{A}$ is u.c.c..*

Proof. Let f be as in the statement of the theorem. Use the Recursion Theorem to define a binary computable function g such that, for each $x \in \omega$, $g_x \equiv \lambda y(g(x,y))$ is such that the partial computable structures $M_{g_x(i)}$, $i \in \omega$, are given by the following procedure.[1]

[1]Readers inexperienced in the use of the Recursion Theorem may find the following more formal definition of g helpful, though it is recommended that the description of g in the body of the proof be read first.

Define the total computable function d so that the partial computable structures $M_{\Phi_{d(e)}(x,2i)}$ and the values $\Phi_{d(e)}(x, 2i+1)$, $e, x, i \in \omega$, are given by the following procedure. ($\Phi_{d(e)}(x, 2i+1)$ is used as a flag; its computation converges when $M_{\Phi_{d(e)}(x,2i)}$ enters its phase 4, as described below.)

The computation of each $M_{\Phi_{d(e)}(x,2i)}$ begins its *phase 1* by emulating M_i. If it is ever the case that, for some $s_{e,x,i} \in \omega$, $\Phi_e(x,2i)[s_{e,x,i}]\downarrow$ and $\Phi_{f(\Phi_e(x,2i))}(x)[s_{e,x,i}]\downarrow = \Phi_{f(i)}(x)[s_{e,x,i}]\downarrow$, then the computation of $M_{\Phi_{d(e)}(x,2i)}$ enters its *phase 2*.

During phase 2, the computation of $M_{\Phi_{d(e)}(x,2i)}$ waits until an embedding of $M_{\Phi_e(x,2i)}[s_{e,x,i}]$ into $\mathcal{A}$ is found (if ever), and then enters its *phase 3*.

Whenever $i \neq j$ and a stage s are found such that

1. $\Phi_e(x, 2i)[s]\downarrow$ and $\Phi_e(x, 2j)[s]\downarrow$,
2. there are embeddings of $M_{\Phi_e(x,2i)}[s]$ and $M_{\Phi_e(x,2j)}[s]$ into $\mathcal{A}$,
3. $\Phi_e(x, 2i+1)[s]\uparrow$ and $\Phi_e(x, 2j+1)[s]\uparrow$,
4. $M_{\Phi_e(x,2i)}[s_i] \subseteq M_{\Phi_e(x,2j)}[s_j]$, and
5. $\Phi_{f(\Phi_e(x,2i))}(x) \neq \Phi_{f(\Phi_e(x,2j))}(x)$,

then the computations of $M_{\Phi_{d(e)}(x,2i)}$ and $M_{\Phi_{d(e)}(x,2j)}$ enter *phase 4*, during which they act as follows. First they search for the least embedding h (in some standard ordering of finite maps) of $M_{\Phi_e(x,2j)}[s_{e,x,j}]$ into $\mathcal{A}$, if any. Then both structures are made to have domain ω and be isomorphic to $\mathcal{A}$ via the isomorphism $k \supset h$ that sends the n^{th} element of $\omega - \operatorname{dom}(h)$ to the n^{th} element of $|\mathcal{A}| - \operatorname{rng}(h)$, if this

The computation of each $M_{g_x(i)}$ begins its *phase 1* by emulating M_i. If it is ever the case that, for some $s_i \in \omega$, $\Phi_{f(g_x(i))}(x)[s_i]\downarrow = \Phi_{f(i)}(x)[s_i]\downarrow$, then the computation of $M_{g_x(i)}$ enters its *phase 2*.

During phase 2, the computation of $M_{g_x(i)}$ waits until an embedding of $M_{g_x(i)}[s_i]$ into $\mathcal{A}$ is found (if ever), and then enters its *phase 3*.

Whenever $i \neq j$ are found such that the computations of $M_{g_x(i)}$ and $M_{g_x(j)}$ are both in phase 3, $M_{g_x(i)}[s_i] \subseteq M_{g_x(j)}[s_j]$ and $\Phi_{f(g_x(i))}(x) \neq \Phi_{f(g_x(j))}(x)$ then the computations of $M_{g_x(i)}$ and $M_{g_x(j)}$ enter *phase 4*, during which, for the least embedding h (in some standard ordering of finite maps) of $M_{g_x(j)}[s_j]$ into $\mathcal{A}$, both structures are made to have domain ω and be isomorphic to $\mathcal{A}$ via the isomorphism $k \supset h$ that sends the n^{th} element of $\omega - \text{dom}(h)$ to the n^{th} element of $|\mathcal{A}| - \text{rng}(h)$.

This completes the definition of g. We compute $\Psi(M; x)$ as follows, where M is a partial structure in the language of $\mathcal{A}$. First, ask whether $x \in |M|$. If not then make $\Psi(M; x)$ diverge. Otherwise, search for $i, s \in \omega$ such that $\Phi_{f(g_x(i))}(x)[s]\downarrow$ and $M_{g_x(i)}[s] = M \upharpoonright \left|M_{g_x(i)}[s]\right|$. If such i and s are found then let $\Psi(M; x) = \Phi_{f(g_x(i))}(x)$.

Clearly, Ψ is a partial computable operator. Let $M_e \cong \mathcal{A}$. We need to show that $\Psi(M_e) : M_e \cong \mathcal{A}$.

Consider $M_{g_x(e)}$. If the computation of this structure never reaches its second phase then $M_{g_x(e)} = M_e$, so we have $M_{g_x(e)} \cong \mathcal{A}$, $x \in \left|M_{g_x(e)}\right|$, but not $\Phi_{f(g_x(e))}(x)\downarrow = \Phi_{f(e)}(x)\downarrow$, which contradicts the definition of f. So the computation of $M_{g_x(e)}$ reaches its second phase, which implies that it reaches its third phase, since $M_e \cong \mathcal{A}$.

However, for all $i \in \omega$, the computation of $M_{g_x(i)}$ cannot reach its fourth phase, since that would mean that there exists a $j \in \omega$ such that $M_{g_x(i)} = M_{g_x(j)} \cong \mathcal{A}$ but $\Phi_{f(g_x(j))} \neq \Phi_{f(g_x(i))}$, which again contradicts the definition of f.

So the computation of $M_{g_x(e)}$ reaches its third phase but never leaves it, which means that $M_{g_x(e)}$ is finite. So there must exist $i, s \in \omega$ such that $\Phi_{f(g_x(i))}(x)[s]\downarrow$ and $M_{g_x(i)}[s] = M_e \upharpoonright \left|M_{g_x(i)}[s]\right|$. (We can take $i = e$, for example.) Since we now define $\Psi(M; x) = \Phi_{f(g_x(i))}(x)$ for the first such i and s that we find, it is enough to show that $\Phi_{f(g_x(i))}(x) = \Phi_{f(e)}(x)$ for any such i. But if $\Phi_{f(g_x(i))}(x) \neq \Phi_{f(e)}(x)$ then, since the fact that the computation of $M_{g_x(e)}$ reaches its second phase implies that $\Phi_{f(g_x(e))}(x) = \Phi_{f(e)}(x)$, it follows that $\Phi_{f(g_x(i))}(x) \neq \Phi_{f(g_x(e))}(x)$, which implies that at

is possible (otherwise, both structures stop growing). Finally, $\Phi_{d(e)}(x, 2i+1)$ and $\Phi_{d(e)}(x, 2j+1)$ are both made to converge at stage s.

By the Recursion Theorem, there is an $e \in \omega$ such that $\Phi_e = \Phi_{d(e)}$. For such an e, define $g(x, i) = \Phi_e(x, 2i)$. It is now straightforward to check that g behaves as described in the body of the proof.

least one of the computations of $M_{g_x(e)}$ and $M_{g_x(i)}$ reaches its fourth phase, which we have already argued is not the case. $\square$

One consequence of Theorem 2.2 is that there does not appear to be any natural notion of uniformity strictly between weak uniform computable categoricity and uniform computable categoricity. Another consequence is that these two notions agree on rigid structures; the following is a stronger version of this fact.

2.3 Corollary. *Let $\mathcal{A}$ be a structure with only finitely many automorphisms. Then $\mathcal{A}$ is weakly u.c.c. (with parameters) if and only if it is u.c.c. (with parameters).*

Proof. It is enough to prove the parameter-free version of the corollary. Let $M_0, M_1, \ldots$ be a computable list of all partial computable structures in the language of $\mathcal{A}$ and let f be as in the definition of weakly u.c.c.. We will build a total computable function $\hat{f}$ satisfying the hypothesis of Theorem 2.2.

Let $g_0, \ldots, g_n$ be the automorphisms of $\mathcal{A}$. Let S be the set of all $j \leqslant n$ such that g_j is computable. Note that if $M_e = M_i$ but $\Phi_{f(e)} \neq \Phi_{f(i)}$ then $\Phi_{f(e)} \circ (\Phi_{f(i)})^{-1} = g_j$ for some $j \leqslant n$, and in this case $j \in S$. Clearly there exist a finite $D \subset |\mathcal{A}|$ and pairwise distinct functions $h_0, \ldots, h_n : D \to |\mathcal{A}|$ such that, for each $i \leqslant n$, g_i extends h_i. Let $d_0 < \cdots < d_m$ be the elements of D.

Define $\hat{f}$ so that $\Phi_{\hat{f}(e)}$ is given by the following procedure. First, wait until all $h_j(D)$, $j \in S$, appear in the range of $\Phi_{f(e)}$. For each $j \in S$ and $i \leqslant m$, let x_i^j be such that $\Phi_{f(e)}(x_i^j) = h_j(d_i)$. Choose a $j \in S$ that minimizes the tuple $\langle x_0^j, \ldots, x_m^j \rangle$ in some fixed ordering of tuples. Now emulate $g_j \circ \Phi_{f(e)}$.

Clearly $\hat{f}$ can be chosen to be total computable, and if $M_e \cong A$ then $\Phi_{\hat{f}(e)} \equiv g_j \circ \Phi_{f(e)}$ for some $j \in S$, which implies that $\Phi_{\hat{f}(e)} : M_e \cong \mathcal{A}$. Finally, if $M_e = M_i \cong \mathcal{A}$ then it is easy to check that $\Phi_{\hat{f}(e)}^{-1}(D) = \Phi_{\hat{f}(i)}^{-1}(D)$, which implies that $\Phi_{f(e)} \circ (\Phi_{f(i)})^{-1} \upharpoonright D$ is the identity, and hence that $\Phi_{f(e)} = \Phi_{f(i)}$. By Theorem 2.2, $\mathcal{A}$ is u.c.c.. $\square$

The following is a consequence of the results in [3] for computable infinitary Π_2^0 sentences, and also follows from Theorem 2.5 below.

2.4 Corollary. *A computable structure is u.c.c. with parameters if and only if it is relatively computably categorical.*

It now follows from Theorem 1.4 that a computable structure is u.c.c. with parameters if and only if it has a Scott family. The following more general result of Ventsov [23] is also a consequence of results in [3]. We include a proof for the sake of establishing Corollary 2.6 below.

2.5 Theorem (Ventsov). *A computable structure is u.c.c. if and only if it has a Scott family without parameters, and is u.c.c. with parameters if and only if it has a Scott family.*

Proof. It is enough to prove the first part of the theorem, from which the second part follows immediately. Let $\mathcal{A}$ be a computable structure and let $M_0, M_1, \ldots$ be a computable list of all partial computable structures in the language of $\mathcal{A}$.

If $\mathcal{A}$ is u.c.c. then let Ψ be as in Definition 2.1. Given $\vec{x} = (x_0, \ldots, x_k) \in |\mathcal{A}|^{k+1}$, let $M_{e(\vec{x})}$ be a finite substructure of $\mathcal{A}$ such that, for every $i \leqslant k$, $x_i \in \left|M_{e(\vec{x})}\right|$ and $\Psi(M_{e(\vec{x})}; x_i)\downarrow$. Let $y_0, \ldots, y_n$ be the elements of $\left|M_{e(\vec{x})}\right|$ other than $x_0, \ldots, x_k$ and let $\delta(\vec{x}, y_0, \ldots, y_n)$ be the conjunction of the finitely many elements of the atomic diagram of $M_{e(\vec{x})}$. Define $\theta_{\vec{x}} \equiv \exists y_0, \ldots, y_n(\delta(\vec{x}, y_0, \ldots, y_n))$.

We claim that $\{\theta_{\vec{x}} \mid \vec{x} \in |\mathcal{A}|^{<\omega}\}$ is a Scott family. It is enough to show that if $\mathcal{A} \vDash \theta_{\vec{x}}(\vec{y})$ then $\vec{y}$ is in the orbit of $\vec{x}$. If $\mathcal{A} \vDash \theta_{\vec{x}}(\vec{y})$ then there is an i and a g such that $g : \mathcal{A} \cong M_i$, $g(\vec{y}) = \vec{x}$, and $M_{e(\vec{x})}$ is a substructure of M_i. It is easy to check that $(\Psi(\mathcal{A}))^{-1} \circ \Psi(M_i) \circ g$ is an automorphism of $\mathcal{A}$ taking $\vec{y}$ to $\vec{x}$.

For the other direction, we can use the standard proof that a structure with a Scott family is computably categorical, since this proof produces computable isomorphisms uniformly.

Suppose that $\mathcal{A}$ has a Scott family without parameters $\{\theta_n \mid n \in \omega\}$. The operator Ψ acts as follows on a structure $\mathcal{B}$.

Assuming that $x \in |\mathcal{B}|$, $\Psi(\mathcal{B}; y)$ has been defined for all $y \in |\mathcal{B}| \upharpoonright x$, and $\Psi(\mathcal{B}; x)$ has not yet been defined, Ψ searches for a θ_n such that $\mathcal{B} \vDash \theta_n(x_0, \ldots, x_{k-1}, x)$, where $x_0 < \cdots < x_{k-1}$ are the elements of $|\mathcal{B}| \upharpoonright x$. If such a formula is found then Ψ searches for a $z \in |\mathcal{A}|$ such that $\mathcal{A} \vDash \theta_n(\Psi(\mathcal{B}; x_0), \ldots, \Psi(\mathcal{B}; x_{k-1}), x)$. If such a z is found then $\Psi(\mathcal{B}; x) = z$.

Now Ψ finds the least $w \in |\mathcal{A}|$ that is not currently in the range of $\Psi(\mathcal{B})$, searches for a θ_m such that $\mathcal{A} \vDash \theta_m(\Psi(\mathcal{B}; x_0), \ldots, \Psi(\mathcal{B}; x_{k-1}), z, w)$, and then searches for a $v \in |\mathcal{B}|$ that is not yet in the domain of $\Psi(\mathcal{B})$ such that $\mathcal{B} \vDash \theta_m(x_0, \ldots, x_{k-1}, x, v)$. If such a v is found then $\Psi(\mathcal{B}; v) = w$.

It is easy to check, using the definition of Scott family, that if $\mathcal{B} \cong \mathcal{A}$ then this procedure guarantees that $\Psi(\mathcal{B})$ is an isomorphism from $\mathcal{B}$ to $\mathcal{A}$. □

In the second half of the proof of Theorem 2.5, there is no need for $\mathcal{B}$ to be computable. This allows us to conclude that relativizing the notion of uniform computable categoricity does not make it any stronger.

2.6 Corollary. *If $\mathcal{A}$ is u.c.c. then there is a partial computable operator Ψ such that, for any presentation $\mathcal{B}$ of $\mathcal{A}$, $\Psi(\mathcal{B}) : \mathcal{B} \cong \mathcal{A}$.*

Another consequence of Theorem 2.5 is that uniform computable categoricity, unlike computable categoricity, is always preserved under expansion by finitely many constants. (This is because if a structure has a Scott family then so does any expansion by finitely many constants; see [12] for details.)

2.7 Corollary. *If $\mathcal{A}$ is u.c.c. (with parameters) and $a \in |\mathcal{A}|$ then $(\mathcal{A}, a)$ is u.c.c. (with parameters).*

Putting together Theorem 2.5 with Corollary 2.3 and the following result of Khoussainov and Shore [12], allows us to conclude that weak uniform computable categoricity is a strictly stronger notion than computable categoricity, a result originally due to Kudinov [14].

2.8 Theorem (Khoussainov and Shore). *There is a rigid computably categorical structure with no Scott family.*

2.9 Corollary (Kudinov). *There is a computably categorical structure that is not weakly u.c.c. with parameters.*

We have yet to show that weak uniform computable categoricity and uniform computable categoricity are different notions. This has been done by Kudinov [16], using a result of V'yugin [25]. We give a direct construction below.

2.10 Theorem (Kudinov). *There is a weakly u.c.c. structure with no Scott family.*

Proof. For each $n \in \omega$, let $[n]$ be the directed graph consisting of $n+3$ many nodes $x_0, x_1, \dots, x_{n+2}$ with an edge from x_0 to itself, an edge from x_{n+2} to x_1, and an edge from x_i to x_{i+1} for each $i \leqslant n+1$. We call x_0 the *top* of $[n]$.

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

Now let $D_0, D_1, \dots$ be a standard list of all finite non-empty sets and let $B^n_0, B^n_1, \dots$ be the n^{th} uniformly c.e. (u.c.e.) collection of sets in some standard ordering of such collections. We will build a u.c.e. collection of finite sets $A_0, A_1, \dots$ with the following properties.

1. There is no c.e. set W such that every A_i contains D_k for some $k \in W$ and $(k \in W \wedge D_k \subseteq A_i, A_j) \Rightarrow A_i = A_j$.

2. Let N be the set of all $i \in \omega$ such that A_i is non-empty. There is a partial computable binary function h with the following property. For each $n \in \omega$, if there is a 1–1 and onto map $g : \omega \to N$ such that, for all $i \in \omega$, $B_i^n = A_{g(i)}$, then $h_n \equiv \lambda i(h(n,i))$ is such a map.

Suppose we have built $A_0, A_1, \ldots$ with these properties and let $\mathcal{A}$ be a computable directed graph consisting of the disjoint union of the graphs $[A_i]$, $i \in \omega$. It is not hard to check that the first property above implies that $\mathcal{A}$ has no Scott family, while the second property implies that $\mathcal{A}$ is weakly u.c.c..

We now proceed with the construction of $A_0, A_1, \ldots$. We assume that if $s < \langle n, i\rangle$ then $B_i^n[s] = \emptyset$. Since we are only interested in collections of sets $B_0, B_1, \ldots$ for which there is a 1–1 and onto map $g : \omega \to N$ such that, for all $i \in \omega$, $B_i = A_{g(i)}$, we also assume without loss of generality that for each $n, i, s \in \omega$ there is a $j \in \omega$ such that $B_i^n[s] \subseteq A_j[s]$. For each $e \in \omega$, let $k(e)$ be such that $D_{k(e)} = \{2e\}$. We begin with $A_i = \emptyset$ for all $i \in \omega$. At stage s, we proceed as follows.

1. Enumerate $2s$ into $A_{\langle s,0\rangle}$.

2. Search for the least $e \leqslant s$ (if any) such that $k(e) \in W_e$ and $A_{\langle e,0\rangle} = \{2e\}$. If such an e is found then enumerate $2\langle e, 0\rangle + 1$ into $A_{\langle e,0\rangle}$ and enumerate $2e$ and $2\langle e, 1\rangle + 1$ into $A_{\langle e,1\rangle}$. Say that e is *active* at stage s.

3. For each $e \leqslant s$, check if there exist $n, i \in \omega$ such that $\langle n, i\rangle \leqslant s$, e has never caught-up for the sake of n before (defined below), $2e \in B_i^n[t]$, where $t \leqslant s$ is a stage at which e is active, and $B_i^n[s] \neq \{2e\}$. If so then proceed as follows. For each $i \leqslant s$ such that $2e \notin A_{\langle e,i\rangle}[s]$, enumerate $2e$ into $A_{\langle e,i\rangle}$. For each $i, j \leqslant s$ such that $2\langle e, j\rangle + 1 \notin A_{\langle e,i\rangle}[s]$, enumerate $2\langle e, j\rangle + 1$ into $A_{\langle e,i\rangle}$. Enumerate $2e$ and $2\langle e, s + 1\rangle + 1$ into $A_{\langle e,s+1\rangle}$. Say that e *catches-up for the sake of* n at stage s.

This completes the construction. We now check that $A_0, A_1, \ldots$ have the desired properties. Clearly, $A_0, A_1, \ldots$ are u.c.e.. Each $e \in \omega$ can be active at most once, since e cannot be active unless $A_{\langle e,0\rangle}$ is a singleton, and once e is active this is no longer the case. Furthermore, if e is active at stage s then e can only catch-up at most once for each n such that $2e \in B_i^n[s]$ for some $i \in \omega$, which implies that e catches-up only finitely often. Since no numbers are enumerated into any $A_{\langle e,i\rangle}$, $i \in \omega$, at any stage at which e is neither active nor catches-up, this means that each A_i is finite.

Fix $e \in \omega$. If $k(e) \notin W_e$ then e is never active, so $A_{\langle e,0\rangle} = \{2e\}$, which means that there is no $k \in W_e$ such that $D_k \subseteq A_{\langle e,0\rangle}$. On the other hand, if $k(e) \in W_e$ then either

e never catches-up, in which case both $A_{\langle e,0\rangle}$ and $A_{\langle e,1\rangle}$ contain $D_{k(e)}$ but $A_{\langle e,0\rangle} \neq A_{\langle e,1\rangle}$, or e catches-up for the last time at some stage s, in which case, both $A_{\langle e,0\rangle}$ and $A_{\langle e,s+1\rangle}$ contain $D_{k(e)}$ but $A_{\langle e,0\rangle} \neq A_{\langle e,s+1\rangle}$. Thus there is no c.e. set W such that every A_i contains D_k for some $k \in W$ and $(k \in W \wedge D_k \subseteq A_i, A_j) \Rightarrow A_i = A_j$.

Now define the map h as follows. Given $n, i \in \omega$, assume that $h(n,j)$ has already been defined for all $j < i$. Wait until a stage s such that $B_i^n[s]$ is non-empty and equal to $A_k[s]$ for some $k \notin \operatorname{rng}(h \restriction i)$ and define $h(n,i) = k$. Clearly, h is partial computable.

Fix $n \in \omega$ for which there is a 1–1 and onto map $g : \omega \to N$ such that, for all $i \in \omega$, $B_i^n = A_{g(i)}$. (Note that this implies that each B_i^n is non-empty.) To complete the proof, we need to show that $h_n \equiv \lambda i(h(n,i))$ is a 1–1 map from ω onto N such that, for all $i \in \omega$, $B_i = A_{h_n(i)}$.

Clearly, if h_n is defined for all $i \in \omega$ then it is 1–1 and onto. (The surjectivity of h_n follows from the fact that, for each $e \in \omega$, there are only finitely many $k \in \omega$ such that $2e \in A_k$.) Suppose that, for all $j < i$, $h_n(j)$ is defined and $B_j^n = A_{h_n(j)}$. Then $h_n(i)$ must be defined, since otherwise there could be no 1–1 and onto map $g : \omega \to N$ such that, for all $l \in \omega$, $B_l^n = A_{g(l)}$.

So we are left with showing that, for all $i \in \omega$, $B_i^n = A_{h_n(i)}$. Fix $i \in \omega$ and let $k = h_n(i)$. First suppose that s in the definition of $h(i)$ is such that $B_i^n[s] = A_k[s]$ is not a singleton. It is easy to check from the construction that if A_l and A_s share two elements then they are equal, so we have $B_i^n = A_{g(i)} = A_k$.

Now suppose that s in the definition of $h(i)$ is such that $B_i^n[s] = A_k[s]$ is a singleton. Then it must be the case that, for some $e \in \omega$, $B_i^n[s] = \{2e\}$ and $k = \langle e, 0\rangle$. If e is never active then $B_i^n = \{2e\} = A_k$. Otherwise, every A_l that contains $2e$ has at least two elements, and thus so does B_i^n. But this means that, at some stage $t \geqslant s$, e catches-up for the sake of n. It is easy to check that this means that $A_l \supseteq B_i^n[t]$ if and only if $l = \langle e, u\rangle$, $u \leqslant t$. But it is also the case that $A_{\langle e,u\rangle} = A_{\langle e,v\rangle}$ for all $u, v \leqslant t$, so in fact $A_l = B_i^n[t]$ if and only if $l = \langle e, u\rangle$, $u \leqslant t$. In particular, $B_i^n = A_k$. $\square$

2.11 Corollary. *There is a weakly u.c.c. structure that is not u.c.c. with parameters.*

Theorem 2.10 raises the possibility that, unlike uniform computable categoricity, weak uniform computable categoricity might not always be preserved under expansion by finitely many constants. The next result shows that, if so, then the constants by which we expand our structure cannot have simple orbits. In particular, it implies that we cannot hide a structure of finite computable dimension greater than 1 within a weakly u.c.c. structure in the same way that we can in some computably categorical structures.

2.12 Theorem. *If $\mathcal{A}$ is weakly u.c.c. (with parameters) and the orbit of $a \in |\mathcal{A}|$ is computable then $(\mathcal{A}, a)$ is weakly u.c.c. (with parameters).*

Proof. It is enough to prove the parameter-free version of the theorem. Let $M_0, M_1, \dots$ be a computable list of all partial computable structures in the language of $(\mathcal{A}, a)$. Since $\mathcal{A}$ is weakly u.c.c., there is a total computable f such that if $M_e \cong (\mathcal{A}, a)$ then, for some b in the orbit of a, $\Phi_{f(e)} : M_e \cong (\mathcal{A}, b)$. We define a partial computable function $\hat{f}$ such that if $M_e \cong (\mathcal{A}, a)$ then $\hat{f}(e)\downarrow$ and $\Phi_{\hat{f}(e)} : M_e \cong (\mathcal{A}, a)$. By the remark following Definition 2.1, this is enough to show that $(\mathcal{A}, a)$ is weakly u.c.c..

Given $e \in \omega$, build M_i as follows, using the Recursion Theorem. Begin by copying $\mathcal{A}$. Whenever a $c \in |M_i|$ is found such that $\Phi_{f(i)}(c) = \Phi_{f(e)}(a^{M_e})$, stop building M_i and ask whether c is in the orbit of a. If not then continue to copy $\mathcal{A}$. Otherwise, search for an embedding g of the currently enumerated part of M_i into $\mathcal{A}$ such that $g(c) = a$. If such an embedding is found then continue to build M_i to be isomorphic to $\mathcal{A}$ via a computable isomorphism h extending g and define $\hat{f}(e)$ to be such that $\Phi_{\hat{f}(e)} \equiv h \circ (\Phi_{f(i)})^{-1} \circ \Phi_{f(e)}$.

Clearly, $\hat{f}$ is partial computable. If $M_e \cong \mathcal{A}$ then $\Phi_{f(e)}(a^{M_e})$ is in the orbit of a, and hence the building of M_i goes through the full procedure described above. That is, c is found and is in the orbit of a, g is found, and $\hat{f}(e)$ is defined. It is easy to check that this implies that $\Phi_{\hat{f}(e)}$ is an isomorphism from M_e to $(\mathcal{A}, a)$. $\square$

It is not hard to modify the proof of Theorem 2.12 to show that we can replace *computable* by Σ^0_2 in the statement of that theorem, but the following question is open.

2.13 Question. If $\mathcal{A}$ is weakly u.c.c. and $a \in |\mathcal{A}|$ then must $(\mathcal{A}, a)$ be weakly u.c.c.?

3 Relations on Computable Structures

We now turn our attention to relations on computable structures. Although we are no longer in the framework of [3], we will discuss results that are quite similar to those of the previous section. (We will summarize these results at the end of this section.)

The study of relations on computable structures began with the work of Ash and Nerode [4], who were concerned with relations that maintain some degree of effectiveness in different computable presentations of a structure.

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

Throughout this section, we will restrict ourselves to *invariant* relations, that is, relations preserved under all automorphisms of the underlying structure, since there do not seem to be reasonable notions of uniformity when dealing with noninvariant relations.

We begin by considering the uniform analog of intrinsic computability. As we will see, in this case our two notions of uniformity coincide.

3.2 Definition. Let $\mathcal{A}$ be a computable structure and let $M_0, M_1, \ldots$ be a computable list of all partial computable structures in the language of $\mathcal{A}$. Let U be an invariant k-ary relation on $\mathcal{A}$ and let $\Phi_0, \Phi_1, \ldots$ be a standard list of all k-ary partial computable functions.

U is *uniformly intrinsically computable* if there is a total computable f such that $M_e \cong \mathcal{A} \Rightarrow \Phi_{f(e)} = U^{M_e}$.

U is *uniformly intrinsically computable with parameters* if there are finitely many elements $a_0, \ldots, a_n \in |\mathcal{A}|$ such that U is uniformly intrinsically computable when considered as a relation on $(\mathcal{A}, a_0, \ldots, a_n)$.

The above definition is similar to one in Ventsov [24] (and is in fact equivalent to Ventsov's definition, by results in that paper and Corollary 3.9 below).

3.3 Theorem. *Let $\mathcal{A}$ be a computable structure and let $M_0, M_1, \ldots$ be a computable list of all partial computable structures in the language of $\mathcal{A}$. Let U be an invariant k-ary relation on $\mathcal{A}$. If U is uniformly intrinsically computable then there is a partial computable operator Ψ such that $M_e \cong \mathcal{A} \Rightarrow \Psi(M_e) = U^{M_e}$.*

Proof. Repeat the proof of Theorem 2.2 using the function f given in Definition 3.2. Note that, since U is invariant, it is automatically the case that $M_e = M_i \cong \mathcal{A} \Rightarrow \Phi_{f(e)} = \Phi_{f(i)}$. $\square$

Turning now to the concept of intrinsic computable enumerability, we find that once again we have two different notions of uniformity.

3.4 Definition. Let $\mathcal{A}$ be a computable structure and let $M_0, M_1, \ldots$ be a computable list of all partial computable structures in the language of $\mathcal{A}$. Let U be an invariant k-ary relation on $\mathcal{A}$ and let $W_0, W_1, \ldots$ be a standard list of all c.e. sets of k-tuples.

U is *weakly uniformly intrinsically c.e.* if there is a total computable f such that $M_e \cong \mathcal{A} \Rightarrow W_{f(e)} = U^{M_e}$.

U is *uniformly intrinsically c.e.* if there is a c.e. operator W such that $M_e \cong \mathcal{A} \Rightarrow W(M_e) = U^{M_e}$.

U is *(weakly) uniformly intrinsically c.e. with parameters* if there are finitely many elements $a_0, \ldots, a_n \in |\mathcal{A}|$ such that U is (weakly) uniformly intrinsically c.e. when considered as a relation on $(\mathcal{A}, a_0, \ldots, a_n)$.

Ventsov [23] studied uniform intrinsic computable enumerability (with parameters). Ventsov [24] also studied a notion similar to weak uniform intrinsic computable enumerability (with parameters).

It is interesting to note that a finite relation need not be weakly uniformly intrinsically c.e. (though it is, of course, always uniformly intrinsically computable with parameters). An example of such a relation is the unary relation on ω with successor that holds only of 0.

The following is a consequence of Theorem 3.3.

3.5 Corollary. *Let U be a relation on a computable structure $\mathcal{A}$. The following are equivalent.*

1. *U is uniformly intrinsically computable (with parameters).*
2. *Both U and its complement are weakly uniformly intrinsically c.e. (with parameters).*
3. *Both U and its complement are uniformly intrinsically c.e. (with parameters).*

As in the case of computable categoricity, the operator notion of uniformity corresponds to a strengthened index-based uniformity.

3.6 Theorem. *Let $\mathcal{A}$ be a computable structure and let $M_0, M_1, \ldots$ be a computable list of all partial computable structures in the language of $\mathcal{A}$. Let U be an invariant k-ary relation on $\mathcal{A}$ and let $\Phi_0, \Phi_1, \ldots$ be a standard list of all $(k+1)$-ary partial computable functions. If there is a total computable f such that $M_e \cong \mathcal{A} \Rightarrow (\Phi_{f(e)}$ total $\wedge \{\vec{x} \mid \exists n(\Phi_{f(e)}(\vec{x}, n) = 1)\} = U^{M_e})$ and $M_e = M_i \cong \mathcal{A} \Rightarrow \Phi_{f(e)} = \Phi_{f(i)}$ then U is uniformly intrinsically c.e..*

Proof. The proof is essentially the same as that of Theorem 2.2. Instead of building a binary function g, we build a $(k+2)$-ary function g defined by specifying structures $M_{g_{\vec{x},n}(i)}$, where $g_{\vec{x},n} \equiv \lambda y(g(\vec{x}, n, y))$, much as before, with certain obvious changes which we leave to the reader.

To enumerate $W(M)$, we search for $\vec{x}$, n, i, and s such that $\Phi_{f(g_{\vec{x},n}(i))}(\vec{x},n)[s]\downarrow = 1$ and $M_{g_{\vec{x},n}(i)}[s] = M \upharpoonright \left|M_{g_{\vec{x},n}(i)}[s]\right|$. Whenever such numbers are found, we enumerate $\vec{x}$ into $W(M)$.

We can now show as before that if $M_e \cong A$ then, for each $\vec{x}$ and n, there exist i and s such that $\Phi_{f(g_{\vec{x},n}(i))}(\vec{x},n)[s]\downarrow$ and $M_{g_{\vec{x},n}(i)}[s] = M \upharpoonright \left|M_{g_{\vec{x},n}(i)}[s]\right|$, and that, for any such i, $\Phi_{f(g_{\vec{x},n}(i))}(\vec{x},n) = \Phi_e(\vec{x},n)$, which implies that $W(M_e) = U^{M_e}$. $\square$

As was the case with uniform computable categoricity, uniform intrinsic computability and uniform intrinsic computable enumerability correspond to natural syntactic notions, in this case ones originally formulated by Ash and Nerode [4]. This was shown by Ventsov [23]. We include a proof below for completeness.

3.7 Definition. A k-ary relation U on a computable structure $\mathcal{A}$ is *formally c.e.* if there is a c.e. sequence $\theta_0, \theta_1, \ldots$ of existential formulas in the language of $\mathcal{A}$ expanded by finitely many constants from $\mathcal{A}$ such that, for every $\vec{x} \in \omega^k$, $U(\vec{x}) \Leftrightarrow \mathcal{A} \vDash \bigvee_{n\in\omega} \theta_n(\vec{x})$.

A relation U on a computable structure is *formally computable* if both it and its complement are formally c.e..

3.8 Theorem (Ventsov). *An invariant relation on a computable structure is uniformly intrinsically c.e. if and only if it is formally c.e. without parameters, and is uniformly intrinsically c.e. with parameters if and only if it is formally c.e..*

Proof. It is enough to prove the first part of the theorem. Let U be an invariant relation on a computable structure $\mathcal{A}$ and let k be its arity. Let $M_0, M_1, \ldots$ be a computable list of all partial computable structures in the language of $\mathcal{A}$.

If U is uniformly intrinsically c.e. then let W be as in Definition 3.4. For each $\vec{x} \in |\mathcal{A}|^k$, search for an $e(\vec{x})$ such that $M_{e(\vec{x})}$ is a finite substructure of $\mathcal{A}$, $\vec{x} \in \left|M_{e(\vec{x})}\right|^k$, and $\vec{x} \in W(M_{e(\vec{x})})$. If such an $e(\vec{x})$ is found, which will happen if and only if $\vec{x} \in U$, then let $y_0, \ldots, y_n$ be the elements of $\left|M_{e(\vec{x})}\right|$ other than the elements of $\vec{x}$ and let $\delta(\vec{x}, y_0, \ldots, y_n)$ be the conjunction of the finitely many elements of the atomic diagram of $M_{e(\vec{x})}$. Define $\theta_{\vec{x}} \equiv \exists y_0, \ldots, y_n(\delta(\vec{x}, y_0, \ldots, y_n))$.

We claim that $U(\vec{y}) \Leftrightarrow \mathcal{A} \vDash \bigvee_{\vec{x}\in U} \theta_{\vec{x}}(\vec{y})$. Clearly, $U(\vec{y}) \Rightarrow A \vDash \bigvee_{\vec{x}\in U} \theta_{\vec{x}}(\vec{y})$. On the other hand, if $A \vDash \theta_{\vec{x}}(\vec{y})$ then there is an i and a g such that $g : \mathcal{A} \cong M_i$, $g(\vec{y}) = \vec{x}$, and $M_{e(\vec{x})}$ is a substructure of M_i. It is easy to check that $\vec{x} \in W(M_{e(\vec{x})}) \Rightarrow \vec{x} \in W(M_i) \Rightarrow U^{M_i}(\vec{x}) \Rightarrow U(\vec{y})$.

As in Theorem 2.5, for the other direction we can adapt the standard proof. Suppose that U is formally c.e. as witnessed by the existential formulas $\{\theta_n \mid n \in \omega\}$. The operator W acts as follows on a structure $\mathcal{B}$.

Given $\vec{x} \in |\mathcal{B}|^k$, W searches for a θ_n such that $\mathcal{B} \models \theta_n(\vec{x})$. If such a formula is found then W enumerates $\vec{x}$ into $W(\mathcal{B})$. It is easy to check that if $\mathcal{B} \cong \mathcal{A}$ then this procedure guarantees that $W(\mathcal{B}) = U^{\mathcal{B}}$. $\square$

Combining Corollary 3.5 and Theorem 3.8, we have the following result.

3.9 Corollary. *An invariant relation on a computable structure is uniformly intrinsically computable if and only if it is formally computable without parameters, and is uniformly intrinsically computable with parameters if and only if it is formally computable.*

There are natural relativized versions of the notions we are considering.

3.10 Definition. Let U be a relation on the domain of a structure $\mathcal{A}$. U is *relatively intrinsically computable* (resp. *c.e.*) on $\mathcal{A}$ if the image of U in any presentation of $\mathcal{A}$ is computable (resp. c.e.) in the degree of the presentation.

Combining Theorem 3.8 and Corollary 3.9 with the following result from [2] and [6], we see that, in this context also, relativization and the operator version of uniformity (with parameters) have the same effect.

3.11 Theorem (Ash, Knight, Manasse, and Slaman; Chisholm). *A relation on a structure is relatively intrinsically c.e. if and only if it is formally c.e..*

3.12 Corollary. *An invariant relation on a computable structure is uniformly intrinsically c.e. with parameters if and only if it is relatively intrinsically c.e., and it is uniformly intrinsically computable with parameters if and only if it is relatively intrinsically computable.*

As in the case of computable categoricity, the fact that, in the second half of the proof of Theorem 3.8, we did not need to require that $\mathcal{B}$ be computable allows us to conclude that relativizing the notions of uniform intrinsic computability and uniform intrinsic computable enumerability does not make them any stronger.

3.13 Corollary. *If an invariant k-ary relation U on the domain of a computable structure $\mathcal{A}$ is uniformly intrinsically c.e. then there is a c.e. operator W such that, for any presentation $\mathcal{B}$ of $\mathcal{A}$, $W(\mathcal{B}) = U^{\mathcal{B}}$.*

If an invariant k-ary relation U on the domain of a computable structure $\mathcal{A}$ is uniformly intrinsically computable then there is a partial computable operator Ψ such that, for any presentation $\mathcal{B}$ of $\mathcal{A}$, $\Psi(\mathcal{B}) = U^{\mathcal{B}}$.

We have yet to show that intrinsic computability does not imply weak uniform intrinsic computable enumerability with parameters and that weak uniform intrinsic computable enumerability does not imply uniform intrinsic computable enumerability with parameters. Fortunately, this has already been done for us.

Manasse [17] built a relation U on a computable structure such that U is intrinsically computable but not formally computable. Combining this result with Corollaries 3.5 and 3.9, we see that it cannot be the case that both U and its complement are weakly uniformly intrinsically c.e. with parameters.

3.14 Corollary. *There is a relation on a computable structure that is intrinsically computable but not weakly uniformly intrinsically c.e. with parameters.*

In Theorem III of [5], Chisholm built a relation on a computable structure that is intrinsically c.e. but not formally c.e.. It is not hard to check from his proof that this relation is in fact weakly uniformly intrinsically c.e.. Together with Theorem 3.8, this result allows us to conclude that weak uniform intrinsic computable enumerability and uniform intrinsic computable enumerability are different notions, which was also shown directly in Ventsov [24], using a result of V'yugin [25].

3.15 Corollary (Ventsov). *There is a relation on a computable structure that is weakly uniformly intrinsically c.e. but not uniformly intrinsically c.e. with parameters.*

We summarize the results of this section as follows. Consider the following statements about a relation U on a computable structure.

C: U is intrinsically computable.

CE: U is intrinsically c.e..

UC: U is uniformly intrinsically computable.

UCP: U is uniformly intrinsically computable with parameters.

WUCE: U is weakly uniformly intrinsically c.e..

WUCEP: U is weakly uniformly intrinsically c.e. with parameters.

UCE: U is uniformly intrinsically c.e..

UCEP: U is uniformly intrinsically c.e. with parameters.

RC: U is relatively intrinsically computable.

RCE: U is relatively intrinsically c.e..

The following implications hold, and no other implications except the ones implied by transitivity hold in general.

$$\begin{tikzcd}
\mathrm{UC} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \mathrm{UCP} \arrow[r, Leftrightarrow] \arrow[d, Rightarrow] & \mathrm{RC} \arrow[r, Rightarrow] & \mathrm{C} \arrow[dd, Rightarrow] \\
\mathrm{UCE} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \mathrm{UCEP} \arrow[r, Leftrightarrow] \arrow[d, Rightarrow] & \mathrm{RCE} & \\
\mathrm{WUCE} \arrow[r, Rightarrow] & \mathrm{WUCEP} \arrow[rr, Rightarrow] & & \mathrm{CE}
\end{tikzcd}$$

4 Suggestions for Further Work

In addition to Question 2.13 and open questions mentioned in [3], there are several directions in which further work on uniformity in computable structure theory might be carried out. One obvious one is to look for other notions whose uniform versions might be of interest. For instance, are there natural notions of uniform computable dimension?

Even along the lines of what was done in the previous sections, there might be further interesting questions. Instead of looking at uniform computable categoricity, one might look at the uniform analogs of $\mathfrak{C}$ categoricity for other classes of degrees $\mathfrak{C}$ (for example, Δ^0_2 categoricity). Similarly, one might look at the uniform analogs of intrinsic $\mathfrak{C}$-ness for classes of relations $\mathfrak{C}$ other than computable and c.e. relations. In both cases, there may well be notions of uniformity intermediate between the index-based and operator notions.

Two final possibilities are to investigate further the connections with type 2 computability mentioned in the paragraph preceding Theorem 2.2 and to examine the index-based notion of uniformity in the general context of [3].

References

[1] C. J. Ash and J. F. Knight, Relatively recursive expansions I, Fund. Math. 140 (1992) 137–155.

[2] C. J. Ash, J. F. Knight, M. S. Manasse, and T. A. Slaman, Generic copies of countable structures, Ann. Pure Appl. Logic 42 (1989) 195–205.

[3] C. J. Ash, J. F. Knight, and T. A. Slaman, Relatively recursive expansions II, Fund. Math. 142 (1993) 147–161.

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

[5] J. Chisholm, The complexity of intrinsically r.e. subsets of existentially decidable models, J. Symbolic Logic 55 (1990) 1213–1232.

[6] J. Chisholm, Effective model theory vs. recursive model theory, J. Symbolic Logic 55 (1990) 1168–1191.

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

[8] R. G. Downey, Computability theory and linear orderings, in Y. L. Ershov, S. S. Goncharov, A. Nerode, and J. B. Remmel (eds.), Handbook of Recursive Mathematics, vol. 138–139 of Stud. Logic Found. Math. (Elsevier, Amsterdam, 1998) 823–976.

[9] S. S. Goncharov, The quantity of nonautoequivalent constructivizations, Algebra and Logic 16 (1977) 169–185.

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

[11] W. Hodges, Model Theory, vol. 42 of Encyclopedia Math. Appl. (Cambridge University Press, Cambridge, 1993).

[12] B. Khoussainov and R. A. Shore, Computable isomorphisms, degree spectra of relations, and Scott families, Ann. Pure Appl. Logic 93 (1998) 153–193.

[13] G. Kreisel, D. Lacombe, and J. R. Shoenfield, Partial recursive functionals and effective operations, in A. Heyting (ed.), Constructivity in Mathematics: Proceedings of the Colloquium held at Amsterdam, 1957 (North–Holland Publishing Co., Amsterdam, 1957) 195–207.

[14] O. V. Kudinov, An autostable 1-decidable model without a computable Scott family of $\exists$-formulas, Algebra and Logic 35 (1996) 255–260.

[15] O. V. Kudinov, Some properties of autostable models, Algebra and Logic 35 (1996) 384–391.

[16] O. V. Kudinov, A description of autostable models, Algebra and Logic 36 (1997) 16–22.

[17] M. S. Manasse, Techniques and Counterexamples in Almost Categorical Recursive Model Theory, PhD Thesis, University of Wisconsin, Madison, WI (1982).

[18] C. F. D. McCoy, Finite computable dimension does not relativize, Arch. Math. Logic 41 (2002) 309–320.

[19] T. Millar, Recursive categoricity and persistence, J. Symbolic Logic 51 (1986) 430–434.

[20] J. Myhill and J. C. Shepherdson, Effective operations on partial recursive functions, Z. Math. Logik Grundlag. Math. 1 (1955) 310–317.

[21] H. Rogers, Jr., Theory of Recursive Functions and Effective Computability, 2nd ed. (MIT Press, Cambridge, Mass., 1987).

[22] R. I. Soare, Recursively Enumerable Sets and Degrees, Perspect. Math. Logic (Springer–Verlag, Heidelberg, 1987).

[23] Y. G. Ventsov, Effective choice for relations and reducibilities in classes of constructive and positive models, Algebra and Logic 31 (1992) 63–73.

[24] Y. G. Ventsov, Effective choice operations on constructive and positive models, Algebra and Logic 32 (1993) 23–28.

[25] V. V. V'yugin, On some examples of upper semilattices of computable enumerations, Algebra and Logic 12 (1973) 287–296.